A Framework for Temporal Web Analytics

Marc Spaniol

UNIVERSITÉ DE CAEN BASSE-NORMANDIE
GREYC – CNRS UMR 6072
ED SIMEM

Habilitation Thesis
in Computer Science

A Framework for Temporal Web Analytics

Marc Thomas Spaniol
defended on December 9, 2014

Committee:

Prof. Patrice Bellot, Université Aix-Marseille, LSIS	Reviewer
Prof. Eric Gaussier, Université Joseph Fourier, LIG	Reviewer
Dr. Mathieu Roche, CIRAD, TETIS	Reviewer
Prof. Mohand Bouganhem, Université Paul Sabatier, IRIT	Examiner
Prof. Aldo Gangemi, Université Paris 13, LIPN	Examiner
Prof. Gerhard Weikum, MPII Saarbrücken (Germany)	Co-Advisor
Prof. Gaël Dias, Université de Caen, GREYC	Advisor

Abstract

Web-preservation organization like the Internet Archive not only capture the history of born-digital content but also reflect the zeitgeist of different time periods over more than a decade. This longitudinal data is a potential gold mine for researchers like sociologists, politologists, media and market analysts, or experts on intellectual property.

Longitudinal data analytics – the Web of the Past – poses research challenges, but has not received due attention. The sheer size and content of Web archives render them relevant to analysts within a range of domains. The Internet Archive holds more than 350 billion versions of Web pages, captured since 1996. This coverage can no longer be maintained, as Web content is growing at enormous rates. A high-coverage archive would have to be an order of magnitude larger.

A Web archive of timestamped versions of Web sites over a long-term time horizon opens up great opportunities for analysts. However, difficulties arise from name ambiguities, requiring a disambiguation mapping of mentions (noun phrases in the text) onto entities. For example, "Bill Clinton" might be the former US president William Jefferson Clinton, or any other William Clinton contained in Wikipedia. Ambiguity further increases if the text only contains "Clinton" or a phrase like "the US president". The temporal dimension introduces additional complexity, for example when names of entities have changed over time (e.g. people getting married or divorced, or organizations that undergo restructuring in their identities). By mapping names and phrases onto canonicalized entities, we raise the entire analytics to a semantic rather than keyword-level in order to make sense of the raw and often noisy Web contents.

Acknowledgement

This habilitation thesis presents the research I have conducted at the Max-Planck-Institute for Informatics (MPI-INF) in Saarbrücken, Germany. At MPI-INF my research has been to a large extent conducted within the scope of three projects funded by the European Union (EU) and an interdisciplinary researcher network funded by the German science foundation (DFG). During my time at MPI-INF and throughout the course of the various projects I had the chance to have inspiring conversations with many colleagues. At this point, I would like to explicitly thank several of them for their support in preparing this thesis:

- I would particularly like to thank Prof. Gerhard Weikum for giving me the opportunity to become a PostDoc in his prestigious group. He enabled me to "discover new worlds" in the area of temporal Web analytics and Big Data. His support and mentoring were indispensable prerequisites in creating this thesis and publishing papers at top tier conferences. He impressed me a lot by his relentless pursuit of perfection and his enormous efforts he undertakes in mentoring his students and PostDocs. I have greatly benefited from the time and experiences gained working under his patronage.

- Next, I would like to thank my colleagues from all over Europe that I learned to know throughout the course of the projects I have been working on. It was a real pleasure working in the scope of collaborative research projects, which I have always considered as enriching and inspiring. In particular, I would like to thank Prof. Andras Benczúr, Dr. Claudia Niederée, Dr. Thomas Risse, Prof. Philippe Rigaux and Mr. Mark Williamson for the great time we spent at meetings, exhibitions and/or reviews.

- Of course, (almost) nobody is able to master complex projects and explore new fields of research in complete isolation. I found great support in my colleagues at MPI-INF, who gave me valuable feedback on my ideas and helped me realizing them. In particular, I would like to thank my (former) doctoral students Dr. Dimitar Denev, Mrs. Natalia Prytkova, Dr. Yafang Wang and Mr. Mohamed Amir Yosef, with whom I have been working in close collaboration.

- In order to manage daily business, I found great support in our secretaries Mrs. Petra Schaaf and Mrs. Andrea Ruffing. They were particularly helpful to me when it came to travel planning, which turned out to be not too infrequent.

- In terms of financial project administration, I found excellent support in Mrs. Anja Zimmer. She assisted me a lot in filling out the forms properly, which also helped to submit the required documents always immaculate and on time.
- I was also very glad to have Prof. Ricardo Baeza-Yates and Mr. Julien Masanès at my side, who helped me in establishing a novel Workshop series on "Temporal Web Analytics" in conjunction with the renowned World Wide Conference. Within this novel and striving research community, it was always a particular pleasure to meet and exchange ideas with Dr. Omar Alonso and Prof. Adam Jatowt.
- Within the scientific network on "Media of collective Intelligence" I was able to discuss aspects of collective Web intelligence against the background of media theory in interdisciplinary workshops. I am very grateful to Prof. Isabell Otto, who invited me to become a member of this network.
- I would also like to express my gratitude to all members of the jury, who enabled me to defend this thesis. In particular, I would like to thank Prof. Gaël Dias, who encouraged me in submitting this thesis at the Université de Caen Basse-Normandie. His guidance helped a lot in paving my way through the administrative process.
- Sincere thanks also goes to my former colleagues from the excellence cluster Multimodal Computing and Interaction (MMCI) Dr. Andreas Broschart and Mr. Christian Pölitz. During our after hour talks it was always a pleasure to reflect about daily business and to chitchat about less serious topics.
- Last but not least, I would like to thank my parents. They helped me to overcome hard times and always encouraged me to carry on.

Encore une fois, merci beaucoup à tous!

Saarbrücken, 14th February 2014

Contents

Chapter 1

Introduction

National libraries and organizations like the Internet Archive (https://archive.org) and its European sibling (http://internetmemory.org) have been capturing Web contents over more than a decade and have protected Web contents from vanishing [Masa06]. The emergence of large Web-contents repositories and digitization projects open up an entirely new range of analytical opportunities and challenges along the temporal dimension [AGBa07]. Studies reveal "culturomics" phenomena [MSAi11], track the trustworthyness of memes over time (truthy[1]) or even investigate the Web's predictive power (such as recorded future[2] or time explorer[3]). The Temporal Web Analytics workshop series (TempWeb[4]) has been launched as a forum for such topics.

1.1 Overall Research Question

The constantly evolving Web reflects the evolution of society in the cyberspace. For instance, knowledge about entities (people, companies, political parties, etc.) evolves over time. New knowledge is added (e.g., awards) or changes (e.g., spouses, CEOs and similar positions). In addition, events related with these changes are reflected on the Web by postings in blogs, updates on corporate Websites, newspaper articles or even Wikipedia (depending on the importance of the respective entity). Furthermore, abstracting from individual entities, we observe long-term changes in terminologies and topical taxonomies. The underlying assumption is that events (such as affairs and awards on the temporal fact level, news coverage and changes in public opinion) depend on each other and show co-occurrence patterns in media. In order to understand these mutual dependencies between

[1] http://truthy.indiana.edu
[2] https://www.recordedfuture.com
[3] http://fbmya01.barcelonamedia.org:8080/future
[4] http://temporalweb.net/

Web contents and the evolving societal knowledge it represents, a systematic approach toward comprehensively tracing and exploiting Web contents is required.

The goal of Web archiving is to preserve the history of Web sites by repeatedly crawling entire sites and adding versions of both page-contents and page-link structures to an append-only archive. The most well-known endeavor of this kind is the work of the Internet Archive, but national libraries and national archives also have specialized activities along these lines. Capturing the history of digitally born information and preserving the cultural and political zeitgeist of an era offers a potential gold mine for all kinds of media and business analysts, such as political scientists, sociologists, media psychologists, market analysts, and intellectual-property lawyers.

For example, one could track and analyze public statements made by representatives of companies such as SAP or Oracle, characterizing the evolution of their attitude towards green IT. Another example could be tracking, over a long time horizon, a politician's public appearances: which cities has she/he visited, which other politicians or business leaders has she/he met, etc.

1.2 Research Setting

This thesis presents the research I have conducted since I obtained my dissertation in August 2007 from RWTH Aachen University and then moved on to the Max-Planck-Institute for Informatics (MPI-INF) in Saarbrücken, Germany. My research interests are situated in the field of databases and information systems, which I have been working in since my graduation in December 2001.

The research at MPI-INF has been conducted within the scope of three projects funded by the European Union (EU) and an interdisciplinary researcher network funded by the German science foundation (DFG). To this end, results in this thesis combine fundamental research in the area of quality-conscious Web archiving, temporal Web mining and Web-scale entity analytics, which has been underpinned from a media theoretic perspective with respect to the societal impacts of media on collective intelligence. Further, research has also been driven by the requirements of application partners from industry. As such, several research prototypes have been successfully transferred into theses companies and organizations for further exploitation. In the following, I will give a brief overview on the background of the before mentioned projects and will highlight the main research aspects pursued.

1.2.1 EU STREP on "Living Web Archives (LiWA)"

LiWA (Living Web Archives, http://liwa-project.eu/) was an EU STREP (Specific Targeted Research Project) that has been started in February 2008 and been successfully finalized in January 2011. LiWA involved 8 partners from academia, industry, and library services (L3S

Hannover, Internet Memory Foundation, Hungarian Academy of Sciences, Hanzo Archives Ltd., Instituut voor Beeld en Geluid, National Library of the Czech Republic, Moravian Library, MPI-INF). The goal of LiWA has been the development of next-generation Web archiving technologies. Within LiWA we have coordinated the work on temporal coherence, which complemented ongoing fundamental research in the area of Web archiving and Web mining. Main research issues were improving the capturing process of Web sites for high-quality archives and the interpretability of contents for later retrieval and analysis. Our novel models and strategies have been fully integrated into the Internet Memory Foundation's Web crawler Heritrix (http://www.crawler.archive.org/).

1.2.2 EU IP on "LivingKnowledge: Facts, Opinions and Bias in Time"

LivingKnowledge was an Integrated Project (IP) in the area of Web mining that ran from February 2009 until January 2012. It investigated the diversity and the evolution of facts, opinions, and bias as expressed in digital media. The project involved 10 partners from academia, industry, and public archival institutions (University of Trento, Yahoo! Research Barcelona, SORA Institute for Social Research and Analysis, Italian National Inter-University Consortium for Telecommunications, Internet Memory Foundation, University of Pavia, University of Southampton, Indian Statistical Institute, L3S Research Center Hannover, MPI-INF). LivingKnowledge considered diversity to be an asset and made it traceable, understandable, and exploitable, by improving navigation, exploration, and search in very large multimodal datasets. Within LivingKnowledge, we coordinated the work on knowledge evolution and contributed to knowledge evolution and advanced fact extraction. Here, we studied the impact of time on entities and emerging concepts. To this end, research has been focused on "raising" plain (Web) text mentions to entities and identifying emerging concepts in order to trace their evolution along the temporal dimension.

1.2.3 EU STREP on "Longitudinal Analytics of Web Archive data (LAWA)"

The project on Longitudinal Analytics of Web Archive data (LAWA) was an EU STREP (Specific Targeted Research Project) that started in September 2010 and has been successfully completed in August 2013 (lawa-project.eu). LAWA and involved 5 other partners from academia and archiving institutions (Hebrew University Jerusalem, Internet Memory Foundation, Hungarian Academy of Sciences, Hanzo Archives Ltd., University of Patras) and was coordinated by MPI-INF. LAWA was funded by FIRE (Future Internet Research & Experimentation, cordis.europa.eu/fp7/ict/fire) initiative and backed by the rich Web repository of the Internet Memory Foundation (formerly called the European Archive), in order to create a blueprint of a Virtual Web Observatory. The goal of LAWA was to build an Internet-based experimental testbed for large-scale data analytics. Its focus has been

on developing a sustainable infrastructure, scalable methods, and easily usable software tools for aggregating, querying, and analyzing heterogeneous data at Internet scale. Within LAWA, we have studied methods for longitudinal data analytics of archived Web data that has been crawled over extended time periods and extended our previous research on entity-level analytics. To this end,edwe develop data analytics software that allowed us to study Web archive data at different levels of granularity. At the fine-grained level, our methods enabled us to disambiguate entities and analyze temporal relationships among them. At the coarse-grained level, the developed approaches allowed investigations at large (e.g. between communities or taxonomic structures). Taken together, our software enabled the full spectrum of entity-level Web archive search and exploration.

1.2.4 DFG scientific network on "Media of collective Intelligence"

The scientific network on "Media of collective Intelligence" funded by the German science foundation has been started in May 2011 and brings together 15 researchers from various disciplines, covering complementary aspects of Web related studies. The network aims at providing a survey of existing research fields and theory formation on themes of collective intelligence covering a wide range of organisation-theoretical, computer scientific, cognitive-scientific, sociological, biological and philosophical approaches. To this end, the members of the network assume a constitutive participation of medial processes in the emergence of collective intelligence. In this context mediality is approached from a procedural-logical and difference-theoretical standpoint and is not restricted to technical apparatuses alone. Instead of an exclusive fixation on the computer as a means the question is asked when and under what conditions a technological or social intermediary can become a mediator of collective intelligence. The network asks which different action forms of collaboration, which aesthetic experiments this mediatory work produces and which political consequences it has. Within the the collaborative research network we systematically examine the different formations of collective intelligence in inter-disciplinary, media-comparative and media-historical discussions of collective intelligence.

1.3 Contributions and Structure of this Thesis

This thesis introduces a framework for temporal Web analytics ranging from quality-conscious Web archiving up to entity-level temporal Web analytics. To this end, this framework describes a unified approach toward semantic exploitation of Web analytics and shows perspectives on future research. The following subsections summarize the key contributions and give an overview on the underlying publications.

1.3.1 Quality-conscious Web Archiving

Section 2 addresses the issue of quality-conscious Web archiving by ensuring the archive's "coherence"[5], which – in terms of a Web site – results in a "harmonious connexion of the several parts, so that the whole 'hangs together"'. From an archiving point of view, the ideal case to ensure highest possible data quality of an archive would be to "freeze" the complete contents of an entire Web site during the time span of capturing the site. Of course, this is illusion and practically infeasible. Consequently, one may never be sure if the contents collected so far are still consistent with those contents to be crawled next. However, temporal coherence in Web archiving is a key issue in order to capture digital contents in a reproducible and, thus, later on interpretable manner. To this end, we have developed strategies that help to overcome (or at least identify) the temporal diffusion of Web crawls that last from a view hours only up to several days. The underlying coherence framework is capable of dealing with proper as well as improper dated contents. Depending on the data quality provided by the Web server, different coherence optimizing crawling strategies have been developed, which outperform existing approaches and have been tested under real life conditions. Even more, due to the development of a smart revisit strategy for crawlers we are also capable of discovering and (as a consequence) of ensuring coherence for contents, which are improperly dated and not correctly interpretable with conventional archiving technologies. To this end, temporal coherence of Web archiving becomes traceable under real life applications and we provide strategies to improve the quality of Web Archives, regardless of how unreliable Web servers are.

1.3.2 Entities on the Web

Web archives contain mentions of named entities such as people, places, organizations, etc. This entity information is an important asset when contents are being analyzed. To this end, the disambiguation of named entities in natural language text needs to map mentions of ambiguous names onto canonical entities in order to semantically exploit Web data. Section 3 explores how to link entities with high quality data and knowledge sources. By aggregating data from various sources, contents can be understood in their respective temporal context. However, names are often ambiguous and the same name can have many different meanings. To this end, methods and tools are required in order to lift analytics from keywords to a semantic level.

1.3.3 Evolution of Entities and Emerging Concepts

Section 4 describes the asset of entity information for making sense of the raw and often noisy data, which allows to track people, companies, products, songs, etc. in Web pages and

[5] http://dictionary.oed.com

social media over time. On the fine-grained level, analyzing the evolution of entities requires the extraction of temporal facts about them. These data enable a semantic interpretation of Web contents and their inherent dependencies. At large, temporal analytics of "entities" aims at the discovery of emerging concepts and their adaptation in widely used categorization schemes that reflect the collective memory/knowledge of society. Effects of this process in taxonomic category schemes are, for instance, the change of a term's meaning (such as Apple additionally becoming a computer category) or a newly appearing topic (such as SARS, iPhone, etc.). As such methods are required that monitor a (prior) oscillation of word and phrase occurrences in documents, such as blogs, Websites, or news articles in order to predict changes in categorization schemes.

1.3.4 Temporal Web Analytics in Action

In combination, the before described building blocks are valuable assets for comprehensive temporal Web analytics. To this end, Section 5 presents methods and tools that make temporal Web analytics better understandable and explainable. On the top level of a high quality Web archive, these software tools support entity-level analytics of heterogeneous data at Internet scale. The analytics applications can be classified into browser plug-ins and analytics interfaces. To this end, the browser plug-ins show-case dedicated technologies (e.g. entity-type classification) on the live as well as on the archived Web.

1.4 Overview on Publications

This thesis is based on publications that have been published at highly visible and selective conferences and workshops. Research in the area of quality-conscious Web archiving has been published at VLDB [DMSW09] and VLDBJ [DMSW11] as well as workshops at WWW [SDM*09] and ECDL [SMDW09]. Publications on entity-related research is subdivided into extraction aspects and the subsequent tracing. To this end, research on entity classification and disambiguation has been published at natural language conferences, such as EMNLP [HYB*11], Coling [YBH*12] and IEEE Data Engineering Bulletin [WHN*12]. In addition, results employing entity information for sophisticated knowledge mining tasks have been published at CIKM [WYQ*11, WDR*12], ACL [WDSW12], EDBT [WZQ*10] and WebDB [PSWe12]. Finally, the before mentioned aspects have been "'glued" together within the scope of the LAWA project (cf. lawa-project.eu for details). Hence, the remaining publications are demonstrators exploiting the previously gathered knowledge that have been published at WWW [WYZ*11], VLDB [YHB*11], ACL [YBH*13] and the World Statistics Congress [SPWe13] as well as selected project results from deliverables [YHP*12, BRSW13, SpBe13] and publications in WWW [SpWe12] and ERCIM News [SBVW12].

Chapter 2

Archiving the Web

Web archiving is commonly understood as a continuous process that aims at archiving the entire Web (broad scope). Despite the initiatives of the International Internet Preservation Consortium (IIPC)[1] and the Internet Archive[2] in capturing Web contents for future generations, limitations such as storage space, bandwidth, and crawling politeness as well as threats such as Web spam and crawler traps heavily affect the crawling performance and, thus, the quality of the collected data [RMSp09, MRSp10]. Current methods are based on snapshot crawls and "exact duplicate" detection [Masa06]. The coherence of data in terms of proper dating and proper cross-linkage is influenced by the temporal characteristics (duration, frequency, etc.) of the crawl process.

2.1 Quality-conscious Web Archiving

A typical scenario in archiving institutions or companies is to periodically – e.g. monthly – create high quality captures of a certain Web site [SDM*09, DMSW09, DMSW11, MDSW10]. These periodic domain scope crawls of Web sites aim at obtaining a best possible representation of a site. A reason for customers having their site archived on a regular basis is, for instance, to guard itself against accusations regarding intellectual property rights, fraud or non-compliance with legal requirements (e.g. EU laws about imprints, terms of use, etc.). Figure 2.1 contains an abstract representation of such a domain scope crawling process. This Web site consists of n pages $(p_1, \ldots, p_n)$. Each of them consists of several successive versions, indicated by the horizontal lines (e.g., p_n has three different versions in $[t; t']$). Ideally, the result of a crawl would be a complete and instantaneous snapshot of all pages at a given point of time. In reality, one crawl requires an extended time period to gather all pages of a site while being potentially modified in parallel, causing thus

[1] http://netpreserve.org
[2] http://www.archive.org

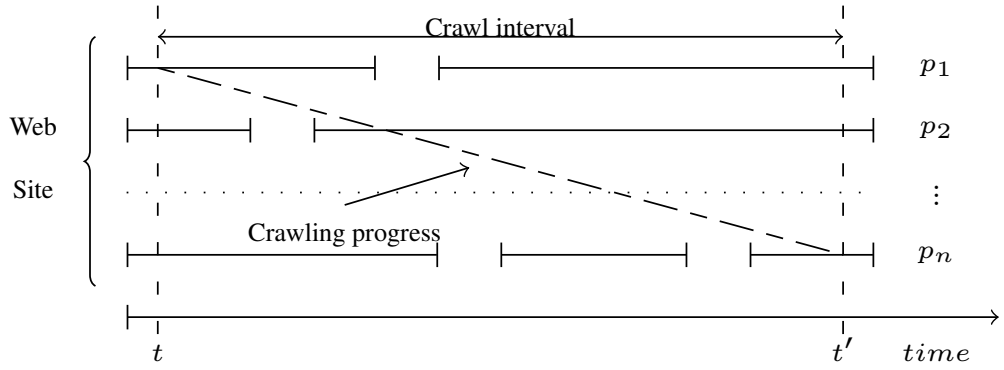

Figure 2.1: Web site crawling process (domain scope)

incoherencies in the archive. The risk of incoherence increases further due to politeness constraints and need for sophisticated time stamping mechanisms.

Figure 2.2 depicts coherence defects in a Web archive [SMDW09]. In this case, the coherence defects are caused by references from a Web page to other pages, which have already been superseded by more recent versions. In this case, the archived documents on the left-hand side are incoherent (highlighted by a red frame) with respect to the entry page (reference time point "as of" 17/02/2007). However, the link from the entry page to the page on the right-hand side that has been archived on 19/02/2007 is coherent (indicated by a green frame), because both pages are valid and unchanged "as of" 17/02/2007.

What is easy to recognize to be incoherent as a human is more difficult for a machine. A computer might only be able to a (very) limited degree interpret the temporal aspect of a page. Nevertheless, given the last modification dates as reference time points of observation, we are able to reason about coherence defects between two instances of a document. To this end, we introduce several techniques at different levels of granularity to reason about the time point when contents have been modified and – subsequently – analyze if coherence defects exist.

2.2 Coherence Defect Analysis

The coherence defect analysis visualizes the quality of archived Web sites based on the amount of change that took place during the time of archiving. To this end, we have developed methods for automatically generating sophisticated statistics and visualizations (e.g. number of defects occurred sorted by defect type).

Figure 2.3 depicts a sample visualization of an mpi-inf.mpg.de domain crawl (about 65.000 Web contents) with the visone[3]. Depending on the nodes' size, shape, and color the user

[3]http://visone.info

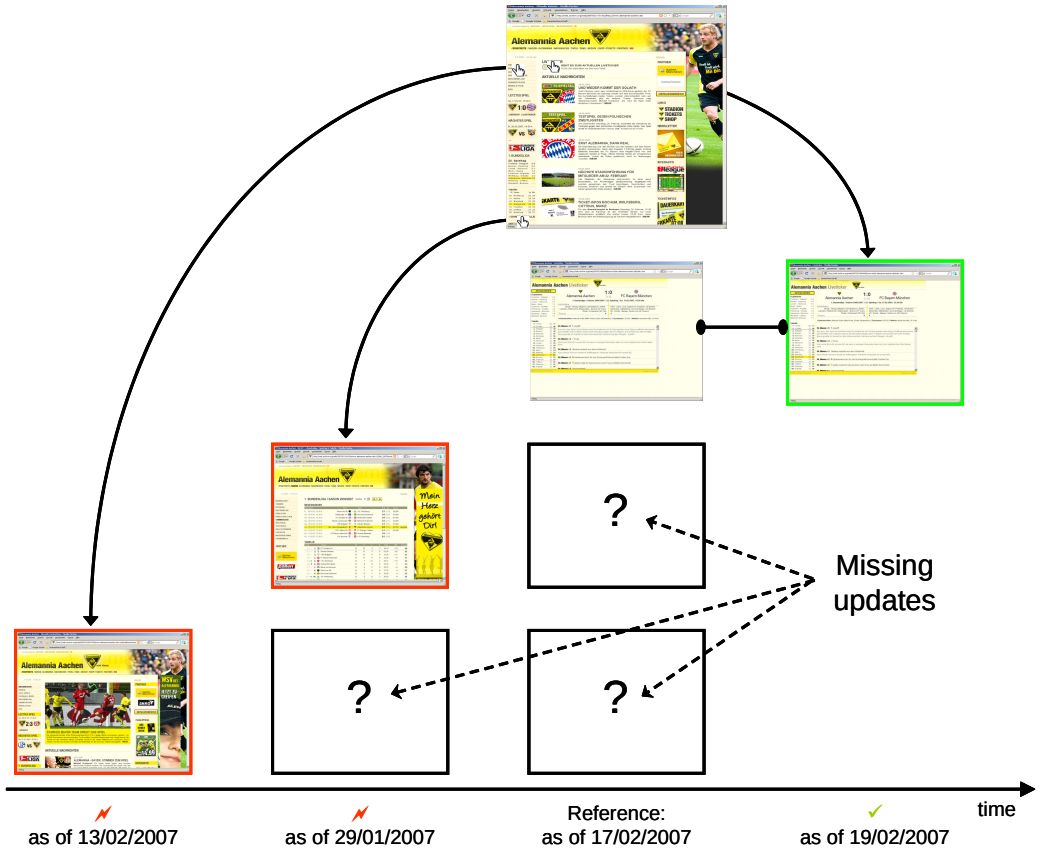

Figure 2.2: Coherence defects in a Web archive for www.alemannia-aachen.de “as of” 17/02/2007

gets an immediate overview on the success or failure of the capturing process. In particular, a node’s size is proportional to the amount of coherent Web contents contained in its subtree. In the same sense, a node’s color highlights its “coherence status”. While green stands for coherence, the signal colors yellow and red indicated (content incoherence and/or link structure incoherence). The most serious defect class of missing contents is colored in black. Finally, a node’s shape indicates its MIME type ranging from circles (HTML contents), hexagons (multimedia contents), rounded rectangles (Flash or similar), squares (PDF contents and other binaries) to triangles (DNS lookups). This visual metaphor is intended as an additional means to automated statistics for understanding the problems that occurred during capturing. Its main field of application is the analysis of high quality (single) Web site crawls.

In order to obtain a perfectly coherent Web archive, an ideal approach would be to have captures for every domain at any point in time whenever there is a (small) change in any of the domain’s pages. Of course, this is absolutely infeasible given the enormous size of the Web, high content-production rates in blogs and other Web 2.0 venues, the disk and server costs of a Web archive, and also the politeness rules that Web sites impose on crawlers. We

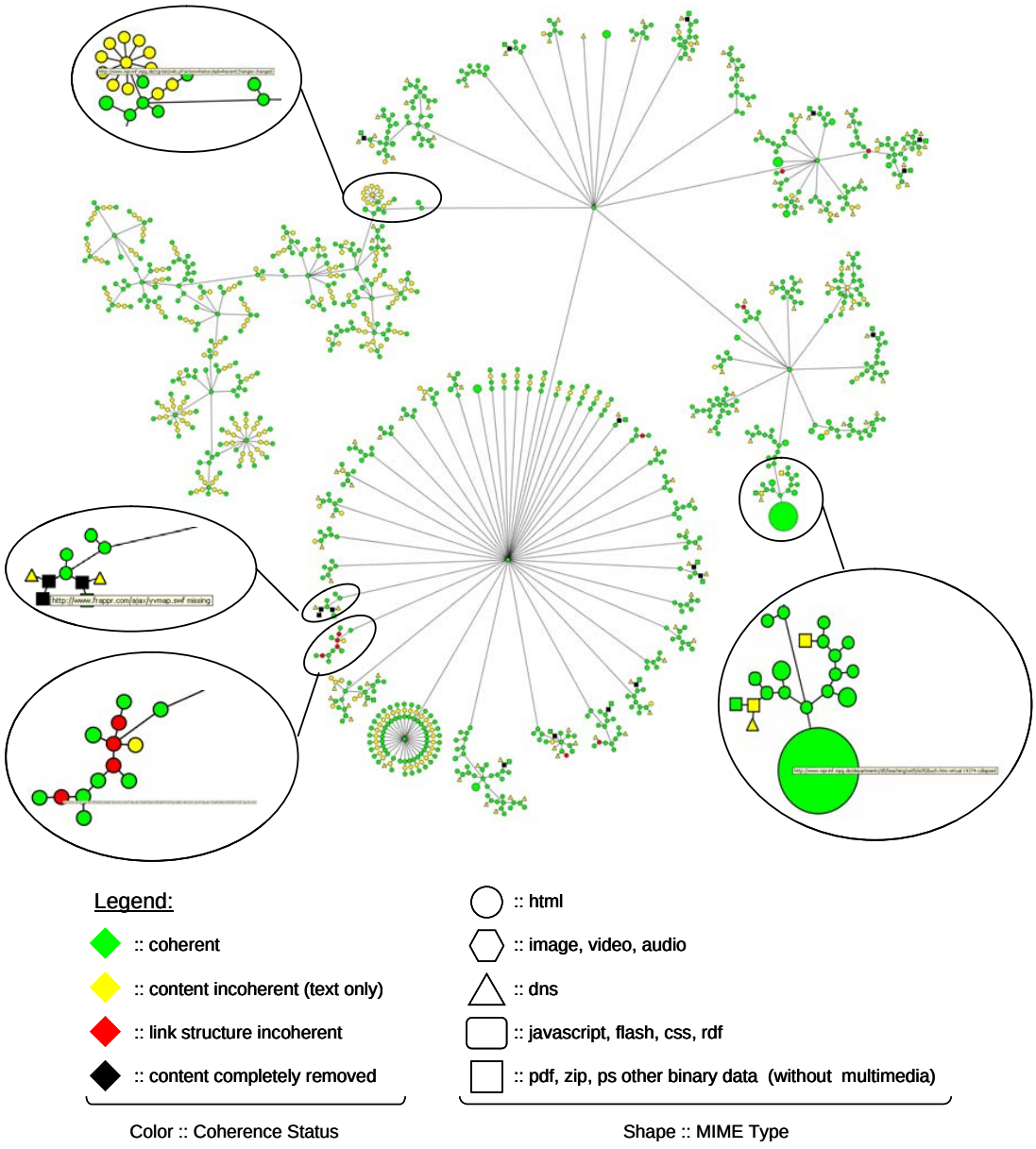

Figure 2.3: Coherence defect visualization of a single crawl-recrawl pair of mpi-inf.mpg.de by visone

therefore settle for the realistic goal capturing Web sites at convenient points (whenever the crawler decides to devote resources to the site and the site does not seem to be highly loaded), but when doing so, the capture should be as "authentic" as possible. In order to ensure an "as of time point x (or interval $[x, y]$)" capture of a Web site, we need to develop an archiving crawler that *ensures coherence of crawls regarding a time point or interval*, and *identifies those contents of the Web site that violate coherence* [SDM*09].

2.3 Archive Coherence Model

We now introduce our archive coherence model. We start with basic assumptions and the notation. After that, we introduce our definition inducible coherence that will be applied to subsequently quantify coherence. Finally, we express the probability of incoherence.

2.3.1 Assumptions

In the following, we assume that a Web site to be crawled consists of n Web pages that change over time that occur independent of each other. Time for downloading contents is neglected and the time between any two subsequent downloads is equal. Change probabilities are considered to be known for any two pages.

2.3.2 Notation

We assume that a Web site to be crawled consists of n Web pages numbered $\{p_1, \ldots, p_n\}$. Changes of these Web pages occur per time unit immediately before download and independent of each other according to the probabilities $\lambda_1, \ldots, \lambda_n$ that are associated with the Web page of the corresponding number. We assume that the delay Δt between the downloads of the pages is the same, and the download time is neglected. For convenience $[t_s, t_e]$ denotes the crawl interval, where $t_s = t_1$ is the starting point (download of the first page) of the crawl and $t_e = \Delta t \cdot n = t_n$ is the ending point of the crawl (download of the last page). The time of downloading page p_i is denoted as $t(p_i) = t_j$ having $j \in [1, n]$. In addition, we assume to retrieve the last modified stamps of pages $\mu_1, \ldots, \mu_n$ (having $\mu_i \leq t(p_i)$) upon download. We call the consecutive process of downloading the Web pages $\{p_1, \ldots, p_n\}$ of an entire Web site a crawl c.

2.3.3 Definitions

The following definitions are based on a common notion of "coherence" applied to the issue of Web archiving and – thus – particularly to Web crawling. Our understanding of coherence

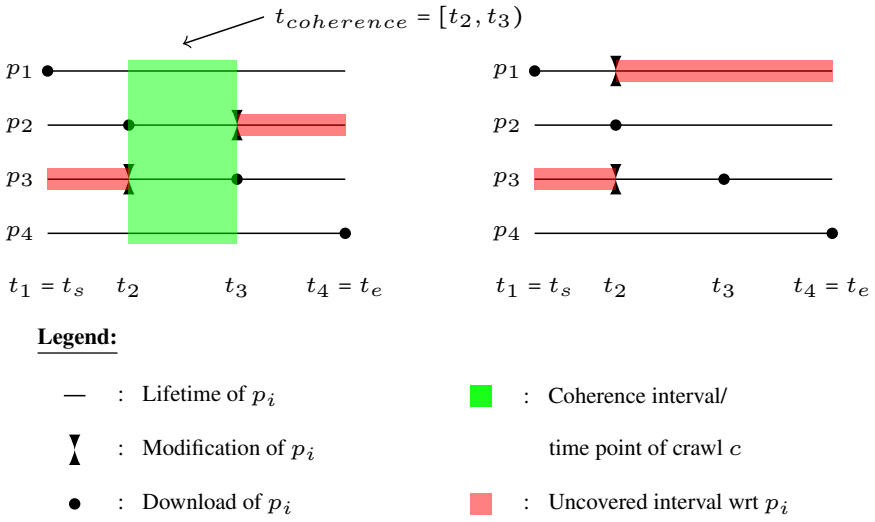

Figure 2.4: Crawl c containing coherence interval (left) and without coherence interval (right)

refers to the explanations given in the Oxford English Dictionary[4] describing coherence as "the action or fact of cleaving or sticking together", which – in terms of a Web site – results in a "harmonious connexion of the several parts, so that the whole 'hangs together'".

Defintion 2.3.1. *Coherence*
1. A single Web page is always coherent.
2. The invariance interval $[\mu_i, \mu_i^*]$ *of page* p_i *lies between the last modified time stamp* μ_i *at time* $t(p_i)$ *of downloading* p_i *(*$\mu_i \leq t(p_i)$*) and the next change* μ_i^* *following* $t(p_i)$*.*
3. Two or more pages are coherent if there is a time point (or interval) $t_{coherence}$ *so that a non-empty intersection among the invariance interval of all pages exists:*

$$\forall p_i, \exists t_{coherence} : t_{coherence} \in \bigcap_{i=1}^{n} [\mu_i, \mu_i^*] \neq \varnothing$$

Figure 2.4 depicts our definition of coherence in a graphical representation of a Web site consisting of four pages $(p_1, \ldots, p_4)$. The download of a page p_i by the crawler is indicated by a black circle. As described above, we assume that there is a single download per time unit. In our example, the download sequence is given as p_1, p_2, p_3, p_4 occurring at t_1, t_2, t_3, t_4. Even more, all pages exist during the whole crawl interval $[t_1, t_4]$. However, some pages are subject to changes taking place in the crawl interval. In the example on the left hand side of figure 2.4, page p_2 changes at t_3 and page p_3 changes at t_2. In combination with the crawl sequence mentioned before, this results in a coherence interval spanning from t_2 till t_3 ($t_{coherence} = [t_2, t_3)$). In the example on the right hand side of figure 2.4, page

[4] http://dictionary.oed.com

p_1 and p_3 change at t_2. In consideration of the same crawl sequence as before, this results in an empty coherence interval ($t_{coherence} \in \varnothing$). Since the union of the uncovered intervals of pages p_1 and p_3 spans the whole crawl interval, there is not a single time point that ensures an "as of time point x (or interval $[x, y]$)" capture of this Web site.

From a practical point of view, the definition of coherence introduced before is of limited value only. The key point is simple: A real life crawler is "left-hand side aware", but "right-hand side blind". Since a Web page might change immediately after its download, a crawler can only be certain about the appearance of page p_i within an interval lasting from (if available) this page's last modification date μ_i until its time of download $t(p_i)$ during the course of crawl c. To this end, we now introduce a definition of observable coherence, which allows a crawler to disclose coherence based on the last modified stamp of Web pages.

Defintion 2.3.2. *Observable Coherence*
Two or more Web contents are observable coherent if there is a single time point $t_{coherence}$ so that there is a non-empty intersection of the intervals spanning the respective download time $t(p_i)$ and the corresponding last modified stamp μ_i retrieved at time of download ($\mu_i \leq t_i$):

$$\forall p_i, \exists t_{coherence} : t_{coherence} \in \bigcap_{i=1}^{n} [\mu_i, t(p_i)] \neq \varnothing$$

Due to the fact that a crawler is "left-hand side aware", but "right-hand side blind" the time point $t_{coherence}$ needs to be carefully selected. In case, a coherence statement is desired about all crawled contents of a Web site a specific case of observable coherence – called measurability – is required. Given only the knowledge obtained through a crawl c spanning the time interval $[t_1, t_n]$, measurability is given only for a single coherence time point ($t_1 = t_{coherence}$).

Lemma 2.3.3. *Measurable Coherence*
Given a Crawl c measurable coherence can only be given if the crawl is observable coherent. Measurability is only given for $t_1 = t_{coherence}$.

Proof. Assume the contrary and pick a random page p_i downloaded after p_1 (that means $t(p_1) < t(p_i)$) to become the reference time point of assurance. Without the loss of generality we pick the next page (that means p_2) to become the new reference time point of assurance. Since we require measurability we have to intersect over all measurable coherence intervals. These are (at least) $[\mu_1, t(p_1)]$ and $[\mu_2, t(p_2)]$, having $\mu_2 \leq t(p_2)$. However, when intersecting the intervals with respect to any other point than t_1 (like $t_{coherence} = t_2$ in this case), the intersection is empty by definition since the interval of p_1 ends at $t(p_1) = t_1$. □

Figure 2.5 highlights the concept of measurable coherence (as a specialization of observable coherence) in a graphical representation of a Web site that consists of n pages. Assuming a download sequence $p_1, \ldots, p_n$ spanning the crawl interval $[t_1, t_n]$ the respective observation

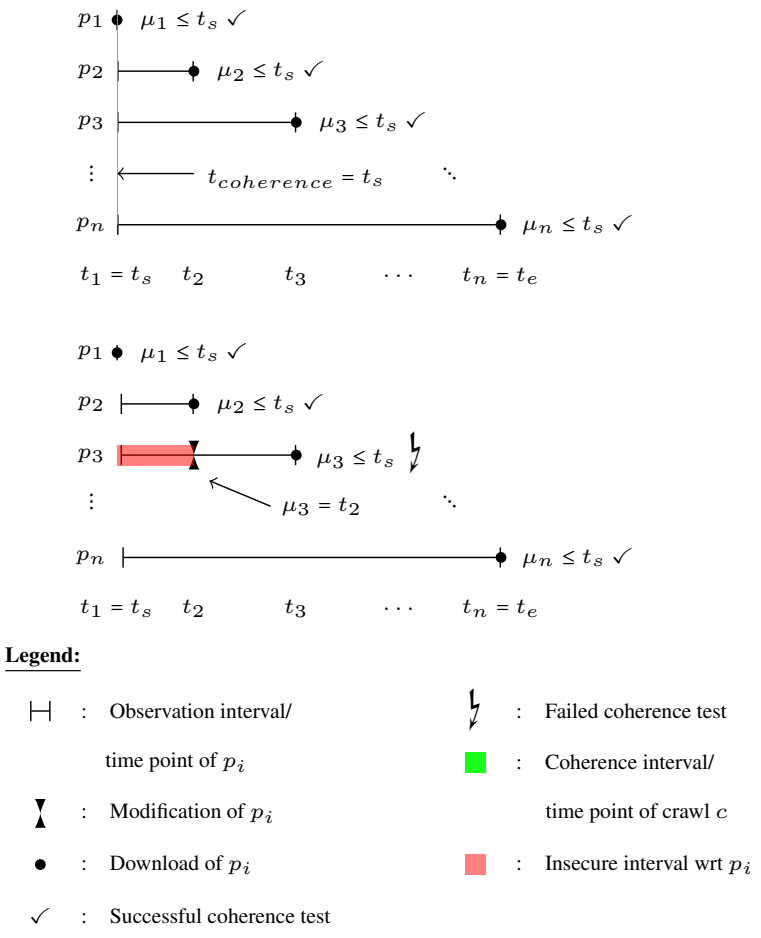

Figure 2.5: Measurable coherence fulfilled (top) and violation of measurable coherence (bottom)

intervals/time points span from t_1 (wrt p_1) up to $[t_1, t_n]$ (wrt p_n). As a consequence, the risk of a single Web page p_i being incoherent heavily depends on its position during the course of crawl c, which will be subject to further investigation in section 2.3.4. The top of figure 2.5 depicts n successful tests of measurable coherence. This results in an assurable coherence statement for the entire Web site valid at time point $t_{coherence} = t_s$. By contrast, the bottom of figure 2.5 indicates a failed measurable coherence test for page p_3. In this case, page p_3 was modified at t_2, which resulted in an updated last modified stamp ($\mu_3 = t_2$). For that reason, there might not be given an assurable coherence statement for the entire Web site, since the time interval $[t_1, t_2)$ is insecure with respect to page p_3.

In reality and against the background of large Web sites it appears almost impossible to achieve an assurable coherence statement. Though, we might not be able to assure coherence for an entire Web site, we might still be interested in specifying how "coherent" the remaining parts of our crawl c are. For that purpose, we introduce a metric that allows

us to quantify the quality of a crawl c.

Defintion 2.3.4. *Quantifying Measurable Coherence*
First we define an error function $f(p_i)$ that counts the occurring incoherences during the crawl:

$$f(p_i) = \begin{cases} 0 & , if\ \mu_i \leq t_s \\ 1 & , else \end{cases}$$

The overall quality of a crawl c is then defined as:

$$q(c) = \frac{\sum_{i=1}^{n} f(p_i)}{n},\ n \geq 1$$

Unfortunately, the reliability of last modified stamps cannot be guaranteed due to missing trustworthiness of Web servers. Hence, the only 100% reliable method is to self create a "virtual time stamp" by comparing the page's etag or content hash with its previously downloaded version. To this end, we introduce an induced coherence measure that allows to gain full control over the contents being compared.

We now apply a crawl-revisit sequence $\Pi(c, r)$, where r is a subsequent revisit of the previously crawled set of Web pages $\{p_1, \ldots, p_n\}$. In this consecutive revisit process we obtain a second (and potentially different) version of the previously crawled pages denoted as $\{\tilde{p}_1, \ldots, \tilde{p}_n\}$. Hence, the crawl-revisit sequence $\Pi(c, r)$ consists of n crawl-revisit tuples $\pi(p_i, \tilde{p}_i)$ having $i \in \{1, n\}$. Technically, the last crawled page p_v having $t(p_v) = n$ is not revisited again, but considered as crawled and revisited page at the same time. Hence, the revisit takes place in the time interval $[t_{n+1}, t_{2n-1}]$. As for visits, the time of downloading page $\tilde{p}_i$ is denominated as $t(\tilde{p}_i) = t_k$ now having $k \in [n, 2n-1]$. In accordance with the definition of the crawl interval, for convenience we denote $[\tilde{t}_s, \tilde{t}_e]$ to be the revisit interval, where $t_e = \tilde{t}_s = t_n$ is the starting point of the revisit (download of the last visited page that is at the same time the first revisited page) and $\tilde{t}_s = \Delta t \cdot (n-1) = \tilde{t}_e$ is the ending point of the revisit (download of the last revisited page). In addition, we define the etag or content hash of a page or an revisited page as $\theta(m)$ having $m \in \{p_i, \tilde{p}_i\}$. Overall, a complete crawl-revisit sequence $\Pi(c, r)$ spans the interval $[t_1, t_{2n-1}]$. It starts at $t_s = t_1$ with the first download of the crawl and ends at $\tilde{t}_e = t_{2n-1}$ with the last revisit download.

Defintion 2.3.5. *Inducible Coherence*
Two or more Web contents are inducible coherent if there is a time point $t_{coherence}$ between the visitation of pages $t(p_i)$ and the subsequent revisits $t(\tilde{p}_i)$ where the etag or content hash of corresponding pages ($\theta(m)$ having $m \in \{p_i, \tilde{p}_i\}$) has not changed:

$$\forall p_i, \exists t_{coherence} : \theta(p_i) = \theta(\tilde{p}_i) \wedge t_{coherence} \in \bigcap_{i=1}^{n} [t(p_i), t(\tilde{p}_i)]$$

Figure 2.6 highlights the functioning of inducible coherence applied to a Web site consisting of n pages. We assume a download sequence $p_1, \ldots, p_n$ spanning the crawl interval

$[t_1, t_n]$ and an inverted subsequent revisit sequence $\tilde{p}_n, \ldots, \tilde{p}_1$ spanning the revisit interval $[t_n, t_{2n-1}]$. Like with measurable coherence, the risk of a single Web page p_i being incoherent heavily depends on its position in the crawl-revisit sequence $\Pi(c, r)$, which will be subject to further investigation in section 2.3.4. The left upper part of figure 2.6 depicts n successful tests of inducible coherence. This results in an assurable coherence statement for the entire Web site valid at time point $t_{coherence} = t_n$. By contrast, the lower left section of figure 2.6 indicates a failed inducible coherence test for the crawl-revisit tuple $\pi(p_3, \tilde{p}_3)$. In this case, page p_3 was modified elsewhere between $t(p_3) = t_3$ and $t(\tilde{p}_3) = \tilde{t}_{n-3} = t_{2n-3}$, which results in a failed inducible coherence test. We are in this case (due to non-existing or non-reliable last modified stamps) not able to determine the exact time point of modification. To this end, we are only able to discover a boolean result because of a failed etag or hash comparison for the crawl-revisit tuple $\pi(p_3, \tilde{p}_3)$. The whole interval is flagged as insecure, even though, the modification might have taken place far beyond the aspired coherence time point ($t_{coherence} = t_n$). Thus, despite being coherent from a global point of view for $t_{coherence} = t_n$, a real life crawler might not be able to figure this out (cf. figure 2.6 for details). Consequently, there might not be given an assurable coherence statement for the entire Web site, since there is an insecure time interval with respect to the crawl-revisit tuple $\pi(p_3, \tilde{p}_3)$.

Likewise for measurable coherence, in reality and against the background of large Web sites it is almost unfeasible to achieve a coherence statement for an entire Web site based on inducible coherence. Though, we might still be interested in specifying how "coherent" the remaining parts of our crawl c are. For that purpose, we introduce a metric that allows us to express the quality of a crawl c.

Defintion 2.3.6. *Quantifying Inducible Coherence*
First we define an error function $f(\pi(p_i, \tilde{p}_i))$ that counts the occurring incoherences for crawl-revisit tuple $\pi(p_i, \tilde{p}_i)$ of the crawl-revisit sequence $\Pi(c, r)$:

$$f(\pi(p_i, \tilde{p}_i)) = \begin{cases} 0 & , if\ \theta(p_i) = \theta(\tilde{p}_i) \\ 1 & , else \end{cases}$$

The overall quality of a crawl c is then defined as:

$$q(c) = \frac{\sum_{i=1}^{n} f(\pi(p_i, \tilde{p}_i))}{n},\ n \geq 1$$

Given the previous definitions we are able to evaluate the quality of a crawl c. Since we intend to increase the overall quality, we examine the probability (and thus the risk) of crawling incoherent contents.

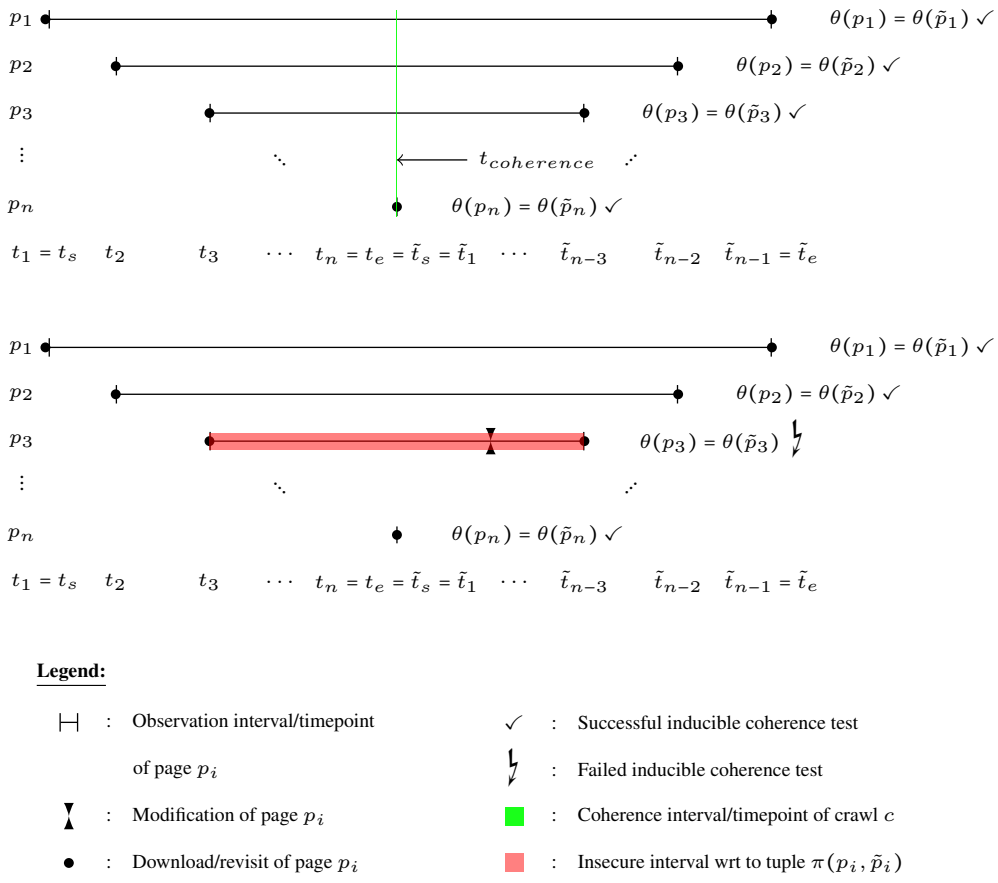

Figure 2.6: Inducible coherence fulfilled(top) and violation of inducible coherence (bottom)

2.3.4 Probability of Incoherence

The probability of a single page p_i being incoherent with respect to the reference time point or time interval $t_{coherence}$ is an important parameter to consider when scheduling a crawl. Incoherence occurs, when a page p_i is subject to one or more modifications μ_i' that are in "conflict" with the ongoing crawl, which is either based on measurable coherence (cf. section 2.3.3) or on inducible coherence (cf. section 2.3.5) with respect to the reference coherence time point $t_{coherence}$.

Conflict Probability in Measurable Coherence

A conflict of measurable coherence is given if:

$$\exists \mu_i' : \mu_i' \in [t_s, t(p_i)] \tag{2.1}$$

That means, a page has been modified at least once since the start of the crawl t_s and the time of downloading this page $t(p_i)$. Given a page's change probability λ_i and its download time $t(p_i)$, the probability of conflict $\kappa(p_i)$ is given as:

$$\kappa(p_i) = 1 - (1 - \lambda_i)^{t(p_i)-t_s} \tag{2.2}$$

Figure 2.8 shows in a graphical representation the pages of a Web site $p_1, \ldots, p_n$ (vertically) to be crawled spanning the crawl interval $[t_1, t_n]$ (horizontally). Given a crawl ordering from top to bottom of pages p_i to be downloaded, the diagram differentiates between those slots where a change of page p_i affects the coherence of crawl c and others that do not. The result is a set of concatenated slots – different in size – that represents (overall) the risk of a crawl being affected by changes. Even more, the length of each slot can be understood as the magnitude of the exponent of $\kappa(p_i)$ in equation 2.2. As can be easily seen, this risk of conflict exponentially increases with the time of download t_i.

Conflict Probability in Inducible Coherence

A conflict of inducible coherence occurs if:

$$\exists \mu_i' : \mu_i' \in [t(p_i), t(\tilde{p}_i)] \tag{2.3}$$

That means, a page has been modified at least once since its download during the crawling phase $t(p_i)$ and its revisit $t(\tilde{p}_i)$. Given a page's change probability λ_i, its download time $t(p_i)$ and its revisit time $t(\tilde{p}_i)$, the probability of conflict $\kappa(p_i)$ is given as:

$$\kappa(p_i) = 1 - (1 - \lambda_i)^{t(\tilde{p}_i)-t(p_i)} \tag{2.4}$$

Potentially conflicting slots in applying inducible coherence are shown in figure 2.7. In this example, a crawl ordering from top to bottom $(p_1, \ldots, p_n)$ and revisits from bottom to top $(p_{n-1}, \ldots, p_1)$ is being applied. Likewise in the previous case, the illustration differentiates between those slots where a change of page p_i affects the coherence of crawl c and others that do not. Again, the result is a set of concatenated slots – different in size – that represents (overall) the risk of a crawl being affected by changes. However, since the crawl c now takes twice the time, each periled slot is now double in size. That means, the necessity of applying inducible coherence increases the risk of conflict in the exponent of $\kappa(p_i)$ by a magnitude of 2 (cf. equation 2.4).

As a consequence, from the previous observations we can identify two factors that influence the potential incoherence of a page p_i with respect to the reference coherence time point $t_{coherence}$: Page p_i's change probability λ_i and its download (and revisit) time $t(p_i)$ (and $t(\tilde{p}_i)$). Hence, we will now introduce coherence optimized crawling strategies incorporating both factors.

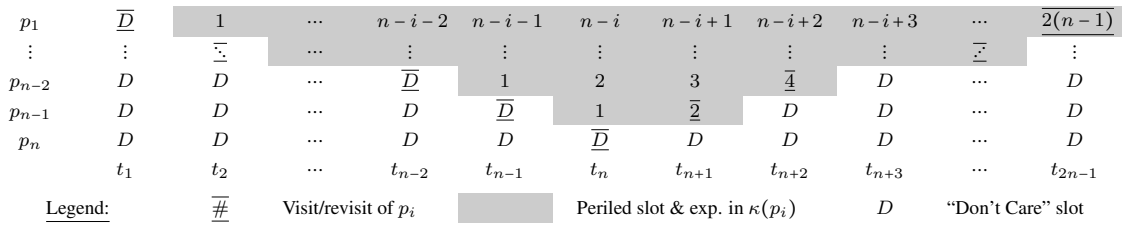

Figure 2.7: Periled slots in inducible coherence

2.4 Coherence Improved Crawling

Conventional archiving crawlers are based on a priority-driven variant of breadth-first-search (BFS) crawling. Even more, they do not incorporate any knowledge about Web sites that have been archived before. However, since information about change probabilities can be derived, our crawling strategy makes use of this "untapped" resource. In addition, there is no revisit concept for "virtual time stamping" in conventional implementations of archiving crawlers.

2.4.1 Crawling for Measurable Coherence

In case of archiving a Web site that provides precise time stamps of Web contents we apply our measurable coherence approach. In a first step, we compute the change probabilities λ_i of those pages p_i to be crawled. Pages are then sorted according to their change probability in descending order and sent into the crawling queue. Our aim is to find a schedule, which allows us to assign all slots (from small to large) according to the triangle-like shape in figure 2.8.

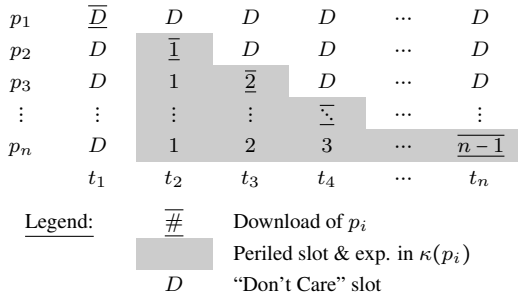

Figure 2.8: Periled slots in measurable coherence

Hence, we introduce a user definable (e.g. by the crawl engineer) threshold η, which allows to specify the readiness to assume risk encountering incoherence upon download. The threshold η is to this end evaluated against the conflict probability (cf. formula 2.2) of the

next page scheduled for download p_{slot}, in order to discard it if a "less risky" page exists in the schedule. We start with assigning the uncritical slot (as we assume that changes of Web pages might occur only per time unit immediately before download) with length 0 at the first position (p_{slot} = 1) to the most critical content ("joker" position). Given t_1, formula 2.2 becomes zero regardless of λ_i. However, from now on t increases stepwise so that any downloaded content bears the risk of having changed since the start of our crawl. In case, the currently assumed conflict probability is less than the given threshold ($\kappa(p_{slot}) < \eta$), the page p_{slot} is downloaded and the crawl is continued. But, if this condition is not fulfilled, we skip the page at p_{slot} for the moment being. Crawling then continuous until the n^{th} position of our queue is reached. Then we continue downloading those pages that have been skipped in the previous stage. In order to keep up a (depending on the predefined threshold η) slight chance of still downloading coherent contents, we now proceed downloading pages in reversed order until all n pages have been downloaded. A pseudo code implementation of the strategy described is shown in figure 2.9.

input: $p_1, \ldots, p_n$ - list of pages in descending order of λ_i,
η - readiness to assume risk threshold
begin
  Start with: $slot$ = 1
  while $slot \leq n$
  do
    if $\kappa(p_{slot}) < \eta$ **then** `/* no conflict expected */`
      Download the page p_{slot}
    end
    Continue with next iteration: $slot$ + +
  end
  Download skipped pages in reversed order of their index
end

Figure 2.9: Measurable coherence crawling

2.4.2 Crawling for Inducible Coherence

Due to the unreliability or non-existence of last modified stamps in most real life crawls, there is a need to ensure coherence based on inducible coherence. As outlined in section 2.3.5 this method is based on self created "virtual time stamps" by comparing the page's etag or content hash with its previously downloaded version in a three-stage process.

Starting point is a list of pages p_i to be crawled sorted in descending order according to their change probabilities λ_i. Like before, the intention is to identify those pages that might overstep the readiness to assume risk threshold η. Since now all pages need to be scheduled according to the reference time point $t_{reference} = t_n$ being the last page to be crawled during the crawling phase, we need a different queuing strategy: We try to create a V-like access

```
input : p_1, ..., p_n - list of pages in descending order of λ_i,
        η - readiness to assume risk threshold
begin
    Start with: slot = 1, last_promising = n
    while slot ≤ last_promising
    do
        if κ(p_slot) ≥ η then                    /* conflict expected!  */
            Move p_slot to position last_promising
            Decrease promising boundary: last_promising − −
        end
        else
            Increase promising boundary: promising + +
        end
    end
    slot = n while slot ≥ 1
    do                              /* visit from hopeless to promising */
        Download page p_slot
        Decrease slot counter: slot − −
    end
    slot = 2 while slot ≤ n
    do                            /* revisit from promising to hopeless */
        Revisit page p_slot
        Increase slot counter: slot + +
    end
end
```

Figure 2.10: Inducible coherence crawling

schedule having the (large) slots of stable pages on top and the (small) slots of instable ones at bottom (cf. figure 2.7). Again, we start with assigning the uncritical slot (as we assume that changes of Web pages might occur only per time unit immediately before download) with length 0 to the most critical content at the first position ($p_{slot} = 1$) of our queue. Since, initially, the length of the slot in the "joker" position (t_n) to be assigned is zero, the threshold condition does not hold. However, from now on t (and thus the size of slots) increases stepwise so that any download bears the risk of being incoherent. To this end, we evaluate the current page's conflict probability (cf. formula 2.4) against the user defined threshold ($\kappa(p_{slot}) \geq \eta$). As it is rarely possible to include all pages in this V-like structure, we split the download schedule into a promising section and a hopeless section. In case, the given threshold is exceeded we move the page at p_{slot} to the $last_{promising}$ position, which is the (at this point in time) the first position after those pages not exceeding the conflict threshold η. Otherwise, the page will be scheduled for download at p_{slot}. This process is continued until all pages p_i have been scheduled either in the promising section or the hopeless section.

In the next stage, the crawl itself starts. During the crawling phase, we begin with the most hopeless ones first until we continue with those pages that have been allocated in the promising section. After completion, we directly initiate the revisit phase in the reverse order. We begin with the first element after the "joker" position (p_{slot} = 2) until the revisit of the remaining pages has been completed. A pseudo code implementation of the strategy described is shown in figure 2.10.

2.5 Experimental Results

We now compare our approach toward coherence improved crawling with related strategies. Experiments were run on synthetic data in order to investigate the performance of versatile crawling strategies within a controlled test environment. In order to resemble real life conditions, we simulated small to medium size crawls of Web sites consisting of 10.000 – 50.000 contents. In addition, we simulated the sites' change behavior to vary from nearly static to almost unstable.

All experiments followed the same procedure, but varied in size of Web contents and change rate. Each page of the data set has a change probability λ_i in the interval $[0; 1]$. Within the simulation environment a change history was generated, which registered every change per time unit. The probability that page p_i changed at t_j is $P(\mu_i) = P[\chi(t_j) \leq \lambda_i]$ where $\chi(t_j)$ is a function that generates per time unit a uniformly distributed random number in $[0; 1]$.

As mentioned before, conventional implementations of archiving crawlers are based on a breadth-first-search (BFS) crawling strategy and do not incorporate revisits. However, "virtual time stamping" is unavoidable in order to determine coherence under real life crawling conditions. Therefore, we compare our coherence improved crawling strategy based on inducible coherence with crawl revisit pairs based on BFS-LIFO (last in, first out) as well as BFS-FIFO (first in, first out). In addition, we indicate baselines for optimal and worst case crawling strategies, which are obtained from full knowledge about changes within all pages p_i during the entire crawl-revisit interval. Hence, these baselines are only considerable as theoretical achievable limits of coherence.

Figure 2.11 depicts the results of our improved inducible crawling strategy compared with its "competitors" BFS-LIFO and BFS-FIFO. Our improved crawling strategy always performs better than the best possible conventional crawling strategy. Experiments are based on a Web site containing 10.000 contents and different readiness to assume risk thresholds η ranging from $[0.45; 0.7]$. In addition, our strategy performs about 10% better given non-pathological Web site behaviour (neither completely static nor almost unstable). Values of η between $[0; 0.45)$ or $(0.7; 1]$ perform less effective. They induce an either too "risk-avoidant" ($\eta \in [0; 0.45)$) or too "risk-ignorant" ($\eta \in (0.7; 1]$) scheduling with minor (or even zero) performance gain, e.g. when acting "risk-ignorant" in heavily changing sites or "risk-avoidant" in mostly static sites.

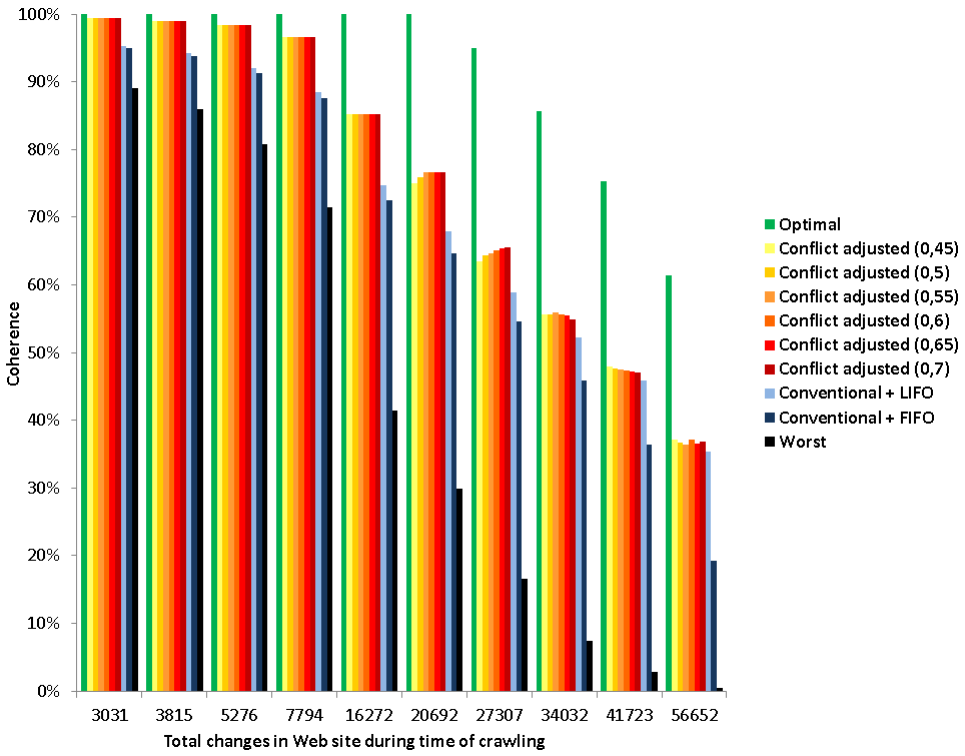

Figure 2.11: Comparison of inducible crawling strategies in a Web site with 10.000 contents

Comparable results have also been produced given larger (and smaller) Web sites having similar change distributions in numerous experiments. In addition, our strategy introduced to improve measurable coherence shows similar performance, considering a by factor 2 decreased exponent of $\kappa(p_i)$.

2.6 Related Work

Related research is focused on improving the efficiency of crawlers for Web indexing in search engines or crawler development in general. The contribution of Pant et al. [PSMe04] describes the technical process involved in Web crawling. A quite technical overview on the Heritrix crawler is given by Mohr et al.. However, they do not address the issue of coherence [MKSR04]. B. E. Brewington and G. Cybenko [BrCy00] analyze changes of Web sites and draw conclusions about how often they must be reindexed. The issue of crawl efficiency is addressed by Cho et al. [CGPa98]. They state that the design of a good crawler is important for many reasons (e.g. ordering and frequency of URLs to be visited) and present an algorithm that obtains more relevant pages (according to their definition) first. In a subsequent study Cho and Garcia-Molina describe the development of an effective incremental crawler [ChGa00]. They aim at improving the collection's

freshness by bringing in new pages in a more timely manner. Into the same direction head their studies on effective page refresh policies for Web crawlers [ChGa03a]. Here, they introduce a poisson process based change model of data sources. In another study, they estimated the frequency of change of online data [ChGa03b]. For that purpose, they developed several frequency estimators in order to improve Web crawlers and Web caches. In a similar direction goes research of Olston and Pandey [OlPa08] who propose a recrawl schedule based on information longevity in order to achieve good freshness. Another study about crawling strategies is presented by Najork and Wiener [NaWi01]. They have found out that breadth-first search downloads hot pages first, but also that the average quality of the pages decreases over time. Therefore, they suggest performing strict breadth-first search in order to enhance the likeliness to retrieve important pages first. Research on improving the scalability of a Web crawler in order to crawl 6 billion pages and beyond is presented by Lee et al. [LLWL08]. Their findings show that changing the BFS crawling order and designing low-overhead disk-based data structures increase the efficiency of large-scale crawlers. A dedicated survey about the evolution and dynamics of wikis as social networks is done by Klamma and Haasler [KlHa08]. Interesting in this paper is the disclosure of social networks based on the hierarchical structure of important and unimportant nodes.

2.7 Summary

Data quality in Web archiving was and – with the advent of Web 2.0 technologies – is an important issue in order to preserve our digital culture. As we have figured out, temporal coherence in Web archiving is a key issue in order to capture digital contents in a reproducible and, thus, later on interpretable manner [SMDW09]. To this end, we have given an overview on strategies that help to overcome (or at least identify) the temporal diffusion of Web crawls that last from a view hours only up to several days [SDM*09]. Based on the coherence framework introduced, we have shown how to schedule crawls in order to reduce the risk of crawling contents being incoherent [DMSW09, DMSW11, MDSW10]. Even more, our experimental results have shown that we are able to improve the data quality in Web archiving by around 10% for non-pathological (neither completely static nor almost unstable) Web sites.

Chapter 3

Entities on the Web

Web contents such as news, blogs and other social media are full of named entities. Each entity belongs to one or more *semantic types* associated with it. Tracking entities on the Web or in Web archives involves finding names of people, companies, products, songs, etc. in Web pages and social media. However, names are often ambiguous. For that purpose, mentions of people, places, or organizations need to be raised to the entity level. This entity information is a great asset for making sense of the raw and often noisy data.

3.1 Entity Disambiguation

Disambiguation of named entities in natural language text needs to map mentions of ambiguous names onto canonical entities in order to allow a semantically exploitation of Web data. To this end, we map mentions of ambiguous names onto canonical entities like people or places, registered in a knowledge base such as DBpedia [ABK*07] or YAGO [SKWe07, HSB*11, HSBW13].

3.1.1 Conceptual Approach

In order to disambiguate text mentions onto canonical entities, we have developed the AIDA (Accurate Online Disambiguation of Named Entities) system [HYB*11, YHB*11]. AIDA is a robust framework centered around collective disambiguation exploiting the prominence of entities, similarity between the context of the mention and its candidates, and the coherence among candidate entities for all mentions. The underlying method unifies the before mentioned approaches into a comprehensive framework that combines three measures: the prior probability of an entity being mentioned, the similarity between the contexts of a mention and a candidate entity, as well as the coherence among candidate entities for all mentions together. Key contributions compared to the previously published work are spatial and temporal extensions for improved named entity extraction and disambiguation.

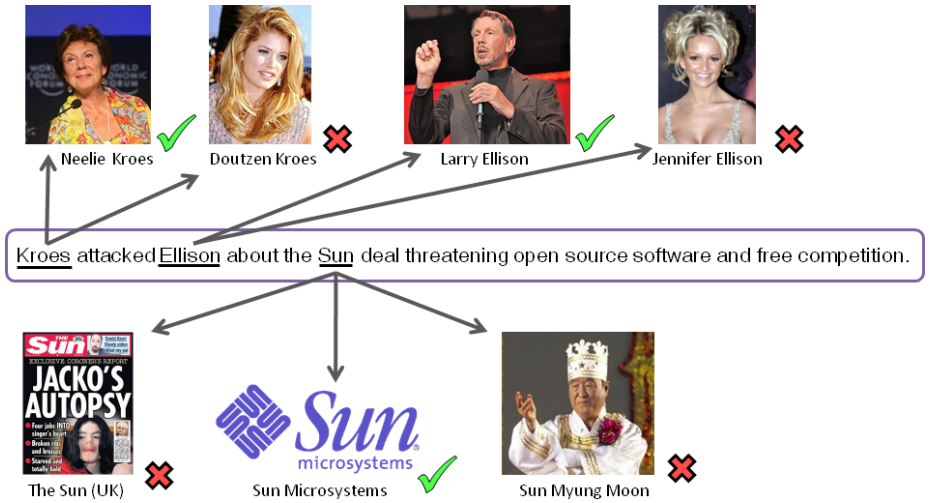

Figure 3.1: Example of entity disambiguation

We aim to identify surface strings representing named entities and map them to their proper entries in a knowledge base, thus giving a disambiguated meaning (cf. figure 3.1 for a graphical example of entity disambiguation). Given the entity candidates of the surface strings extracted with the help of Stanford NER Tagger [FGMa05], we now aim to disambiguate them using coherence. Given the previous example and surface strings "Kroes", "Ellison" and "Sun" extracted with the help of Stanford NER Tagger [FGMa05], we now aim to disambiguate the text mentions using coherence by combining three different disambiguation measures:

(a) *Prior*: How often did other entities link to this entity in Wikipedia?

(b) *Similarity*: How good do entity keyphrases and the text context overlap?

(c) *Coherence*: Are the disambiguated entities related?

Since, each entity has a context in the underlying knowledge base(s): other entities that are connected via semantic relationships (e.g., memberOf) or have the same semantic type (e.g., politician). An asset that knowledge bases like DBpedia and T-YAGO [WZQ*10, WYZ*11] provide us with is the same-as cross-referencing to Wikipedia. This way, we can quantify the coherence between two entities by, e.g., the overlap among their related entities or some form of type distance. In addition, we have exploited the spatial and temporal information to better disambiguate named entities. The spatial distance between two named entities with geo-coordinates is defined as the normalized great circle distance, while the temporal coherence of two named entities is defined as the difference of the center points of the entity's existence time interval, normalized by the maximum distance of any two entities in the current set of entity candidates.

3.1.2 Framework and Algorithms

The input to AIDA is an arbitrary text, optionally with HTML or XML markup or in the RDF N3 form, with mentions of named entities (people, music bands, songs, universities, etc.). The goal is to find the correct mapping for the mentions onto canonical entities in a knowledge base (currently YAGO).

Mentions are automatically detected using the Stanford NER Tagger[1]). We first identify noun phrases (e.g., "Larry Page", "Apple", "Chief Seattle", "Dances with Wolves", etc.) that potentially denote named entities. For possible entities (with unique canonical names) that a mention could denote, we harness existing knowledge bases. For each entity they provide a set of short names (e.g., "Apple" for `Apple Inc.` and paraphrases (e.g., "Big Apple" for `New York City`). In YAGO, these are available by the `means` relation, which in turn is harvested from Wikipedia disambiguation pages, redirects, and links.

Popularity Prior for Entities

Prominence or popularity of entities can be seen as a probabilistic prior for mapping a name to an entity. The most common way of estimating this are the Wikipedia-based frequencies of particular names in link anchor texts referring to specific entities, or number of inlinks.

Mention-Entity Graph

For collective mapping, we use a graph-based approach. The graph is constructed with mentions and their candidate entities as nodes. We have two types of edges:

- **Mention-entity edges:**
  between mentions and their candidate entities with weights that capture the similarity between the context of a mention and a candidate;
- **Entity-entity edges:**
  between different entities with weights that capture the coherence (semantic relatedness) between two entities.

Our goal is to reduce this graph to a dense sub-graph where each mention node is connected to one and only one candidate entity node, which provides our output mapping. Density here refers to the total weight of the sub-graph's edges, or alternatively, to the minimum weighted degree in the sub-graph. Once the graph is constructed, we use a greedy algorithm to compute the sub-graph. In each iteration, we perform two steps:

[1] http://nlp.stanford.edu/software/CRF-NER.shtml

- identify the entity node that has the lowest weighted degree (sum of the weights of the node's incident edges), and
- remove this node and its incident edges from the graph unless it is the last remaining candidate entity for one of the mentions.

Figure 3.2 illustrates the mention-entity graph for an input text with highlighted mentions (left) and candidate entities (middle) based on a knowledge base (right). The thickness of edges between entities depicts different edge weights. Next, we describe the features and measures for computing the edge weights.

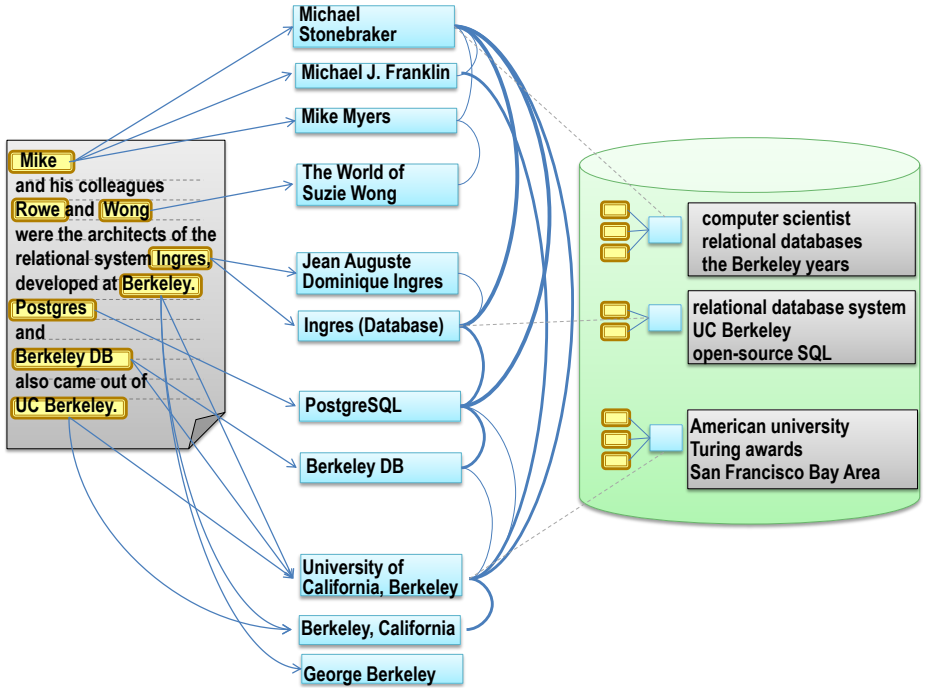

Figure 3.2: Example of an Mention-Entity Graph

The *similarity* between a mention and a candidate entity is computed as a linear combination of two ingredients. The first one is the prominence of an entity, e.g., Prince Harry of England vs. Harry Kelly, a lesser known American basketball player. This acts as a prior probability for each potential mapping. We compute this prior by collecting statistics on href anchor texts and their link targets in Wikipedia. The second ingredient for the mention-entity edge weights is based on the overlap between a mention's context and a candidate entity's context. For a mention we consider the full input text as its context. For entities, we consider *entity keyphrases*, pre-computed from the Wikipedia articles that YAGO's entities connect to. We define the notion of keyphrases to be all phrases in link anchors, including category names, citation titles, and external references in the entity article. We extended this further by

considering also the titles of incoming links for an entity's article, as an additional source of describing the entity.

While keyphrase overlap between the contexts of a mention and entity is an expressive similarity measure, we will rarely find perfect matches for multi-word keyphrases in the input text. Consider entity `Manchester United`, which has keyphrases such as "2008 UEFA Champions League Winner" But the input text may say "winner of the champions league in 2008", not nearly a full match of the keyphrase. Therefore, we devised a partial-match model to improve coverage. To avoid degrading accuracy, we consider the size of the window that covers all words of the keyphrase that appear in the input text. For example, the above wording in the text matches 4 of the 6 words of the keyphrase within a window of size 7. Moreover, we penalize the keyphrases that occur in the text by their distance from the mention under consideration. Obviously, once we allow partial matches, different words in a phrase have different degrees of importance. To accommodate this aspect, we collect statistics from a large corpus (e.g., Wikipedia) about the co-occurrence frequency of a word an the entity of interest. We use the Mutual Information (MI) measure (aka. relative entropy) to quantify the specificity of a word for an entity. These values serve as per-word weights for scoring the partial matches in the input text. The scores for all matches are aggregated by summation with distance decay.

For the *coherence* weights of entity-entity edges, we harness the Wikipedia link structure. We define the coherence between two entities to be proportional to the number of incoming links that are shared between their Wikipedia articles [MiWi08]. For the dense sub-graph that yields the final disambiguation, we expect the final candidate entities of different mentions to be mutually connected by high edge weights.

To aim for the best disambiguation mappings, our framework combines prior, similarity, and coherence measures into a combined objective function: for each mention m_i, $i = 1..k$, select entity candidates e_{j_i}, one per mention, such that

$$\alpha \cdot \sum_{i=1..k} \text{prior}(m_i, e_{j_i}) +$$

$$\beta \cdot \sum_{i=1..k} \text{sim}(\text{cxt}(m_i), \text{cxt}(e_{j_i})) +$$

$$\gamma \cdot \text{coh}(e_{j_1} \in \text{cnd}(m_1) \ldots e_{j_k} \in \text{cnd}(m_k)) = \max!$$

where $\alpha + \beta + \gamma = 1$, $\text{cnd}(m_i)$ is the set of possible meanings of m_i, cxt( ) denotes the context of mentions and entities, respectively, and coh( ) is the coherence function for a set of entities. More details on the features and algorithms of this approach are included in [HYB*11].

3.1.3 Related Work

Recognizing named entities (NER tagging) in natural-language text has been extensively addressed in NLP research. The output is labeled noun phrases. However, these are not yet canonical entities, explicitly and uniquely denoted in a knowledge repository. Approaches that use Wikipedia for explicit disambiguation date back to [BuPa06] and have been further pursued by [Cuce07, HaZh09, MiWi08, NgCa08, MiCs07]. [BuPa06] defined a similarity measure that compared the context of a mention to the Wikipedia categories of an entity candidate. [Cuce07, MiWi08, NgCa08] extended this framework by using richer features for the similarity comparison. [MiWi08] additionally introduced a supervised classifier for mapping mentions to entities, with learned feature weights rather than using the similarity function directly. Further, they introduced a notion of semantic relatedness between a mention's candidate entities and the unambiguous mentions in the textual context. The relatedness values are derived from the overlap of incoming links in Wikipedia articles. [HaZh09] considered another feature: the relatedness of common noun phrases in a mention's context, matched against Wikipedia article names. While these features point towards semantic coherence, the approaches are still limited to mapping each mention separately. Nonetheless, this line of feature-rich similarity-driven methods achieved very good results in experiments, especially for the task of predicting Wikipedia link targets for a given href anchor text. On broader input classes such as news articles (called "wikification in the wild" in [MiWi08]), the precision was reported to be about 75%.

The first work with an explicit collective-learning model for joint mapping of all mentions has been [KSRC09]. This method starts with a supervised learner for a similarity prior, and models the pair-wise coherence of entity candidates for two different mentions as a probabilistic factor graph with all pairs as factors. The MAP (maximum a posteriori) estimator for the joint probability distribution of all mappings is shown to be an NP-hard optimization problem, so that [KSRC09] resorts to approximations and heuristics like relaxing an integer linear program (ILP) into an LP with subsequent rounding or hill-climbing techniques. Their experiments show that this method is superior to the best prior approaches, most notably [MiWi08]. However, even approximate solving of the optimization model has high computational costs.

Coreference resolution is the task of mapping mentions like pronouns or short phrases to a preceding, more explicit, mention. Recently, interest has arisen in cross-document coreference resolution [MAD*09], which comes closer to named entity disambiguation (NED), but does not aim at mapping names onto entities in a knowledge base. Word sense disambiguation [McCa09, Navi09] is the more general task of mapping content words to a predefined inventory of word senses. While the NED problem is similar, it faces the challenges that the ambiguity of entity names tends to be much higher (e.g., mentions of common lastnames or firstname-only).

Projects on automatically building knowledge bases [DGRV08] from natural-language text include KnowItAll [BCS*07], YAGO and its tool SOFIE [SSWe09, NTWe11], StatSnow-

ball [ZNL*09], ReadTheWeb [CBK*10], and the factor-graph work by [WCRC09]. Only SOFIE maps names onto canonical entities; the other projects produce output with ambiguous names. SOFIE folds the NED into its MaxSat-based reasoning for fact extraction. This approach is computationally expensive and not intended for online disambiguation.

3.2 Entity Type Classification

Named entity classification (NEC) addresses the issue of assigning proper types to entity mentions. In contrast to named entity disambiguation (NED), this also addresses the classification of out of knowledge base entities, such as "emerging" (e.g. Taifun Haiyan) or "unknown" (e.g. Marc Spaniol) entities. To this end, inferring lexical type labels of entity mentions in texts is an important asset for NLP tasks like semantic role labeling and support of NED. Prior work has so far focused on flat and relatively small type systems where most entities belong to exactly one type. However, each entity belongs to one or more *semantic types* associated with it. For instance, noun phrases such as "songwriter Dylan", "Google founder Page", or "rock legend Page" can be easily mapped to the entities Bob Dylan, Larry Page, and Jimmy Page if their respective types `Singer`, `BusinessPerson`, and `Guitarist` are available (cf. Figure 3.2 for an illustrative example).

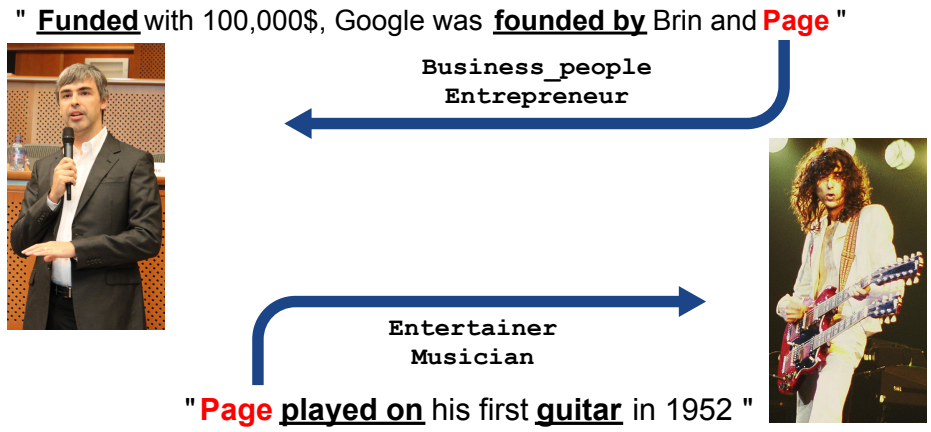

Figure 3.3: Fine-grained entity type classification

As shown by the previous example, type classification is not only based on hierarchical sub-type relationships (e.g. `Musician isA Person`), but also entails multi-labeling. Within a very fine-grained type hierarchy, many entities naturally belong to multiple types. For example, a guitarist is also a musician and a person, but may also be a singer, an actor, or even a politician. Consequently, entities should not only be assigned the most (fine-grained) label associated to them, but with all labels relevant to them. So we face a *hierarchical multi-label classification* problem [TZZh12]. To this end, we developed the HYENA (Hierarchical tYpe classification for Entity NAmes) system in order to automatically assign input text

PERSON	LCOATION	ORGANIZATION	EVENT	ARTIFACT
contestant	location	institution	social event	instrumentality
athlete	region	unit	act	medium
player	geographical area	company	show	structure
leader	district	educational institution	movie	creation
entertainer	administrative district	school	activity	product
intellectual	tract	association	contest	album
performer	site	musical organization	diversion	facility
communicator	point	club	action	building
scholar	structure	enterprise	group action	movie
alumnus	geographic point	secondary school	change	station

Table 3.1: Top 10 Subtypes of the 5 Top-Level Types

with very fine-grained types organized in a hierarchical taxonomy, with several hundreds of types at different levels [YBH*12, YBH*13].

3.2.1 Fine-grained Type Hierarchy

We have systematically derived a very fine-grained type taxonomy from the YAGO knowledge base [SKWe07, HSBW13] which comes with a highly accurate mapping of Wikipedia categories to WordNet synsets [Fell98]. We start with five broad classes namely `PERSON`, `LOCATION`, `ORGANIZATION`, `EVENT` and `ARTIFACT`. Under each of these superclasses, we pick 100 prominent subclasses. The selection of subclasses is based on the population of the classes: we rank them in descending order of the number of YAGO entities that belong to a class, and pick the top 100 for each of the top-level superclasses. We then connect subclasses to all of their valid parents (if they have multiple hypernyms). Note that if a subclass belongs to more than one superclass, it is only counted once. This implies that there might actually be (slightly) more than 100 classes per superclass (due to hypernyms). This results in a very fine-grained reference taxonomy of 505 types, organized into a directed acyclic graph with 9 levels in its deepest parts. For instance, this includes fine-grained classifications of an `Adminstrative District` in order to distinguish between `Municipality`, `Township`, `Commune`, etc. or differentiations of `Publications` into `Books`, `Periodicals` and `Magazines`. Table 3.1 lists the top 10 subtypes under each of the 5 top-level types.

We are not aware of any similarly rich type hierarchies used in prior work on NER and entity typing. While the classes are picked from the YAGO type system, the approach is generic and can be applied to plug in alternative type taxonomies (e.g. derived from Freebase or DBpedia as in [LiWe12], or from hand-crafted resources such as WordNet).

3.2.2 Feature Set

For a general approach and for applicability to arbitrary texts, we use only features that are automatically extracted from input texts. We do not use any features that require manual annotations, such as sense-tagging of general words and phrases in training documents. This discriminates our method from some of the prior work which used WordNet senses as features (e.g., [RaNg10]). In the following, we briefly discuss each group of features and how they are derived. Table 3.2 summarizes our feature set.

Mention String: We derive the mention string (a noun phrase of one or more consecutive words) as well as unigrams, bigrams, and trigrams that overlap with the mention string.

Sentence Surrounding Mention: We derive from a bounded window (size 3) around the mention: all unigrams, bigrams, and trigrams in the sentence along with their distance to the mention, and all unigrams along with their absolute distance to the mention.

Mention Paragraph: We consider the mention paragraph in order to obtain additional topical cues about the mention type. We extract unigrams, bigrams, and trigrams in a bounded window (2000 characters) around the mention (truncated at the paragraph boundaries).

Grammatical Features: We use part-of-speech tags (with/without distance) of the tokens within a bounded window. Further, we resolve the first "he" or "she" pronoun in the same and in the subsequent the closest preceding verb-preposition pair.

Gazetteer Features: We build type-specific gazetteers of words occurring in entity names derived from the YAGO knowledge base. YAGO has a dictionary of name-entity pairs extracted from Wikipedia. We construct a binary feature whether the mention contains a word in this type's gazetteer or not. This does not mean determining the mention type(s) (e.g. "Alice" occurs in person subclasses but also in locations, songs, organizations, etc.).

3.2.3 Classifier

Hierarchical Classifier

Based on the feature set defined in the previous section, we build a set of type-specific classifiers using the SVM software liblinear [FCH*08, ChLi11]. As our YAGO-based type system integrates WordNet and Wikipedia categories, we obtain ample training data from Wikipedia effortlessly, by following the anchor texts to the corresponding YAGO entities.

For each type, we consider Wikipedia mentions (and their context, cf. Section 3.2.2) of the type's instances as positive training samples. For discriminative learning, we use all siblings in the type hierarchy as negative samples. That is, the classifier considers one type against

Input	Derived Features
Mention String	MENTION UNIGRAM_MENTION BIGRAM_MENTION TRIGRAM_MENTION
Mention Sentence	UNIGRAM_REL UNIGRAM_ABS BIGRAM_REL TRIGRAM_REL
Mention Paragraph	PARA_UNIGRAM PARA_BIGRAM PARA_TRIGRAM
Grammatical Features	POS Stanford NER Tagger FIRST_PRP_HE_SHE_SAME_SENT_AFT_MENTION_REL FIRST_PRP_HE_SHE_NEXT_SENT_REL LAST_VERB_PREP_TUPLE_BEF_MENTION
Gazetteer Features	OCCURS_TYPE1_WORDS OCCURS_TYPE2_WORDS ...

Table 3.2: Summary of Features Used for Classification

all other types that have the same parent type (e.g., Artist vs. Politician, Athlete, etc. – all under the same parent). As the subclasses of type t do not necessarily cover all entities, we add a subclass `Others` to each non-leaf type. Positive samples for `Others` are instances of type t that do not belong to any of its subclasses. Conversely, the classifiers for non-leaf nodes include all instances of their subtypes as positive samples (with full weight). HYENA performs type-specific classification in a top-down manner. A mention is assigned to all types for which the classifier signals acceptance. If rejected, classification stops at this level.

Meta Classifier

HYENA uses a global threshold θ for accepting to a class. Using a single parameter for all types is not fully satisfying, as different types may exhibit very different characteristics. So the optimal acceptance threshold may be highly type-dependent. To overcome this limitation, we devised a meta classifier that ranks the types for each test mention by decreasing confidence values and then predicts the "right" number of top-n labels to be assigned to a mention, similar to the methodology of [TRNa09]. We use the confidence values of the type-specific classifier ensemble as meta-features, and train a multi-class logistic regression classifier to obtain a suitable value n of features. We combine the base classifiers and the meta classifier by first running the entire ensemble top-down along the type hierarchy, and then letting the meta model decide on how many of the highest-scoring types we accept for a mention.

3.3 Experiments

In the following subsections we describe the evaluation of HYENA. First, we will introduce the experimental setup. Then we will describe our experiments on different ground truth datasets geared for high precision and high recall. Afterwards we will introduce results based on employing a meta-classifier in order to improve precision for quality-sensitive use cases. Finally, we analyze the impact of various features employed by HYENA.

3.3.1 Setup

System: The described methods are implemented in HYENA. The Stanford NLP tools are used to identify mentions of named entities and to extract grammatical features from the context. We used the YAGO2 knowledge base to construct gazetteer features.

Data: We used the English Wikipedia edition as of 2012-05-02. In order to obtain ground-truth type labels, we exploited the links to other Wikipedia articles, resolved the corresponding YAGO2 entity and retrieved the semantic types. For example, from the Wikipedia markup:

> "In June 1989, Obama met [[Michelle Obama|Michelle Robinson]] when he was employed as a summer associate at the Chicago law firm of [[Sidley Austin]]"

the following YAGO2 entities are assigned:

> Michelle Robinson → http://yago-knowledge.org/resource/Michelle_Obama
>
> Sidley Austin → http://yago-knowledge.org/resource/Sidley_Austin

HYENA is trained on 50,000 randomly Wikipedia articles selected, containing around 1.6 million entity mentions. 92% of the corresponding entities belong to at least one of our 5 top-level types, with 11% belonging to at least two top-level types. Testing of HYENA is performed on 10,000 randomly selected Wikipedia articles withheld from the same Wikipedia edition and disjoint from the training data. Properties of training and test data are summarized in Table 3.3. All experimental data is available at http://www.mpi-inf.mpg.de/yago-naga/hyena.

Performance Measures: We report micro- and macro-evaluation numbers for of our approach for precision, recall and F1 scores. To this end, we define the measures as follows:

Let T be the set of all types in our hierarchy, and let I_t be the set of instances tagged with type t, and let $\hat{I}_t$ the set of instances that are predicted to be of type t. Micro-evaluation measures used are:

data property	training	testing
# of articles	50,000	10,000
# of instances (all types)	1,613,340	253,029
# of location instances	489,003 (30%)	86,936 (34.4%)
# of person instances	426,467 (26.4%)	62,446 (24.6%)
# of organization instances	219,716 (13.6%)	38,293 (15.1%)
# of artifact instances	204,802 (12.7%)	31,899 (12.6%)
# of event instances	176,549 (10.9%)	28,952 (11.4%)
# instances in 1 top-level class	1,131,994 (70.2%)	179,240 (70.8%)
# instances in 2 top-level classes	182,508 (11.3%)	33,399 (13.2%)
# instances in more than 2 top-level classes	6,492 (0.4%)	828 (0.3%)
# instances not in any class	292,346 (18.1%)	39,562 (15.6%)

Table 3.3: Properties of Training and Testing Data

$$Precision_{micro} = \frac{\sum_{t \in T} \left| I_t \cap \hat{I}_t \right|}{\sum_{t \in T} \left| \hat{I}_t \right|} \quad and \quad Recall_{micro} = \frac{\sum_{t \in T} \left| I_t \cap \hat{I}_t \right|}{\sum_{t \in T} |I_t|}$$

and macro-evalution measures are:

$$Precision_{macro} = \frac{1}{|T|} \sum_{t \in T} \frac{\left| I_t \cap \hat{I}_t \right|}{\left| \hat{I}_t \right|} \quad and \quad Recall_{macro} = \frac{1}{|T|} \sum_{t \in T} \frac{\left| I_t \cap \hat{I}_t \right|}{|I_t|}$$

Competitors: From the related work in Section 3.3.6 we identified those prior methods that target fine-grained, multi-level type classification and used publicly available corpora on which we could run HYENA for direct comparison. These are the methods of [FlHo02] referred to as *HOVY*), [RaNg10] referred to as *NG*), and *FIGER* by [LiWe12]. We preferred experiments on the competitors' datasets to avoid re-implementation and to give our opponents the benefit of their original optimization and tuning.

3.3.2 Multi-label Classification

We present multi-label experiments that are geared for high precision and high recall. Experiments are performed against ground truth coming from Wikipedia, the BBN Pronoun Coreference Corpus and Entity Type Corpus (LDC2005T33)[2] and the FIGER-Gold dataset. When applying HYENA to a different dataset than Wikipedia, we present results for HYENA configurations adopted for those settings as well.

[2]http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC2005T33

	Macro			Micro		
	Prec.	Rec.	F1	Prec.	Rec.	F1
5 Top-level Types	0.941	0.922	0.932	0.949	0.936	0.943
All 505 Types	0.878	0.863	0.87	0.913	0.932	0.922

Table 3.4: Overall Experimental Results for HYENA on Wikipedia 10000 articles

		Macro			Micro		
		Prec.	Rec.	F1	Prec.	Rec.	F1
5 Top-level Types	*HOVY*	0.522	0.464	0.491	0.568	0.51	0.537
	HYENA	**0.941**	**0.922**	**0.932**	**0.949**	**0.936**	**0.943**
All 505 Types	*HOVY*	0.253	0.18	0.21	0.405	0.355	0.378
	HYENA	**0.878**	**0.863**	**0.87**	**0.913**	**0.932**	**0.922**

Table 3.5: Results of HYENA vs *HOVY* (trained and tested on Wikipedia 10000 articles)

HYENA experiments on Wikipedia

HYENA is trained on a randomly selected set of 50,000 Wikipedia articles, containing around 1.6 million entity mentions. Testing of HYENA was performed on 10,000 randomly selected Wikipedia articles, withheld from the same Wikipedia edition and disjoint from the training data. The results of our HYENA approach on Wikipedia are shown in Table 3.4. HYENA achieves very high F1 scores of around 94% for its 5 top-level types. Evaluated against the entire hierarchy, F1 scores are still remarkably high with F1 scores of 87% and 92% for macro and micro evaluations, respectively. The slightly weaker results for the macro evaluation are explainable by our fine-grained hierarchy, which also contains a few "long-tail" types. However, the overall micro results show that these types contain relatively instances only.

In order to compare against *HOVY*, we emulated their method within the HYENA framework. This is done by specifically configuring the feature set, and using the same training and testing instances as for HYENA. Results are shown in Table 3.5. HYENA significantly outperforms *HOVY*. Similar to the results reported in [FlHo02] *HOVY* shows decent performance for the 5 top-level types, but performance sharply drops for subtypes at deeper levels. However, it becomes obvious that *HOVY* has not been designed for such a fine-grained type hierarchy as its performance sharply drops for more subtypes at deeper levels. Reasons for this behavior can be explained with the greater number of features and surrounding context considered in HYENA, as well as the lack of gazetteers in *HOVY*.

	Macro			Micro		
	Prec.	Rec.	F1	Prec.	Rec.	F1
FIGER	**0.75**	0.743	0.743	**0.828**	**0.838**	**0.833**
HYENA	0.745	0.631	0.684	0.815	0.645	0.72
HYENA (at least one tag)	0.724	**0.801**	**0.75**	0.788	0.814	0.801

Table 3.6: Results of HYENA vs *FIGER* (trained on Wikipedia and tested on FIGER-Gold)

HYENA Experiments on FIGER-GOLD

The FIGER-GOLD dataset consists of 18 news reports from a university website, as well as local newspapers and specialized magazines [LiWe12]. The test dataset was annotated with at least one label per mention. This resulted in a total of 434 sentences with 563 entities having 771 labels coming from 42 out of the 112 types. The original evaluation for FIGER was instance-based. In order to compare against HYENA, a per-type evaluation is needed. To this end, we created a per-type based classification of FIGER based on their output data. Since the distribution of mentions on different types in the FIGER dataset is heavily skewed (e.g. 217 of the 562 entities are of type `PERSON` without finer-grained subtype annotation) we cover in our evaluation the most 10% populated classes (covering around 70% of the tags). These classes were then mapped onto the hierarchy of HYENA. Since all instances in the FIGER-GOLD dataset are tagged with at least one class, we ran HYENA in two configurations: without any modification as before (using a classifier trained to deal with abstract concepts, e.g. `Chinese Philosophy`, that are of generic type `ENTITY_OTHER`) as well as by enforcing the assignment of at least one class for all instances (referred to as "at least one tag").

Results are shown in Table 3.6. In the standard configuration, HYENA shows precision scores close to *FIGER*. However, HYENA suffers from the training against abstract concepts. In the second configuration, both systems achieve results in the same range with slight advantages for *FIGER* on micro-average and overall better results of HYENA on macro-average. The overall 10% drop of HYENA's performance compared with the experiments on Wikipedia are due to the nature of the FIGER-GOLD dataset, which comes with short sentences so that context features of HYENA are not that effective. However, 771 type labels for 562 entity mentions (not entities) is only a very moderate amount of multi-label classification. This is disadvantageous for HYENA, which has been designed for data where the number of labels per mention is higher.

HYENA Experiments on BBN

The BBN Pronoun Coreference and Entity Type Corpus consists of 2311 manually annotated documents. Since *NG* exploits WordNet word-senses for disambiguation, the corpus is restricted to those 200 documents (160 training, 40 testing) that have corresponding

	Macro			Micro		
	Prec.	Rec.	F1	Prec.	Rec.	F1
NG (trained on BBN)	0.859	0.864	0.862	0.812	0.871	0.84
HYENA (trained on Wikipedia)	**0.943**	0.406	0.568	**0.932**	0.371	0.531
HYENA (trained on Wikipedia, at least one tag)	0.818	0.671	0.737	0.835	0.632	0.719
HYENA (trained on BBN)	0.916	**0.909**	**0.911**	0.919	**0.881**	**0.899**

Table 3.7: Results of HYENA vs *NG* (tested on BBN Corpus)

annotations. For comparison against *NG* we performed a mapping onto the hierarchy of HYENA. Among the 16 types for the *NG* dataset (cf. [RaNg10]), there are 8 non-entity types (e.g. `Date`) and 5 descriptor types (`_DESC`) which cannot be mapped. This resulted in mapping the 3 top-level types: `Person`, `Organization` and `GPE` (country, city, states, etc.). Similar to the FIGER-GOLD dataset, there are no unclassified mentions in the BBN corpus. Hence we ran HYENA in three configurations: standard ("trained on Wikipedia"), enforcing at least one type label to be assigned ("trained on Wikipedia, at least one tag") and HYENA trained on the *NG* training set ("trained on BBN").

Results on the BBN dataset exhibit high precision of HYENA already with its standard configuration (cf. Table 3.7). However, it suffers from low recall in this setting, due to training against abstract concepts and not assigning any type. When enforcing HYENA to assign at least one tag, F1 scores strongly improve. In the third configuration, the fairest side-by-side comparison, we clearly outperform *NG* which shows the performance of HYENA when trained on a data set of more similar text style.

3.3.3 Meta-Classification

In application use-cases for type labeling (e.g. NED), precision is often more important than recall. This is particularly demanding for types that suffer from data sparsity (less prominent and/or less populated types) deep in the type hierarchy. For example in NED, it may be crucial to distinguish a `Painter` from a `Musician`. In order to improve the precision of fine-grained type labeling, we applied a meta-classifier as described in Section 3.2.3. The meta-classifier adjusts, per level, the threshold for the number of types that an individual mention should have. Typically this results in a more conservative behavior of the classifier. When applied, meta classification (see Section 3.2.3 for details) improves macro-precision over all 505 types by more than 1% (cf. Table 3.8). When focusing on the 5% types that performed worst without it, we even gained more than 2% in precision, as shown in Table 3.9. The top-5 winners in this group are (in ascending order) `Tour`, `Pageant`, `Presentation`, `Battalion` and `Ghost Town` with performance gains ranging from 5% up to 13%.

		Macro			Micro		
		Prec.	Rec.	F1	Prec.	Rec.	F1
All 505 Types	HYENA	0.878	**0.863**	**0.87**	0.913	**0.932**	**0.922**
	HYENA + meta-classifier	**0.89**	0.837	0.862	**0.916**	0.914	0.915

Table 3.8: Performance gain in precision by meta-classification

	Macro			Micro		
	Prec.	Rec.	F1	Prec.	Rec.	F1
HYENA	0.673	**0.638**	**0.644**	0.659	**0.681**	**0.67**
HYENA + meta-classifier	**0.693**	0.619	0.638	**0.674**	0.66	0.667

Table 3.9: Meta-classifier impact on the 5% worst-performing classes

3.3.4 HYENA Feature Analysis

In addition to a comprehensive feature set, HYENA exploits a large amount of training data and the gazetteer features derived from YAGO. To assess the impact of each asset, we varied the number of training instances and en-/disabled gazetteer features (cf. Table 3.10). Precision and recall improve from a larger training corpus, particularly for sparsely populated types at deeper levels of the type hierarchy . When gazetteer features are disabled, performance drops significantly.

3.3.5 Extrinsic Study on Named Entity Disambiguation

We present an extrinsic study on harnessing HYENA for named entity disambiguation (NED). Specifically, we consider a state-of-the-art NED tool, AIDA, provided by the authors of [HYB*11]. This NED method uses a combination of contextual similarity and entity-entity coherence for joint inference on how to map a set of entity mentions in an input text onto canonical entities registered in a knowledge base. It uses advanced graph algorithms which are computationally expensive. Alternative methods with similarly strong results would be based on machine learning with probabilistic factor graph which is equally if not more expensive. Therefore, it is desirable to prune the search space of potentially relevant candidate entities as much as possible and as early as possible. Hence, we use the type predictions by HYENA in order to identify candidate entities that are unlikely to be among the true entities for the given mentions (e.g. for the sentence "He was born in Victoria" and the mention "Victoria", the entities of type `Person`, `River` and `Lake` should be dropped).

To this end, we use the confidence scores of HYENA to remove entities of types with type scores below some threshold θ. Our technique proceeds in three steps:

(a) Invoke HYENA on the mention to obtain the predicted types for this mention as well

Size of training set (# of articles)	5 Top-level Types			All 505 Types		
	Prec.	Rec.	F1	Prec.	Rec.	F1
50,000	0.949	0.936	0.942	0.913	0.932	0.922
20,000	0.937	0.924	0.93	0.893	0.917	0.905
5,000	0.92	0.903	0.912	0.869	0.89	0.879
50,000 (without gazetteers)	0.915	0.825	0.868	0.82	0.718	0.766

Table 3.10: Micro-average impact of varying the number of Wikipedia articles used for training

Threshold	% dropped Entities	% unsolvable Mentions	avg. Document Prec.	avg. Mention Prec.
0.0	49.2	16.1	0.659	0.639
−0.5	45.7	12.3	0.738	0.713
−1.5	28.8	4.7	0.791	0.779
−2.5	17.7	2.2	0.802	0.798
AIDA	0	0	0.82	0.823

Table 3.11: Impact of Varying Type Prediction Confidence Threshold on NED Results

as their and confidence scores.

(b) Generate entity candidates using AIDA and its underlying name-entity dictionary.

(c) For each candidate, if there is no overlap between the entity types and the predicted mention types with confidence greater than or equal to θ, drop the candidate.

(d) Run AIDA on the reduced candidate space.

When dropping the correct entity, a mention becomes *unsolvable*. We vary the relaxation parameter θ to investigate search space rdecution versus mentions that are rendered *unsolvable*. We varied θ from -2.5 up to 0 with step size 0.5, and also compared to the variant without any pruning ($\theta = -\infty$). We performed our experiment on the extended CoNLL 2003 NER dataset with manual entity annotations from [HYB*11]. With a pruning threshold of $\theta = -1$, we can prune almost 40% of all entities while rendering less than 8% of the mentions unsolvable (cf. Table 3.11). The search space reduction of 40% actually results in a much larger saving in run-time because the graph algorithm that AIDA uses for NED has super-linear complexity (NP-hard in the worst case, but typically $O(n \log n)$ or $O(n^2)$ with appropriate approximation algorithms).

3.3.6 Related Work

There is little prior work on the task of classifying named entities, given in the form of (still ambiguous) noun phrases, onto fine-grained lexical types. The following methods are also considered in our experiments as state-of-the-art baselines.

[FlHo02] has been the first work to address type granularities that are finer than the handful of tags used in classical NER work (person, organization, location, date, money, other – see, e.g., [WRCh97, AlMa02, Cunn02, FGMa05]). It considered 8 sub-classes of the `Person` class, and developed a decision-tree classifier. [ESFP10] developed a maximum entropy classifier using word-level features from the mention contexts, but experimental results are flagged as non-reproducible in the ACL Anthology. [RaNg10] considered a two-level type hierarchy consisting of 29 top-level classes and a total of 92 sub-classes. These include many non-entity types such as date, time, percent, money, quantity, ordinal, cardinal, etc. The method uses a rich set of features, including WordNet senses of noun-phrase head words in mention contexts. [Giul09] proposed an SVD-based latent topic model with a semantic kernel that captures word proximities. The method was applied to a set of 21 different types; each mention is assigned to exactly one type. The work of [LiWe12] considered a two-level taxonomy with 112 tags taken from the Freebase knowledge base, forming a two-level hierarchy with top-level topics and 112 types (with entity instances). [LiWe12] trained a CRF for the joint task of recognizing entity mentions and inferring type tags. The feature set included the ones used in earlier work (see above) plus patterns from ReVerb [FSEt11].

3.4 Summary

Data cleaning and text mining, methods for entity resolution provide key assets for tracking named entities in the evolving Web, news, and social media. Based on the AIDA system we are able to identify text mentions of named entities in news articles, blog postings or contents from a discussion forum and map the mentions onto canonical entities [HYB*11, YHB*11]. AIDA does not annotate common words (like song, musician, idea, etc.). Also, AIDA does not identify mentions that have no entity in the repository. Once a name is in the dictionary containing all candidates for surface strings, AIDA maps it to the best possible candidate, even if the correct one is not in the entity repository.

In order to deal with less prominent (“out of knowledge base”) entities, we have developed the HYENA for fine-grained type classification of entity mentions [YBH*12, YBH*13]. In contrast to prior methods, we can deal with hundreds of types in a multi-level hierarchy, and consider that a mention can have many different types – a situation that does not (likely) occur in prior work with few types on merely two levels. We have shown that HYENA achieves high quality not only for the top-level types, but for all levels of our 9-level hierarchy. The proposed meta-classifier helps to further improve precision for difficult types. Finally, we have demonstrated the benefits of HYENA type predictions for reducing the search space and thus improving the efficiency of named entity disambiguation.

Chapter 4

Knowledge Evolution and Emerging Concepts

The constantly evolving Web reflects the evolution of society in the cyberspace. For instance, knowledge about entities (people, companies, political parties, etc.) evolves over time. New knowledge is added (e.g., awards) or changes (e.g., spouses, CEOs and similar positions). In addition, events related with these changes are reflected on the Web by postings in blogs, updates on corporate Websites, newspaper articles or even Wikipedia (depending on the importance of the respective entity).

Furthermore, abstracting from individual entities, we observe long-term changes in terminologies and topical taxonomies. The underlying research hypothesis is that events (such as affairs and awards on the temporal fact level, news coverage and changes in public opinion) "depend" on each other and show co-occurrence patterns in media. In order to understand these mutual dependencies between Web contents and the evolving societal knowledge it represents, we will now introduce systematic approaches toward comprehensively tracing knowledge evolution on the Web.

4.1 Granularity of Knowledge Evolution

There are at (least least) two levels of granularity when investigating knowledge evolution: fine-grained knowledge evolution and knowledge evolution at large. In the following, we will their characteristics and point out their conceptual differences.

4.1.1 Fine-grained Knowledge Extraction

In the field of *fine-grained knowledge extraction* we aim at gathering knowledge about entities (people, companies, political parties, etc.) that evolves over time. In order to achieve

this goal, we focus on a given input source (e.g., a particular Web page, news site, or discussion forum) and consider all conceivable relations at once. These data are harvested in our very large semantic knowledge base called YAGO [HSB*11]. It contains more than 2 million entities (like persons, organizations, cities, etc.) and 20 million facts about their relationships, which have been carefully harvested from Wikipedia and reconciled with the taxonomic class system of WordNet [Fell98].

As part of our research, we have extended YAGO toward time-awareness (Temporal YAGO or T-YAGO for short) [WZQ*10, WYZ*11], which allows us to link temporal facts to any kind of Web contents that may help us to trace the evolution of opinions such as ratings, reviews, comments, news, or discussion board entries. Next, we discuss the underlying information extraction techniques and our approach toward temporal information extraction.

4.1.2 Knowledge Evolution at Large

In the area of *knowledge evolution at large* we abstract from individual entities and rather looking into the long-term changes in terminologies and topical taxonomies. Collectively maintained Web catalogs organize links to interesting Web sites into topic hierarchies, based on community input and editorial decisions. These taxonomic systems reflect the interests and diversity of ongoing societal discourses. Catalogs evolve by adding new topics, splitting topics, or restructuring in order to capture newly emerging concepts of long-lasting interest.

To this end, we aim at the discovery of terminology shifts and their adaptation in widely used categorization schemes that reflect the collective memory/knowledge of society. Effects of this process in taxonomic category schemes are, for instance, the change of a term's meaning (such as Apple additionally becoming a computer category) or a newly appearing topic (such as SARS, iPhone, etc.). This entails that changes in categorization schemes are recognizable by a (prior) oscillation of word and phrase occurrences in documents, such as blogs, Websites, or news articles.

4.2 Knowledge Extraction about Entities

The world is highly dynamic and nothing lasts forever! Knowledge about entities evolves over time, and many facts are fairly ephemeral, e.g., winners of sports competitions, and occasionally even CEOs and spouses. In addition, many information needs by advanced users require *temporal knowledge* [StGe10, LiWe10, MaDa10]. For example, consider the following example question: "When did Dietmar Hopp found SAP and when did he leave the company?" Such a question is not being supported by existing knowledge bases. The problem we tackle is to automatically distill, from news articles and biography-style texts such as Wikipedia, *temporal facts* about entities for a given set of relations. By this we mean instances of the relations with additional time annotations that denote the validity

point or span of a relational fact. For example, for the *wasCreatedOnDate* relation between people and companies, we want to augment facts with the time points of the respective events; and for the *worksForCompany* relation between business people and companies, we would add the timespan during which the fact holds. This can be seen as a specific task of extracting ternary relations, which is much harder than the usual information extraction issues considered in prior work.

Targeted vs. Open Fact Extraction

Fact extraction can be pursued in an *output-oriented targeted* ("closed") manner or in an *input-oriented generic* ("open") manner. In the case of *output-oriented targeted fact extraction*, we are driven by a given set of relations for which we would like to gather instances. We are flexible in choosing our sources (e.g., can go only for easier or cleaner sources with high return) and we can exploit redundancy on the Web. Moreover, we have great flexibility regarding how deep natural-language text is analyzed in a demand-driven manner. A typical use case is to find the Alma Mater of as many scientists as possible, using a small set of seed facts for training. In the case of *input-oriented generic fact extraction*, we are focusing on a given input source (e.g., a particular Web page, news site, or discussion forum) and consider all conceivable relations at once. This approach inevitably requires deep analysis of natural-language text, and it can be successful only if sufficient training data is provided. Here training data typically has the form of fine-grained annotations for complete sentences or entire passages (e.g., the PropBank or FrameNet corpora). TextRunner and tools for semantic role labeling fall into this category of information extraction (IE) approaches. A typical use case is to automatically annotate news and extract as many relations as possible from each news item. Input-oriented generic IE is more ambitious than output-oriented targeted IE, but requires much more computational efforts for natural-language analysis and critically depends on the availability of sufficiently large and representative training data. Therefore, output-oriented targeted IE usually achieves significantly higher accuracy, and is generally more robust. To this end, we will pursue output-oriented targeted IE methods.

Temporality in Fact Extraction

Fact extraction is extended by *temporality* in identifying the time point or time interval for which a fact is valid. This is usually determined from the same sentence or passage from which the fact itself is extracted. But multiple extractions for the same fact (from different sources) can be combined to strengthen the temporal information. This can be in the form of refining the temporal resolution (e.g., the exact date rather than only year for the begin of some holding a political position), filling incomplete information (e.g., unknown end of the term for a political position), or invalidating false hypotheses by consistency checks (e.g., positions seemingly held after a person's death). Temporal expressions can be explicit like dates (e.g., July 15, 2009) or implicit like adverbial phrases (e.g., years later). The former

can be extracted by regular expression matching, the latter require deep natural-language analysis (e.g., dependency parsing). For both it is often necessary to a) validate that they actually refer to the considered fact (and not to another aspect of the same sentence) and b) determine the exact denotation that connects the fact and the temporal expression. For example, an expression may denote the begin of an interval during which the fact holds, its end, both, or a relative timepoint or interval (e.g., before this event, years later). These steps also need some form of natural-language analysis, ranging from part-of-speech tagging for very simple sentences to dependency parsing for complex sentences. In addition, a classifier may be called for to determine the nature and denotation of a temporal expression, based on the sentence's features (e.g., its dependency graph). To cope with the variety of temporal expressions, a unified representation is helpful. Explicit expressions should have (earliest, latest) bounds for timepoints and the begin and end of intervals. This is convenient for capturing different resolutions, and checking, cleaning, and refining temporal information. For implicit expressions, a suitable representation would still have to be developed.

4.2.1 Temporal Fact Harvesting from Web Contents

In order to support large-scale temporal fact harvesting from aggregated Web (archive) data, we have developed the PRAVDA (label Propagated fAct extraction on Very large DAta) system [WYQ*11, WDSW12, WDR*12, WDSW12]. Temporal facts distill the evolving knowledge over time, such as winners of sports competitions, or CEOs, spouses, etc.. This kind of temporal knowledge is an indispensable asset to support many information needs by advanced users and is used to populate the YAGO2 knowledge base. For example, consider the following example questions: "Who were the team mates of Diego Maradona during the 1990 FIFA World Cup?" "When did Madonna get married, when did she get divorced?" None of these questions are supported by existing knowledge bases.

The problem we address is to automatically distill from Web contents temporal facts for a given set of relations. By this we mean instances of the relations with additional time annotations that denote the validity point or span of a relational fact. For example, for the *winsAward* relation between people and awards, we want to augment facts with the time points of the respective events; and for the *worksForClub* relation between athletes and sports clubs, we would add the timespan during which the fact holds. This can be seen as a specific task of extracting ternary relations, which is much harder than the usual information extraction issues considered in prior work.

PRAVDA gathers fact candidates and distills facts with their temporal extent based on a new form of label propagation (LP). This is a family of graph-based semi-supervised learning methods, applied to (in our setting) a similarity graph of fact candidates and textual patterns. LP algorithms start with a small number of manually labeled seeds, correct facts in our case, and spread labels to neighbors based on a graph regularized objective function which we aim to minimize. We adopt the specific algorithm of [TaCr09], coined MAD (Modified Adsorption), with an objective function that combines the quadratic loss between initial

labels (from seeds) and estimated labels of vertices with a data-induced graph regularizer and an L2 regularizer. The graph regularizer is also known as the un-normalized graph Laplacian, which penalizes changes of labels between vertices that are close. We develop substantial extensions, and show how to judiciously construct a suitable graph structure and objective function. Notably, we consider inclusion constraints between different relation labels for the same node in the graph. For example, we may exploit that a relation like *joinsClub* (with time points) is a sub-relation of *worksForClub* (with time spans). The outcome is an assignment of labels to nodes which can be interpreted as a per-node probability distribution over labels. In our scenario, the labels denote relations to which the fact in a correspondingly labeled node belongs.

Facts and Observations

We aim to extract factual knowledge transient over time from free text. More specifically, we assume *time* $\mathcal{T} = [0, T_{max}]$ to be a finite sequence of time-points with yearly granularity. Furthermore, a *fact* consists of a relation with two typed arguments and a time-interval defining its validity. For instance, we write $worksForClub(Beckham, RMadrid)@[2003, 2008)$ to express that Beckham played for Real Madrid from 2003 to 2007. Since sentences containing a fact and its full time-interval are sparse, we consider three kinds of textual observations for each relation, namely *begin*, *during*, and *end*. "Beckham signed for Real Madrid from Manchester United in 2003." includes both the *begin* observation of Beckham being with Real Madrid as well as the *end* observation of working for Manchester. A *positive seed fact* is a valid fact of a relation, while a *negative seed fact* is incorrect (e.g., for relation $worksForClub$, a *positive seed fact* is $worksForClub(Beckham, RMadrid)$, while $worksForClub(Beckham, BMunich)$ is a *negative seed fact*).

Framework

As depicted in Figure 4.1, our framework is composed of four stages, where the first collects candidate sentences, the second mines patterns from the candidates sentences, the third extracts temporal facts from the sentences utilizing the patterns and the last removes noisy facts by enforcing constraints.

Preprocessing

We retrieve all sentences from the corpus comprising at least two entities and a temporal expression, where we use YAGO for entity recognition and disambiguation (cf. [HYB*11]).

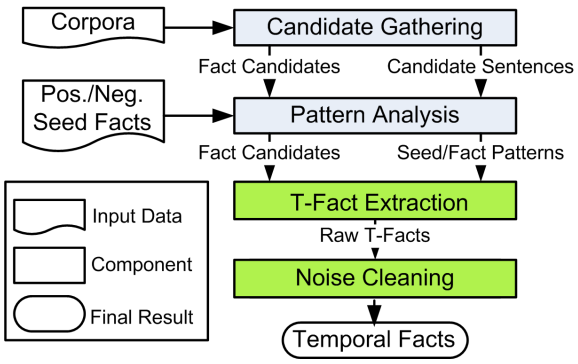

Figure 4.1: System Overview

Pattern Analysis

A *pattern* is a n-gram based feature vector. It is generated by replacing entities by their types, keeping only stemmed nouns, verbs converted to present tense and the last preposition. For example, considering "Beckham signed for Real Madrid from Manchester United in 2003." the corresponding pattern for the *end* occurrence is "sign for CLUB from". We quantify the *strength* of each pattern by investigating how frequent the pattern occurs with seed facts of a particular relation and how infrequent it appears with negative seed facts.

Fact Candidate Gathering

Entity pairs that co-occur with patterns whose strength is above a minimum threshold become fact candidates and are fed into the next stage of label propagation.

4.2.2 T-Fact Extraction

Building on [WYQ*11] we utilize Label Propagation [TaCr09] to determine the relation and observation type expressed by each pattern.

Graph

We create a graph $G = (\mathcal{V}_F \dot{\cup} \mathcal{V}_P, \mathcal{E})$ having one vertex $v \in \mathcal{V}_F$ for each fact candidate observed in the text and one vertex $v \in \mathcal{V}_P$ for each pattern. Edges between $\mathcal{V}_F$ and $\mathcal{V}_P$ are introduced whenever a fact candidate appeared with a pattern. Their weight is derived from the co-occurrence frequency. Edges among $\mathcal{V}_P$ nodes have weights derived from the n-gram overlap of the patterns.

Labels

Moreover, we use one label for each observation type (*begin*, *during*, and *end*) of each relation and a dummy label representing the unknown relation.

Objective Function

Let $\mathbf{Y} \in \mathbb{R}_{+}^{|\mathcal{V}|\times|Labels|}$ denote the graph's initial label assignment, and $\widehat{\mathbf{Y}} \in \mathbb{R}_{+}^{|\mathcal{V}|\times|Labels|}$ stand for the estimated labels of all vertices, $\mathbf{S}_l$ encode the seed's weights on its diagonal, and $\mathbf{R}_{*l}$ contain zeroes except for the dummy label's column. Then, the objective function is:

$$\mathcal{L}(\widehat{\mathbf{Y}}) = \sum_{\ell} \begin{bmatrix} (\mathbf{Y}_{*\ell} - \widehat{\mathbf{Y}}_{*\ell})^T \mathbf{S}_\ell (\mathbf{Y}_{*\ell} - \widehat{\mathbf{Y}}_{*\ell}) \\ +\mu_1 \widehat{\mathbf{Y}}_{*\ell}^T \mathbf{L} \widehat{\mathbf{Y}}_{*\ell} + \mu_2 \|\widehat{\mathbf{Y}}_{*\ell} - \mathbf{R}_{*\ell}\|^2 \end{bmatrix} \tag{4.1}$$

Here, the first term $(\mathbf{Y}_{*\ell} - \widehat{\mathbf{Y}}_{*\ell})^T \mathbf{S}_\ell (\mathbf{Y}_{*\ell} - \widehat{\mathbf{Y}}_{*\ell})$ ensures that the estimated labels approximate the initial labels. The labeling of neighboring vertices is smoothed by $\mu_1 \widehat{\mathbf{Y}}_{*\ell}^T \mathbf{L} \widehat{\mathbf{Y}}_{*\ell}$, where $\mathbf{L}$ refers to the Laplacian matrix. The last term is a L2 regularizer.

4.2.3 Cleaning of Fact Candidates

To prune noisy t-facts, we compute a consistent subset of t-facts with respect to temporal constraints (e.g. joining a sports club takes place before leaving a sports club) by an Integer Linear Program (ILP).

Variables

We introduce a variable $x_r \in \{0, 1\}$ for each t-fact candidate $r \in \mathcal{R}$, where 1 means the candidate is valid. Two variables $x_{f,b}, x_{f,e} \in [0, T_{max}]$ denote begin (b) and end (e) of time-interval of a fact $f \in \mathcal{F}$. Note, that many t-fact candidates refer to the same fact f, since they share their entity pairs.

Objective Function

The objective function intends to maximize the number of valid raw t-facts, where w_r is a weight obtained from the previous stage:

$$\max \sum_{r \in \mathcal{R}} w_r \cdot x_r$$

Intra-Fact Constraints

$x_{f,b}$ and $x_{f,e}$ encode a proper time-interval by adding the constraint:

$$\forall f \in \mathcal{F} \quad x_{f,b} < x_{f,e}$$

Given a single relation, we assume the sets $\mathcal{R}_b$, $\mathcal{R}_d$, and $\mathcal{R}_e$ to comprise t-fact candidates with respect to the *begin*, *during*, and *end* observations. Then, we introduce the constraints:

$$\forall l \in \{b, e\}, r \in \mathcal{R}_l \qquad t_l \cdot x_r \leq x_{f,l} \tag{4.2}$$

$$\forall l \in \{b, e\}, r \in \mathcal{R}_l \quad x_{f,l} \leq t_l \cdot x_r + (1 - x_r) T_{max} \tag{4.3}$$

$$\forall r \in \mathcal{R}_d \quad x_{f,b} \leq t_b \cdot x_r + (1 - x_r) T_{max} \tag{4.4}$$

$$\forall r \in \mathcal{R}_d \qquad t_e \cdot x_r \leq x_{f,e} \tag{4.5}$$

where f has the same entity pair as r and t_b, t_e are begin and end of r's time-interval. Whenever x_r is set to 1 for *begin* or *end* t-fact candidates, Eq. (4.2) and Eq. (4.3) set the value of $x_{f,b}$ or $x_{f,e}$ to t_b or t_e, respectively. For each *during* t-fact candidate with $x_r = 1$, Eq. (4.4) and Eq. (4.5) enforce $x_{f,b} \leq t_b$ and $t_e \leq x_{f,e}$.

Inter-Fact Constraints

Since we can refer to a fact f's time interval by $x_{f,b}$ and $x_{f,e}$ and the connectives of Boolean Logic can be encoded in ILPs [Karp72], we can use all temporal constraints expressible by Allen's Interval Algebra [Alle83] to specify inter-fact constraints. For example, we leverage this by prohibiting marriages of a single person from overlapping in time.

Previous Work

In comparison to [TWMi12], our ILP encoding is time-scale invariant. That is, for the same data, if the granularity of $\mathcal{T}$ is changed from months to seconds, for example, the size of the ILP is not affected. Furthermore, because we allow all relations of Allen's Interval Algebra, we support a richer class of temporal constraints.

4.2.4 Experiments

In this section, we study the impact of constraints on temporal fact extraction.

Corpus
Experiments are conducted in the soccer and the celebrity domain by considering the *worksForClub* and *isMarriedTo* relation, respectively. For each person in the "FIFA 100 list" and "Forbes 100 list" we retrieve their Wikipedia article. In addition, we obtained about 80,000 documents for the soccer domain and 370,000 documents for the celebrity domain from BBC, The Telegraph, Times Online and ESPN by querying Google's News Archive Search[1] in the time window from 1990-2011. All hyperparameters are tuned on a separate data-set.

Seeds
For each relation we manually select the 10 positive and negative fact candidates with highest occurrence frequencies in the corpus as seeds. For instance, a *positive seed fact* for the $worksForClub$ would be $worksForClub(Beckham, RMadrid)$, while $worksForClub$ $(Beckham, BMunich)$ would be a *negative seed fact*.

Evaluation
We evaluate *precision* by randomly sampling 50 (*isMarriedTo*) and 100 (*worksForClub*) facts for each observation type and manually evaluating them against the text documents. All experimental data is available for download from our website[2].

Pipeline vs. Joint Model

Setting
In this experiment we compare the performance of the pipeline being stages 3 and 4 in Figure 4.1 and a joint model in form of an ILP solving the t-fact extraction and noise cleaning at the same time. Hence, the joint model resembles [RoYi04] extended by Section 4.2.3's temporal constraints.

Results
Table 4.1 shows the results on the pipeline model (lower-left), joint model (lower-right), label-propagation w/o noise cleaning (upper-left), and ILP for t-fact extraction w/o noise cleaning (upper-right).

Analysis
Regarding the upper part of Table 4.1 the pattern-based extraction works very well for *worksForClub*, however it fails on *isMarriedTo*. The reason is, that the types of *worksForClub* distinguish the patterns well from other relations. In contrast, *isMarriedTo*'s patterns interfere with the huge number of other person-person relations making constraints a decisive asset. When comparing the joint model and the pipeline model, the former sacrifices recall in order to keep up with the latter's precision level. That is because the joint model's ILP decides with binary variables on which patterns to accept. In contrast, label propagation addresses the inherent uncertainty by providing label assignments with confidence numbers.

[1] http://news.google.com/archivesearch
[2] http://www.mpi-inf.mpg.de/yago-naga/pravda

Relation	Observation	Label Propagation		ILP for T-Fact Extraction		
		Precision	# Obs.	Precision	# Obs.	
worksForClub	*begin*	80%	2537	81%	2426	*Without Noise Cleaning*
	during	78%	2826	86%	1153	
	end	65%	440	50%	550	
isMarriedTo	*begin*	52%	195	28%	232	
	during	76%	92	6%	466	
	end	62%	50	2%	551	
worksForClub	*begin*	85%	2469	87%	2076	*With Noise Cleaning*
	during	85%	2761	79%	1434	
	end	74%	403	72%	275	
isMarriedTo	*begin*	64%	177	74%	67	
	during	79%	89	88%	61	
	end	70%	47	71%	28	

Table 4.1: Pipeline vs. Joint Model

Increasing Recall

Setting

In a second experiment, we move the t-fact extraction stage away from high precision towards higher recall, where the successive noise cleaning stage attempts to restore the precision level.

Results

The columns of Table 4.2 show results for different values of μ_1 of Eq. (4.1). From left to right, we used $\mu_1 = e^{-1}, 0.6, 0.8$ for *worksForClub* and $\mu_1 = e^{-2}, e^{-1}, 0.6$ for *isMarriedTo*. The table's upper part reports on the output of stage 3, whereas the lower part covers the facts returned by noise cleaning.

Analysis

For the conservative setting label propagation produces high precision facts with only few inconsistencies, so the noise cleaning stage has no effect, i.e. no pruning takes place. This is the setting usual pattern-based approaches without cleaning stage are working in. In

contrast, for the standard setting (coinciding with Table 4.1's left column) stage 3 yields less precision, but higher recall. Since there are more inconsistencies in this setup, the noise cleaning stage accomplishes precision gains compensating for the losses in the previous stage. In the relaxed setting precision drops too low, so the noise cleaning stage is unable to figure out the truly correct facts. In general, the effects on *worksForClub* are weaker, since in this relation the constraints are less influential.

		Conservative		Standard		Relaxed		
		Prec.	**# Obs.**	**Prec.**	**# Obs.**	**Prec.**	**# Obs.**	
worksForClub	*begin*	83%	2443	80%	2537	80%	2608	*Without Noise Cleaning*
	during	81%	2523	78%	2826	76%	2928	
	end	77%	377	65%	440	62%	501	
isMarriedTo	*begin*	72%	112	52%	195	44%	269	
	during	90%	63	76%	92	52%	187	
	end	67%	37	62%	50	36%	116	
worksForClub	*begin*	83%	2389	85%	2469	84%	2536	*With Noise Cleaning*
	during	88%	2474	85%	2761	75%	2861	
	end	79%	349	72%	403	70%	463	
isMarriedTo	*begin*	72%	111	64%	177	46%	239	
	during	90%	62	79%	89	54%	177	
	end	69%	36	68%	47	38%	110	

Table 4.2: Increasing Recall

4.2.5 Related Work

Recently, there have been several approaches that aim at the extraction of temporal facts for the automated construction of large knowledge bases, but time-aware fact extraction is still in its infancy. An approach toward fact extraction based on coupled semi-supervised learning for information extraction (IE) is NELL [CBW*10]. However, it does neither incorporate constraints nor temporality. Temporal information extraction (TIE) [LiWe10]

binds time-points of events described in sentences, but does not disambiguate entities or combine observations to facts. Further, it uses training data with fine-grained annotations to learn an inference model based on Markov Logic. This involves using consistency constraints on the relative ordering of events. This machinery is computationally expensive and cannot easily be scaled up. [WZQ*10] focuses on extracting relevant timepoints and intervals from semistructured data in Wikipedia: dates in category names, lists, tables, infoboxes. However, there is no support for processing free text. A pattern-based approach for temporal fact extraction is PRAVDA [WYQ*11], which utilizes label propagation as a semi-supervised learning strategy, but does not incorporate constraints. Similarly, TOB is an approach of extracting temporal business-related facts from free text, which requires deep parsing and does not apply constraints as well [ZSWe08]. It workes reasonably well, but is computationally expensive, requires extensive training, and cannot be easily generalized to other relations. In contrast, CoTS [TWMi12] introduces a constraint-based approach of coupled semi-supervised learning for IE, however not focusing on the extraction part. Building on TimeML [PCI*03] several works [VMS*05, MVW*06, ChJu08, VGS*09, YRAM09] identify temporal relationships in free text, but don't focus on fact extraction.

4.3 Evolution of Collective Web Catalogs

The constantly evolving Web reflects the evolution of society in the cyberspace. Projects like the Open Directory Project (http://www.dmoz.org) or Yahoo Directory (http://dir.yahoo.com) can be understood as a collective memory of society on the Web. It represents the topical knowledge in a structured way, in form of taxonomy. This taxonomy consists of topics, which are connected by "parent-child" relations. A parent topic is broader, whereas all its children deal with more narrow topics. Each topic in the taxonomy holds a set of links to related resources in the Internet. The taxonomy, once constructed, is constantly evolving, reflecting the human cognition of societal trends. There are several types of changes which can be performed in the taxonomy: topic merging or splitting, renaming, removal, and addition. In this paper we focus on the formation of new concepts that lead to adding a topic to the taxonomy.

New concepts first appear, in latent form, in sources external to the Web catalogues: in news, blogs, and other social media. Figure 4.3 shows the presence of the words "Japan nuclear plant tsunami" in news articles as a function of time. There is a considerable and sudden increase of interest in these terms around March 2011. The intensity remains at a high-level for several months. The bottom part of Figure 4.3 shows two consecutive versions of the DMOZ topic "Safety and Accidents". The snapshots of this topic from late 2010 and mid 2011 differ in the subtopics they contain. There is a clear correlation between the massive emergence of news dedicated to the news coverage about "Japan nuclear plant tsunami" and the extension of the DMOZ taxonomy with the topic "Fukushima 2011". We hypothesize, that any concept emerging in news has first to reach a certain cognition level to become a part of taxonomy.

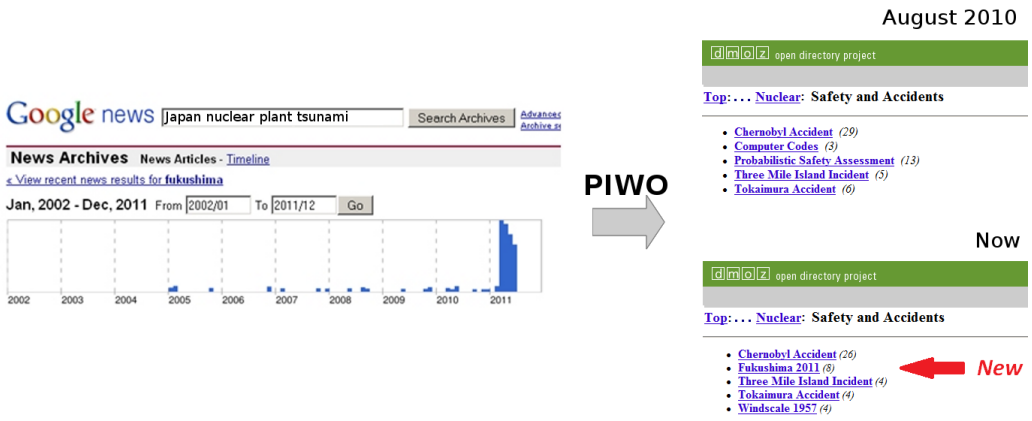

Figure 4.2: "Japan nuclear plant tsunami" in the news and the emergence of the new DMOZ topic "Fukushima 2011" in "Safety and Accidents"

Temporal analysis of news archives reveals that concepts discussed in news media behave differently over time. Some concepts appear occasional while others are persistent. Occasional concepts attract the attention of news media for very short periods of time and fade out afterwards. An example is a meeting of Germany's chancellor with the American president. Concepts of this kind do not lead to new topics in the Web catalog. Persistent concepts, on the other hand, receive attention almost every day over long time periods; an example is global warming.

To this end, we have developed a data-analysis and prediction system called PIWO (Predicting evolution In Web catalOgues) [PSWe12]. PIWO has the following salient properties:

(a) models for extracting emerging latent concepts and predicting novel topics in a taxonomy, based on a temporal term relatedness graph built from news articles;

(b) a judiciously designed clustering algorithm, based on maximal cliques in the temporal term relatedness graph, to identify latent concepts in a scalable manner using Map-Reduce computing;

(c) an algorithm for predicting structural changes in Web catalogues, based on statistical measures of latent concepts;

(d) experiments with the New York Times (NYT) archive for concept extraction and predicting new sub-topics in the "Health", "Business" and "Science/Technology" topics of the Open Directory Project (http://www.dmoz.org).

As such, PIWO is a valuable tool for discovering emerging concepts, by easing the task of editors and strengthening the quality of crowdsourcing-based methods.

4.3.1 Concept Mining

The input to the PIWO system is a time-ordered sequence of *snapshots* of news and taxonomy states. Each news snapshot is either a batch of newspaper articles from an archive (e.g., on a monthly or weekly basis) or a set of incoming articles from online news and social media (e.g., on a daily basis). Each taxonomy state is a tree or directed acyclic graph (DAG) of explicitly named topics. Typically, the taxonomy changes on a monthly basis. Figure 4.3 shows the architecture of PIWO. It consists of two main components: the *concept miner* and the *taxonomy change predictor*. The concept miner discovers latent concepts in the news snapshots, which can enrich the next state of the taxonomy. In addition, it filters out unimportant short-living concepts with low cognition level and long-living concepts without remarkable burst periods, which we treat as background noise. The taxonomy change predictor takes as input a set of emerging concepts, compares them to the current state of the taxonomy, and identifies topics expected to be restructured by adding a new sub-topic.

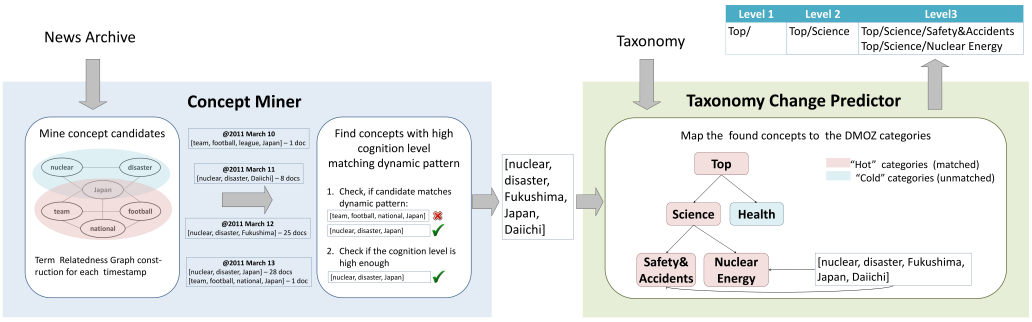

Figure 4.3: Overview of the PIWO system

Concept Candidate Gathering

To gather latent concepts we employ clustering on terms (words or phrases), using co-occurrence-based relatedness measures as similarity. This should allow term polysemy, terms not belonging to any concept (noise), and a variable number of latent concepts. These goals determine the class of appropriate clustering methods: hierarchical agglomerative clustering (HAC) allowing overlaps between clusters as well as left-out data points (not assigned to any cluster) [ScPe95]. The relatedness between two terms can be computed by the cosine similarity of the corresponding document vectors. The relatedness of terms has a temporal nature and is defined with respect to a timepoint. For instance, shortly after a presidential election, the terms *president*, *elections*, *results* may form a concept and be closely related to each other, while a year after the election they may not co-occur anymore.

Based on the pairwise term relatedness, the sets of mutually related terms at time t can be extracted. One way of modelling the inter-related term sets is to represent them in form of a

graph. Nodes of the graph correspond to the terms from the lexicon W. If the relatedness of two terms is above a predefined threshold, the corresponding nodes are connected with an edge.

Definition An undirected graph $TRG^t = (V, E)$ is a *term-relatedness graph* at time t, where V is the set of terms in the lexicon W at time t and an edge $e_{i,j} \in E$ exists if $\forall i, j \in V, i \neq j : rel(w_i, w_j, t) > \tau_{rel}$. τ_{rel} is a specified threshold value for term relatedness. $rel(w_i, w_j, t)$ is the cosine similarity between document vectors corresponding to terms w_i, w_j at time t.

The term sets identified this way are referred to as call *concept candidates*. The set of candidates is constructed for each snapshot of the underlying news data. The candidates can then be temporally ordered by their first appearance.

Definition Let W be the lexicon of the snapshot at time t. A set of terms $c^t = \{w_1, w_2 \dots w_n\} \subseteq W$ is a *concept candidate* at time t, if $\forall\ i, j \in [1..n]\ i \neq j : rel(w_i, w_j, t) > \tau_{rel} \geq 0$. The set of candidates for time t is denoted as C^t.

Our clustering method on the term-relatedness graph can be seen as a horizontal cut in a HAC dendrogram. Given the threshold τ_{rel}, the clustering is then equivalent to finding all maximal cliques in the graph. As this is an expensive computation, we devised a distributed Map-Reduce algorithm to parallelize and speed up the clustering (inspired by Wu et al. [WYZW09]). To this end, the graph is partitioned with replication. We hash-partition the graph nodes and store all first- and second-order neighbors of the respective nodes in the same partition. Then each partition computes maximal cliques in parallel. By the definition of a maximal clique, no clique can be missed this way nor will there be any false positives. Of course, this parallelization does not escape the NP-hardness of the problem: a huge clique will slow down one of the partitions and will be the bottleneck in the Map-Reduce computation. Empirically, however, we found that most cliques are not that big, so that we achieved good scale-up performance. This clustering is carried out once for each snapshot; so our scalable method is crucial for practical viability.

Concept Candidate Dynamics

Not all term sets mined from the archive are equally appropriate. Some term sets appear only once during the whole period of observation. They might be noise or represent an unimportant entity or event and therefore are not worth to be added to taxonomy. Unlike them, there are mature candidates which, having emerged once, attract attention for longer periods of time, such that their cognition level grows.

For terms sets to be truly relevant concepts they have to present for an extended time period, spanning multiple snapshots. However, such a set of co-occurring terms may vary over

time, dropping and adding terms across snapshots. If this variation is low, we refer to the latent concept candidate as a *stable term set*. The deviation of two candidates $dev(c^t, c^{t'})$ is measured by the Jaccard similarity of the corresponding term sets.

Definition Let $\tilde{c} = \{c^{t_j}, \ldots c^{t_{j+s}}\}$ be a sequence of concept candidates for the time window from t_j to t_{j+s}. We call $\tilde{c}$ a *stable term set* for time point t_{j+s} if$\forall\ i \in [j \ldots j+s-1]:\ dev(c^{t_{j+s}}, c^{t_i}) \leq \tau_{dev}$, where τ_{dev} is a specified threshold for the deviation.

For computing stable term sets, we scan the concept candidates in time order using a window of size s snapshots, and compare term sets between successive snapshots. For the qualifying candidates, we can then express the *concept dynamics* as a sequence of occurrence counters.

Definition The *dynamics of concept* $\tilde{c}$ is defined as $dyn(\tilde{c}) = \{occ(c^{t_i}), \ldots occ(c^{t_{i+s}})\}$ where $occ(c^t)$ is the number of articles containing candidate c^t.

We are now ready to define the appearance of a new concept, emerging in a time window but not present at the begin of the window.

Definition The concept dynamics $dyn(\tilde{c})$ forms an *emerging pattern* if there is a time point $t_{i+k} \in [t_i \ldots t_{i+s}]$ such that:

$$\forall t' < t_{i+k}\ occ(c^{t'}) = 0 \wedge\ \forall t' \geq t_{i+k}\ occ(c^{t'}) > 0$$

Concept Candidate Cognition Level

Among all candidates with stable term sets and emerging patterns, we select only those with high *cognition level*, as measured by the frequency and prominence of candidate occurrences. This measure considers the number of documents containing the term set, the lifespan of the concept, and the quality of the sources where we observe the concept within a time window. The source quality is relevant if our news snapshots comprise different collections, for instance, newspapers, blogs, and social-media postings.

Definition Let R be the set of news collections. The *cognition level* of concept c reached within the time window $[t_i \ldots t_{i+s}]$ is the aggregation of cognition levels reached in each collection within this sliding window:

$$cogn_{t_{i+s}}(dyn_R(c)) = \sum_{r \in R} quality(r) \cdot \frac{docs_r(c)}{lifespan_r(c)}$$

where $quality(r)$ is a measure for collection quality, $lifespan_r(c)$ denotes the number of snapshots in $[t_i \ldots t_{i+s}]$ in which c exists in collection r, and $docs_r(c)$ is the total number of documents in r containing c within $[t_i \ldots t_{i+s}]$.

This definition gives flexibility in treating information from different sources. In our experiments, the collection solely consists of the NYT news archive; we will set $quality(r) = 1$ in this special case.

4.3.2 Interesting Concepts

We now define a notion of *interesting concept* (inspired by [ScSp06]), by combining the following three requirements:

(a) a concept must be a stable term set for some time window in the news history, as opposed to a sporadic, very short-lived concept;

(b) it must be an emerging pattern at some time point (not necessarily in the window of term-set stability), as opposed to a persistent concept that exists independent of time;

(c) it must reach sufficient cognition level across all snapshots, as opposed to a highly special, low-caliber concept.

Definition Let $t_1, t_2 \ldots t_m$ be m consecutive snapshots of the news archive on a given granularity level. A candidate $\tilde{c} = c^{t_m}$ is a *concept*, if:

(a) there are $k \leq m$ consecutive snapshots of the news archive where a candidate $c^{t_i} \in C^{t_i}$ $i \in [m-k..m-1]$ can be found, such that $dev(\tilde{c}, c^{t_i}) \leq \tau_{dev}$

(b) the dynamics of the concept matches the occurrence-based dynamic pattern

(c) the accumulated cognition level is $cogn(\tilde{c}) \geq \tau_{cogn}$

where τ_{dev} and τ_{cogn} are predefined thresholds.

The first property guarantees a considerable life span of a concept. We allow the candidates representing the same concept to differ over time by at most τ_{dev}. Matching the occurrence-based dynamic pattern delivers the set of new concepts, that is unseen until some point within the sliding window. The third property ensures a necessary concept cognition level, accumulated over the observation period.

4.3.3 Taxonomy Change Prediction

The Open Directory Project (or DMOZ) is a human-edited Web directory, which aims at organizing Web resources. It is a classification lightweight ontology ([LiOz09]), which is intended for classifying, describing and accessing a large collection of resources on the Web.

Given the formation of new latent concepts, the task now is to predict a new sub-topic that will be added to the taxonomy of a Web catalogue.

The *topics* in a taxonomy are hierarchically organized by means of several relations, most notably, a specialization/generalization tree or DAG. Another relation connects related topics, orthogonally to the main hierarchy. In the following we focus on this main hierarchy, assuming that it is a DAG. Each topic node is associated with a set of *Web links* and short text snippets describing the corresponding Web sites that have been assigned to the topic.

Definition A *taxonomy* at time point t is a directed acyclic graph $T^t = \langle N, E \rangle$, where the node set N corresponds to a set K^t of topics and the edge set E corresponds to topic pairs $k_1, k_2 \in K^T$ such that k_2 is thematically subsumed by (i.e., is more narrow than) k_1. We assume the graph has a single root $Top \in N$ that comprises all topics in the taxonomy. Each node k is associated with a set of terms $Ext(k)$ consisting of the terms in the topic description and the descriptions and URLs of the topic's Web links. $W_T^t = \cup_k Ext(k)$ is the lexicon of T^t.

When new substantial concepts appear in the news or in the blogosphere, a taxonomy tends to capture these changes, by adding a new sub-topic under an existing parent topic.

We hypothesize that all interesting concepts, as identified by the methods in Section 4.3.1, should lead to new topics in the next snapshot of the taxonomy. As observed in [BGMl08], sometimes not a single topic, but an entire subtree is inserted. This happens when a new topic contains a set of auxiliary topics. For instance, the topic *'Authors'* is likely to be added along with an alphabetical list to search an author by last name. We do not aim at predicting auxiliary changes of that type.

Definition Let T^t and T^{t+1} be consecutive snapshots of a taxonomy at time point t and $t+1$ respectively. The *creation of a new concept* is reported, iff there is a topic k' present in T^{t+1} but not in T^t and a topic k present in both T^{t+1} and T^t such that (k, k') is an edge in T^{t+1} and the extension of k does not overlap by more than θ with any topic in T^t, where θ is a specified threshold.

The property that a new topic k' does not inherit a large portion of Web links from any pre-existing topic reflects the freshness of k'. Otherwise, k' could have been derived from an old topic by renaming or merging/splitting categories. Such purely structural changes are not considered here. In Figure 4.4, the topic k' is a new concept. It has associated terms, which are not already contained in previously existing topics.

In order to predict taxonomy changes, we devise two strategies: concept-based and topic-based prediction.

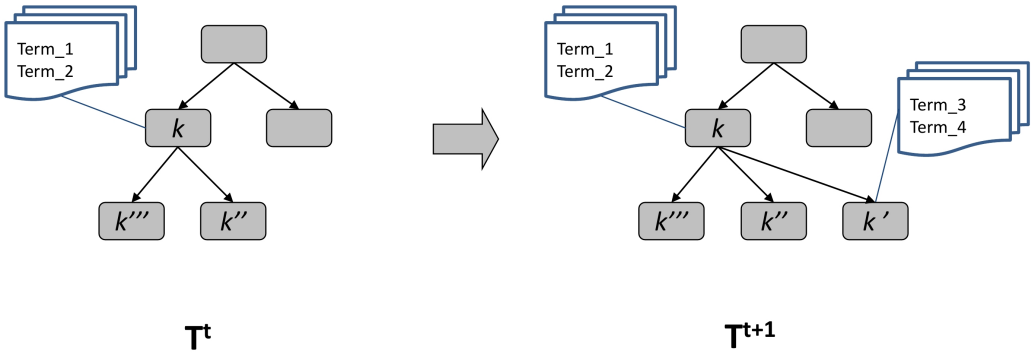

Figure 4.4: Creation of a new concept

Concept-based Prediction

This type of prediction is based on the concept dynamics and the concept cognition level discussed in the previous section. We hypothesize that all interesting concepts are individual candidates for future subtopics in the taxonomy.

Definition Let C^t be a set of interesting concepts obtained at time point t. Then, for all $C_i^t \in C^t$, C_i^t is reported to be a new topic in $T^{t'}$ where $t < t'$.

In Figure 4.3, the concept *C = {nuclear, disaster, Fukushima, Japan, Daiichi}* is identified as a new topic.

Topic-based Prediction

This type of prediction is based on the hypothesis that interesting concepts do not necessarily represent new topics by themselves, but rather identify the taxonomic contexts where new topics should appear. To this end, we map interesting concepts to their best matching topics in the current taxonomy version and predict that these topics will exhibit sub-topics in future taxonomy versions.

Definition Let C^t be a set of interesting concepts at time point t, and let T^t be the current taxonomy version. A topic $k_j \in N(T^t)$ is *hot*,if there is $c^t \in C^t$ such that more than a fraction of τ_{match} of the c^t terms is contained in $Ext(k_j)$.

Definition Let K be a set of hot topics in T^t. Then, for all $k_i \in K$, k_i is reported as a topic which will undergo the creation of a new concept change in $T^{t'}$, where $t < t'$.

In the example shown on Figure 4.3, the topic-based predictor determines that the concept *"nuclear, disaster, Fukushima, Japan, Daiichi"* matches two DMOZ topics, *"Top/Science/Safety*

and Accidents" and *"Top/Science/Nuclear Energy"*. These are the currently "hot" topics, which are likely to obtain a new child topic in future versions of DMOZ. However, the actual change may as well be more coarsely located in *"Top/Science/"*.

4.3.4 Experimental Evaluation

Datasets and System Configuration

Experiments were conducted on the New York Times (NYT) archive for concept finding. This archive contains about 1.5 million daily articles, spanning 20 years from $1^{st} January$ 1987 to 9^{th} $June$ 2007. The total size of the vocabulary is ca. 1 million terms. For the prediction task we used the snapshots of the Open Directory Project http://www.dmoz.org, containing about 80 DMOZ snapshots between $February$ 2003 and $August$ 2009. For a meaningful prediction, we focused on the time overlap between the NYT archive and the DMOZ snapshots on a weekly basis, covering 232 weeks from January 2003 till June 2007.

Experimental Setup

We ran the pairwise term-relatedness computation algorithm. All term pairs with cosine similarity higher than τ_{rel} = 0.9 were considered as related. The properties of constructed term-relatedness graph are summarized in Table 4.3.

avg $\|V(G)\|$	17965
avg $\|E(G)\|$	322777
avg density	0.002
avg $\|\Gamma(k) \cup \Gamma(\Gamma(k))\|$	963
avg number of cliques	58438

Table 4.3: Properties of term-relatedness graph G

We traced the mined candidates over a sliding window of six weeks. The deviation threshold value τ_{dev} was set to 0.4. We considered concepts existing for at least two weeks (τ_{cogn} = 2). So we accepted concepts only if they appeared at least twice a week on average.

We considered all DMOZ subtopics related to health ("Health" branch), business ("Business" branch), and technology ("Science/Technology" branch). To define the ground-truth for the new-concept prediction, we identified the set of topics which did not exist in the previous snapshot of DMOZ. We performed some data cleaning by excluding all empty topics and auxiliary topics (not populated with links). In addition, we paid attention to the fact that a large portion of newly added topics is merely caused by renaming, moving, merging, or splitting operations. Such topics inherit the links from the previous version. To exclude these types of changes, we selected only new topics which contained new Web links not present at all in the previous snapshot of DMOZ.

[MoMo65] showed that $\mathcal{O}(3^{n/3})$ is the upper bound on maximal cliques in a graph with n nodes. In our computations, the number of available reducers r lessens this complexity by factor r. The overhead caused during the MapReduce job is $\mathcal{O}(f(n^3))$ for lexicon size n and Hadoop-dependent run time cost function f, taking into account sorting and splitting the input data.

4.3.5 Results

We experimented with two prediction strategies: concept-based and topic-based. We considered the use case of aiding Web catalog editors. Therefore, we aimed at achieving high precision; recall was not a priority. NYT is mainly dedicated to US news with emphasis on the financial and business domains. Therefore, high coverage of all fine-grained topics (e.g., under health or technology) is infeasible.

Top/Health			**Top/Business**			**Top/Science/Technology**		
Depth	Precision	Recall	Depth	Precision	Recall	Depth	Precision	Recall
3	0.857	0.214	3	0.669	0.249	3	0.976	0.488
4	0.625	0.048	4	0.438	0.0975	4	0.500	0.138
5	0.333	0.009	5	0.266	0.040	5	0.229	0.087

Table 4.4: Concept-based prediction results

Top/Health			**Top/Business**			**Top/Science/Technology**		
Depth	Precision	Recall	Depth	Precision	Recall	Depth	Precision	Recall
3	0.847	0.324	3	0.730	0.239	3	0.952	0.476
4	0.695	0.136	4	0.462	0.0953	4	0.502	0.126
5	0.353	0.028	5	0.218	0.033	5	0.181	0.073

Table 4.5: Topic-based prediction results

Concept-based prediction considers all mined concepts as individual topics to be added to DMOZ. Table 4.4 shows the precision and recall of such concepts being really added at different levels of the taxonomy. We achieved pretty good precision between 67% and 98%. For example, PIWO correctly predicted a new concept *a-h1n1* in spring 2003. This is semantically very close to *SARS*, which was indeed added to DMOZ around this time.

Topic-based prediction aims to predict the existing topic where a new concept will be added. Table 4.5 shows the results. At high levels in the taxonomy, PIWO achieves high precision between 73% and 95%. For example (cf. Table 4.6 and Table 4.7), we correctly predicted the new sub-topic on *a-h1n1* under */Top/Health/ Conditions and Diseases/*. Another example is the emerging concept {*launching, space, international*} that caused PIWO to predict changes in the *Top/Science/Technology/Space/* topic. Indeed, new sub-topics *CALIPSO* and *CloudSAT* were added two months later, corresponding to launched satellites CloudSAT and international CALIPSO.

Space/Missions/Unmanned/Earth Observing/Aqua
Space/Missions/Unmanned/Earth Observing/CALIPSO
Space/Missions/Unmanned/Earth Observing/CloudSAT
Space/Missions/Unmanned/Earth Observing/Glory
Structural Engineering/Bridge/Failures/Minneapolis

Table 4.6: Examples of sub-categories added in the *Top/Science/Technology/* branch

Space
Space/Spacecraft and Satellite Design

Table 4.7: Example of topic-based predictor output for the *Top/Science/Technology/* branch

4.3.6 Related Work

Topic Modeling

Latent topic models [Blei12], like LSI [DDF*90, LMDK07] or LDA [BNJo03], capture the joint distribution of terms (or other observable features) and documents such as Web pages. Clustering methods and matrix-factorization techniques also fall into this wider class of models. They reduce the dimensionality of the underlying co-occurrence data and thus bring out the most important concepts in unlabeled form. However, there is no topical hierarchy and there is no consideration of the data's dynamics. The ThemeMonitor of [ScSp06] explores topic evolution in document collections. A set of disjoint clusters is constructed for each collection snapshot and traced over time. Long-living clusters are interpreted as topics.

All of the above models require predefining the number of latent concepts (clusters, topics) – quite a limitation when dealing with real-life data at large scale. Our model does not make any assumptions of this kind.

Dynamics Modeling

There are numerous time-series-based models of data dynamics. In particular, there is ample work on detecting general or topic-specific bursts in social media. [Klei03] models the temporal behavior of a topic using an infinite-state automaton. Each state corresponds to the particular intensity of topic mentions. Then a burst is a sequence of high-intensity states. However, finding an optimal state sequence is NP-hard problem. Based on this seminal work, [YCHZ12] discovers bursts of specific tag co-occurrences in collaborative tagging systems. Such bursty events are mapped to a taxonomy based on a tag hierarchy. [RHC*12] computes correlations between micro-blogging activities and stock market events. [MaKo10, AMRW11] develop methods for detecting emergent topics in blogs and tweets. Their notions of popular or emergent tag sets are related to our approach for mining

interesting concepts. However, the focus of this work is on the real-time efficiency at the expense of using relatively simple models for emerging topics. Our model is more sophisticated by considering variable-cardinality and overlapping term sets, with additional considerations on temporal stability and saliency. [NBGr11] study features of Twitter messages associated with a trend. We analyze the temporal behavior of concepts, but not the features of documents in the collection.

None of the above work considers pre-specified taxonomies like those of Web catalogs.

Prediction of Structural Changes in Taxonomies

Predicting topic additions to DMOZ has been considered in [BGMl08]. The approach is based on the hypothesis that a new subtopic is created when its parent topic contains multiple groups of tightly related Web links, leading to a topic split. The prediction is solely based on the state of the taxonomy itself. In contrast, we consider information that is external to the taxonomy, such as news, and predict totally new topics. [KiLe04] develops methods for automatically organizing a stream of incoming documents into a taxonomy, and interprets new documents that do not fit any category as a trigger for creating a new topic.Our approach treats the DMOZ structure as a gold standard, and proposes new topics only if there is substantial evidence that the current categories are insufficient. [WLWZ11] automatically builds a probabilistic taxonomy from Web contents. This is carried out as a single batch computation. There is no consideration to the dynamics of the taxonomy. [CYCH10] builds an evolving taxonomy from social tags of Web pages, using techniques from association rule mining. [DKM*07] builds timelines of tag clouds from a stream of tagged pages, but does not impose any taxonomic structure on them.

4.4 Summary

This section presented approaches toward an automatic mining of societal events in Web contents. On the fine-grained level, this has been pursued by the underlying approach of output-oriented targeted IE. Here, we have been gathering facts for a given set of relation of interest. Therefore, we have developed a unified framework for harvesting base facts and temporal facts from textual Web sources. From the experimental results we have demonstrated the viability and high accuracy of our approach. Our extended LP method is nicely geared for scale-out on distributed platforms.

On the coarse-grained level, the evolution of Web catalogs – representing the collective memories of society on the Web – can be predicted by detecting emerging latent concepts in the news. The PIWO system automatically discovers such concepts predicts new topics to be added to the Web catalog's taxonomy. Our experiments showed that PIWO achieves high precision when predicting changes for the third and fourth taxonomy levels.

Chapter 5

Temporal Web Analytics in Action

Web collections consist of both large crawls that harvest data from the Web by navigating through discovered URLs, social media such as Twitter API, Wikipedia structured temporal dumps, research collections and alike. These collections consist of a massive amount of widely heterogeneous resources [WHN*12, WNS*11]. Heterogeneity pertains to several dimensions: format (HTML documents, but also CSS, PDF, images, Office documents), content, language, temporal and spatial information.

By means of the work introduced in the previous sections, we enable temporal Web-scale analysis of data. In order to "glue" the previously presented components together, a unified framework is required. To this end, we present a "Virtual Web Observatory" that supports controlling, management and semantic enrichment of Web archive data. As show-case, selected demonstrators (either browser plug-ins and/or analytics interfaces) are presented. To this end, results presented here comprise project results (e.g. the overall framework, cf. section 5.1) and own contributions (the demonstrator section 5.2 on entity exploration and knowledge linking in section 5.3).

5.1 Virtual Web Observatory

The main objective of the Virtual Web Observatory (VWO) is to support Web content analytics for application stakeholders [SBVW12, BRSW13]. The VWO is also intended to support experimental research by academic users, on large-scale Internet data. Typical use-cases for realistic applications could be in opinion mining. For example, a business analyst may want to track the market share mobile operating systems like iOS and Android, and compare opinions on corresponding mobile phones in review forums and social media.

In the following, we introduce the key components of the VWO that has been realized has a demonstration showcase of the LAWA project [SpWe12]. The VWO integrates all analytics applications into a unified view and can be used as a blueprint for adding further Web analytics tools [BRSW13, BRSp13].

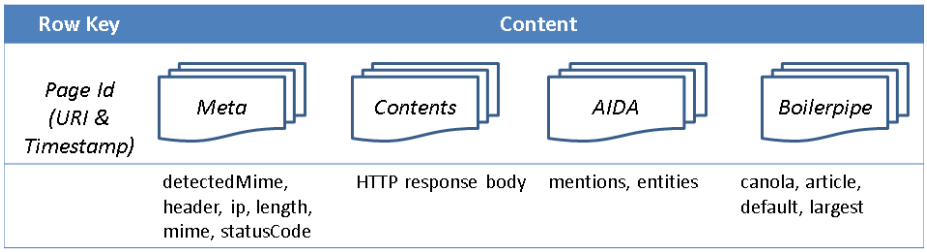

Figure 5.1: Overview of Content Table Schema

5.1.1 Scalable Storage

For flexible access, the captured Web data is loaded into a database platform, casting file-level raw data into a judiciously designed table structure. For scalability, we rely on scale-out distributed platforms. Based on a thorough study of state-of-the-art technologies, incl. novel NoSQL platforms, we selected HBase [HBase, Geor11], an open source Apache project, as the main platform. Alternatives with similar functionality and scalability would be Cassandra, Hive, Pig, Voldemort, and perhaps MongoDB. All these provide support for tables, as opposed to simpler key-value collections. In our experience, the more basic platforms that the above systems build on, most notably, Hadoop [Hadoop], are not sufficiently rich in functionality. In terms of functionality, SQL-style relational databases, such as PostgreSQL or MySQL, are a viable option as long as the data volume stays at a medium level (e.g., hundreds of Gigabytes or few Teraybytes).

The data is stored in relational tables following the storage paradigm of column orientation, which scales better than row orientation. The columns are organized in column families – physically stored close together and assuring that sparseness of the values will not hurt the performance. The overall structure of the resulting **keypath data** is a multidimensional map, where the key to each value is a tuple of the form

<row_id, column_family, column qualifier, timestamp>.

The collections are kept in several column families, each meant for a different purpose, e.g., raw content, metadata acquired at crawl, etc. Each entry is identified by its unique URL and crawl time. For subset citation, collection identifiers are created, without copying or moving data for this purpose. Instead, sub-collections are just views defined over the full contents.

HBase supports indexing its primary data, and we additionally build additional customized index structures. These are also stored in HBase tables; see below for detail. Some of the components in the extract-transform-annotate (ETA) and Analytics Layers create value-added annotations. This data is stored in the HBase tables as well, so that applications can access the full spectrum from original to semantically enriched in a unified manner, through the API of the Management Layer. Figure 5.1 gives an overview of the HBase schema used for this unified storage.

Indexing

Building indexes, for efficient access via different columns, starts with versioned Web pages in an HBase table with the page URI as key and timestamped by the time of the crawl or the version date for bulk-imported contents. So each Web page has several versions at different time points. A suite of Map-Reduce jobs run over the table data and build additional indexes in a scalable manner. The mappers in a Map-Reduce job parse the version history of a page and select the needed part of the page for insertion into an index table (e.g., type, title, etc.). Figure 5.2 gives an overview of the index tables constructed within HBase.

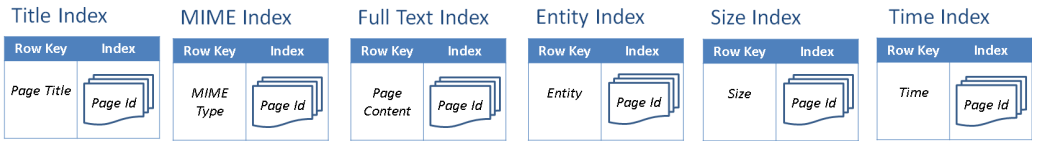

Figure 5.2: Overview of Index Tables Schema

The main configuration of the VWO uses six index tables:

- on **page titles** as a major attribute for filtering collections;
- on **MIME types** as another effective way of defining sub-collections and drilling down on application-relevant data;
- on **page contents** as a standard means of full text search
- on **entities** that occur within a page
- on **size**, to support an additional means of high-level filtering;
- on **time** for temporal reconciliation.

In addition, there are means for creating customized indexes for specific aspects of Web contents, most importantly, for **text**, **entities** and **time**. For indexing the full text of Web pages,we use the Lucene platform [Lucene, MHGo10], with a specific form of scale-out across multiple computers or cluster nodes. Alternative forms of distributed text indexing could be easily plugged in. The text index is crucial not only for efficient filtering by words and phrases, but is also the basic support for text-oriented analytics, for example, word counting or analysis of frequent n-grams, computing correlations among words and phrases, and so on.

For indexing timestamps and support a variety of temporal search predicates (see next subsection), we employ special data structures [SPNT13]. For a given timestamp t, the index returns all versions of pages that are valid (not yet overwritten by newer versions) as of t. For a given time range $[s, e]$, the index returns all versions of pages whose validity spans overlap with the interval $[s, e]$.

Querying

For search and exploration, a REST API is provided that takes HTTP get or post requests as input and returns JSON or XML objects as output. The supported queries are **conjunctions of key-value** conditions over the various columns in the HBase table schema: URI, timestamp, page title, full text, etc. Additional metadata and annotations (e.g., MIME type or mentioned entities), extracted by the ETA and Analytics Layers and stored back into HBase tables, can be queried in the same manner.

These kinds of queries can be combined with **text predicates**, implemented over the Lucene-based specific indexes. Text predicates are of the form

column: {set of keywords or phrases}

and find all matching rows where the specified column, such as page title or entities, contains all specified words or phrases. This architecture would easily allow adding further text-centric functions such as disjunctive matching, result ranking, adjacency predicates for words, relaxed proximity predicates, and so on. These are not implemented, though.

Finally, all this can be combined with **temporal predicates**, using the versioning in HBase and/or the specific temporal indexes. Options for search over time (as a basis for longitudinal contents analytics) include the following query types:

- *Containment queries of type A*: given a query interval $[s, e]$, return the items whose validity interval is completely contained in $[s, e]$.
- *Containment queries of type B*: given a query interval $[s, e]$, return the items whose validity interval completely contains $[s, e]$.
- *Stabbing queries*: given a time point t, return items that are "alive" at t; that is, items whose validity intervals contain t. This is a special case of type B containment queries where the query interval is a single point, but such queries arise frequently enough to warrant provision for added optimizations.
- *Intersection queries*: given a query interval $[s, e]$ return the items whose intervals intersect $[s, e]$.

5.1.2 VWO User Interfaces

The VWO search interface is the main entering point of temporal queries (cf. http://vwo.lawa-project.eu for details). Its main purpose is to provide a standard search interface that is well-known and familiar for all users and let them gradually explore the advanced functionalities. To this end, a dedicated entity search engine has been developed for the VWO. This engine has been designed and implemented to allow multidimensional search queries,

including text, time and entities. The technical implementation is based on a JavaScript interface communicating with the archived data stored in HBase [YHP*12].

The user interface (Figure 5.3) is arranged along the following functionalities:

- The tab bar (top) may switch between the search list and advanced visualization.
- The search bar (top, below the tab bar) may initiate temporal search in different collections of choice.
- The advanced functionalities section on the left shows the buttons and input boxes implemented for the selected collection. The "OR" button opens additional search boxes, emphasizing that topics can be defined as collections containing any of the given terms.

As a result, it is now possible to issue temporal search queries for an entity based on name, outgoing link or entity type. In comparison with the URL-driven query interface of the Wayback Machine (cf. https://archive.org/web/) or any other simple keyword-based search interface, the VWO raises querying to the entity-level. Figure 5.3 shows the results for "Mario Draghi" on the archived Wikipedia revision history.

The typical use of this central hub is the following. The user first defines her topic of interest in the search box similar to a Google query. The temporal dimension is emphasized by the "after" and "before" date boxes. By pressing the button of a collection, full text search results are displayed. The list also includes the entities detected in the document.

Given the search results, the user may now further want to learn more about the entities contained. For instance, a user may want to search for other documents mentioning the same entity, see the relation among all entities mentioned in a document, or learn about closely related entities. The entity visualization and exploration connects graphical view and full text search. First, the user may select entities of a document and initiate a call to the visualization interface. There additional search and browse operations can be conducted. Second, over the visualization interface, the user may select an entity and initiate a search in the archive for the selected entity as seen in the example of "Mario Draghi" in Figure 5.4 (based on the Yammut graph brwoser [SpBe13]).

Finally, the user might be interested in temporal trends. Here, a temporal trend analyzer tool provides real time term co-occurrence counts over the results of ad hoc queries. Given a topic characterized by a set of keywords and a time interval, we visualize and evolving set of words that have the highest increase of frequency within the topic. For example, the topical tag clouds in Figure 5.5 show the terms associated with "Mario Draghi" right before (left side) and after (right side) his appointment as president of the European Central Bank (cf. [SpBe13] for details).

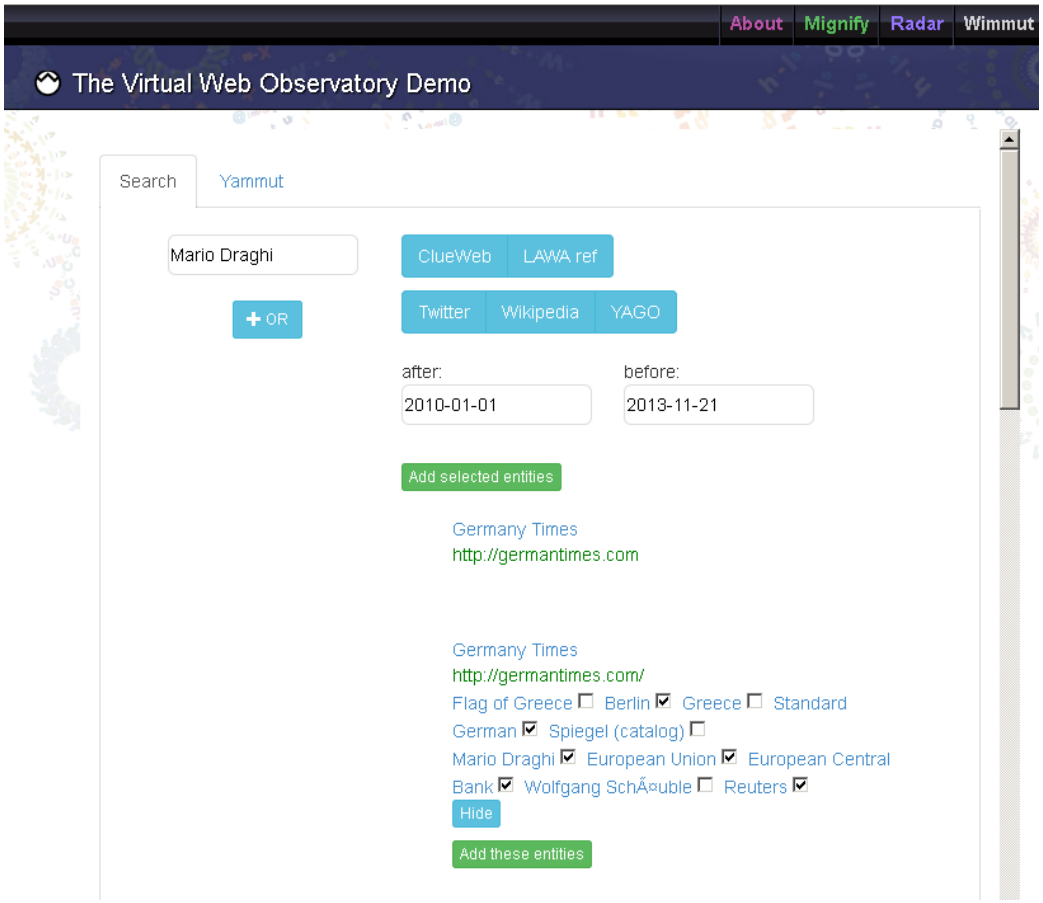

Figure 5.3: Search Result for an Entity-level Archive Query on “Mario Draghi”

5.2 Entity Type Exploration

In order to enable users to better understand the semantics of Web (archive) contents, contextual information about the entity types contained in a document are required. For instance, when wanting to understand, whether or not, politicians and sportspersons co-occur more frequently in the media during election campaigns than during the remainder of a legislative period. In order to reveal mutual dependencies among entities like in the aforementioned example, efficient and accurate entity typing is required. To this end, our type classifier HYENA (cf. Section 3.2) has been implemented as a Web application for on-the-fly type classification called “HYENA-live” [YBH*13].

As described in Section 3.2, HYENA classifies mentions of named entities onto a hierarchy of 505 types using large set of features. To this end, we build type-specific classifiers using the SVM software LIBLINEAR (cf. http://liblinear.bwaldvogel.de). Each model comes with a comprehensive feature set. While larger models (with more features) improve the accuracy, they significantly affect the applicability of the system. A single model file occupies around

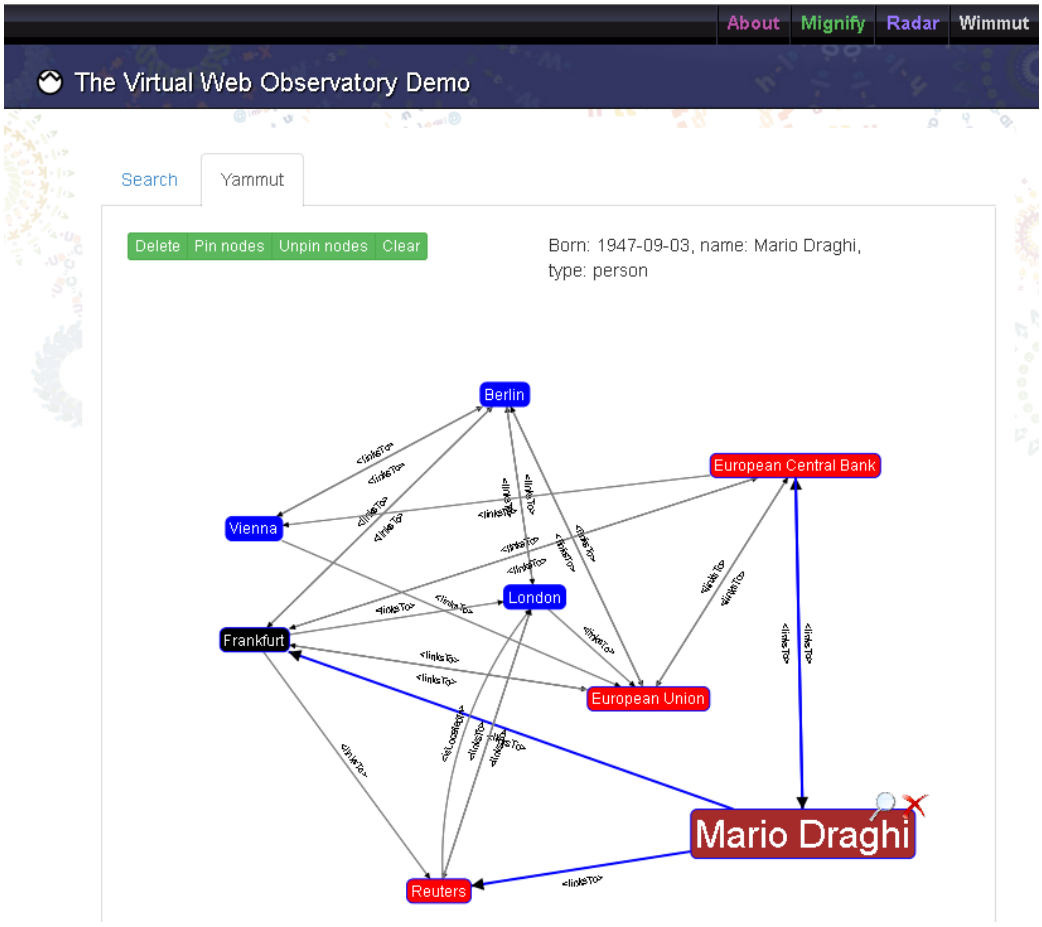

Figure 5.4: Visualization of entities related to “Mario Draghi”

150MB disk space leading to a total of 84.7GB for all models. As a consequence, there is a substantial setup time to load all models in memory and a high-memory server (48 cores with 512GB of RAM) is required for computation. An analysis showed that each single feature contributes to the overall performance of HYENA, but only a tiny subset of features is relevant for a single classifier. Hence, most of the models are extremely sparse.

5.2.1 Sparse Models Representation

There are several workarounds applicable to batch mode operations, e.g. by performing classifications per level only. However, this is not an option for on-the-fly computations. For that reason we opted for a sparse-model representation. LIBLINEAR model files are normalized textual files: a header (data about the model and the total number of features), followed by listing the weights assigned to each feature (line number indicates the feature ID). Each model file has been post-processed to produce two files:

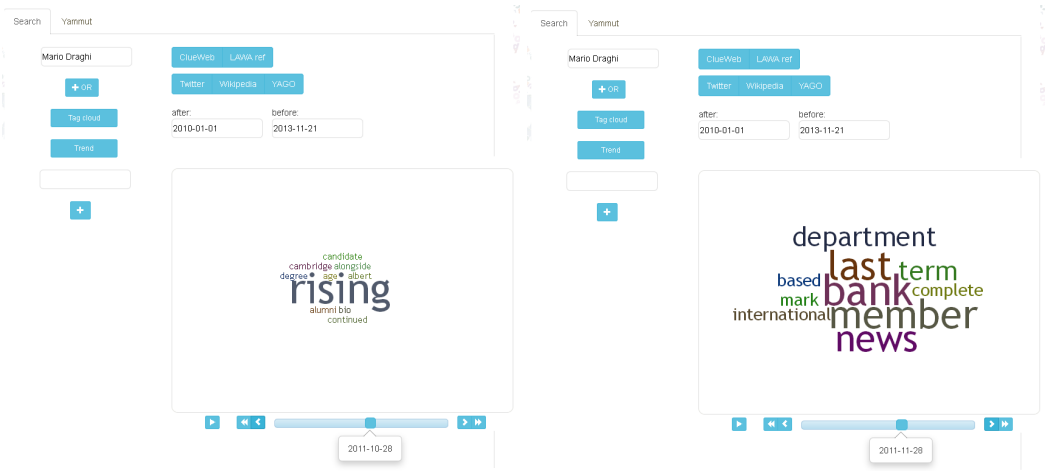

Figure 5.5: Tag cloud on topics associated with "Mario Draghi" before (left side) and after (right side) his appointment as president of the European Central Bank

- A compacted model file containing only features of non-zero weights. Its header reflects the reduced number of features.
- A meta-data file. It maps the new features IDs to the original feature IDs.

Due to the observed sparsity in the model files, particularly at deeper levels, there is a significant decrease in disk space consumption for the compacted model files and hence in the memory requirements.

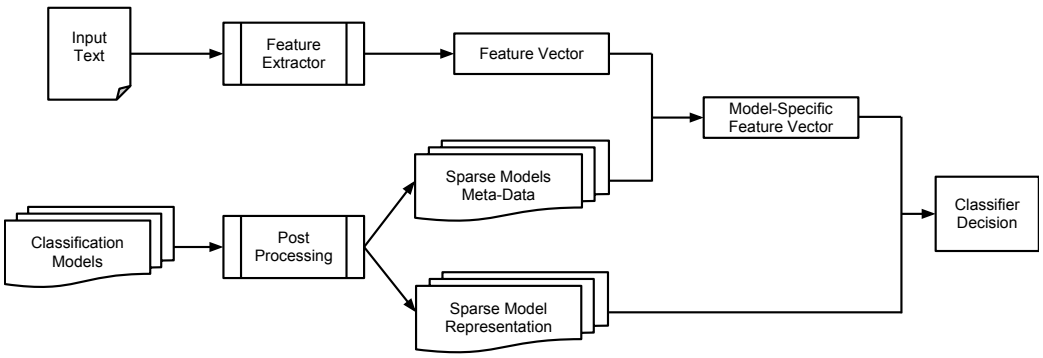

Figure 5.6: HYENA-live system architecture designed for handling sparse models

5.2.2 Sparse Models Classification

By switching to a sparse model representation the whole system architecture is affected. In particular, modified versions of feature vectors need to be generated for each classifier;

this is because a lot of features have been omitted from specific classifiers (those with zero weights). Consequently, the feature IDs need to be mapped to the new feature space of each classifier. The conceptual design of the architecture is illustrated in Figure 5.6.

5.2.3 Entity Type Visualization and Exploration

HYENA-live allows end-users to insert natural-language text for semantic type labeling of entity mentions. Thanks to its efficient implementation and compacted feature representation, the system is able to process text inputs on-the-fly while still achieving equally high precision as leading state-of-the-art implementations. It offers a Web service and an online interface where natural-language text can be inserted, which returns semantic type labels for entity mentions. To this end, it performs domain independent entity type classification under real time conditions with subsequent visualization and navigation facilities along the type-hierarchy. The system comes with a dedicated browser plug-in[1] for ad-hoc text mark-up in Web pages or alternatively offers an online interface where natural-language text can be inserted, which returns semantic type labels for entity mentions. Figure 5.7 shows the user interface of HYENA-live in a Web browser:

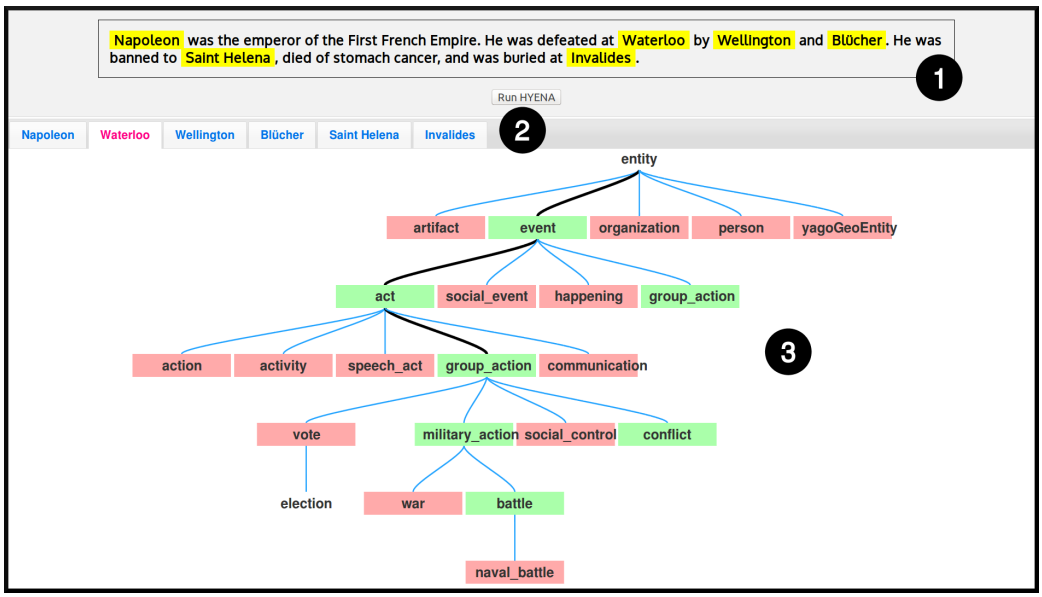

Figure 5.7: Interactively exploring the types of the "Battle of Waterloo" in the HYENA interface

(a) On top, there is a panel where a user can input any text, e.g. by copy-and-paste from news articles. We employ the Stanford NER Tagger to identify noun phrases as candidates of entity mentions. Alternatively, users can flag entity mentions by

[1] https://addons.mozilla.org/de/firefox/addon/hyena

double brackets (e.g. "Harry is the opponent of [[you know who]]"). For the sake of simplicity, detected entity mentions by HYENA-live are highlighted in yellow. Each mention is clickable to study its type classification results. If the user would like to refine the text for another query, the text annotations will automatically disappear upon editing the input area.

(b) The output of type classification is shown inside a tabbed widget. Each tab corresponds to a detected mention by the system and tabs are sorted by the order of occurrence in the input text. To open a tab, the tab header or the corresponding mention in the input area needs to be clicked.

(c) The type classification of a mention is shown as a color-coded interactive tree. While the original type hierarchy is a directed acyclic graph, for the ease of navigation the classification output has been converted into a tree. In order to do so, nodes that belong to more than a parent are duplicated. There are three different types of nodes:

- Green Nodes: referring to a class that has been accepted by the classifier. These nodes can be further expanded in order to check which sub-classes have been accepted or rejected by HYENA-live.
- Red Nodes: corresponding to a class that was rejected by the classifier, and hence HYENA-live did not traverse deeper to test its sub-classes.
- White Nodes: matching classes that have not been tested. These nodes are either known upfront (e.g. `ENTITY`) or their super class was rejected by the system.

It is worth noting that HYENA-live automatically adjusts the layouting so that as much as possible of the hierarchy is shown to the user. For the sake of explorability, this is being dynamically adjusted once the user decides to navigate along a certain (child-)node. When a node is clicked, the interface automatically tries to show as much as possible of the hierarchy to reduce the clicks required by the user. But when a node has so many children that it gets impossible to automatically expand all of them, the system leaves the decision to the user to select which node to expand.

The data transfer between the client and the server is done via JSON objects. Hence, we also provide HYENA-live as a JSON compliant entity classification Web-service. As a result, the back-end becomes easily interchangeable (e.g. by a different classification technique or a different type taxonomy) with minimum modifications required on the user interface side.

5.3 Knowledge Linking

In order to understand the mutual dependencies between Web (archive) contents and the evolving societal knowledge it represents, contextual information such as online statistics

are desirable. Statistic portals such as eurostat's "Statistics Explained"[2] provide a wealth of articles constituting an encyclopedia of European statistics. Together with its statistical glossary, the huge amount of numerical data offers an abundance of contextual information. However, identifying the most suitable statistical document given a specific Web (archive) content is a non-trivial task. This is due to the different nature of textual Web contents and statistical documents. While Web contents usually describe an event mentioning concrete entities, such as people, organizations or locations, statistical articles are fairly abstract by covering a certain topic, e.g. "'Renewable Energy Statistics", instead. As a consequence, standard approaches that "simply" create contextual information by matching keywords or keyphrases onto a thesaurus are of limited advantage given this setting. In order to overcome this shortcoming, a semantic exploitation of the content is required, which allows a proper alignment of Web contents with online statistics. To this end, we explore a hybrid approach that combines textual similarity measures with semantics captured in knowledge bases.

To this end we have developed a system called LILIANA (LIve LInking for online statistic ANAlytics) that allows live linking of Web (archive) contents with online statistics, thus, providing contextual information [SPWe13]. In order to identify the relevant statistical article(s), we raise Web (archive) contents to the entity-level so that we can align them with statistical categories of eurostat's "Statistics Explained". In addition to textual similarity measures, the relevant statistical articles can be dynamically interlinked and, thus, provide valuable contextual information. As a result, we intend to narrow and ultimately bridge the gap between numerical statistics and textual Web contents.

5.3.1 Conceptual Approach

In order to support knowledge linking for online statistics, LILIANA pursues a multi-stage procedure. This approach incorporates a semantic interpretation and linkage of Web contents. Since eurostat's "Statistics Explained" with its statistical glossary and its well-defined thesaurus represents a Wikipedia-like source, it can be interlinked with any other knowledge base such as Freebase [BEPS08], DBpedia [ABK*07] or YAGO [HSB*11]. As such, we first face an alignment problem on the ontological/entity-level [GPSW12]. Then, we raise Web (archive) contents onto the entity-level. Finally, we perform live linking and ranking based on their textual and semantic similarity. By doing so, we provide contextual information from the relevant statistical article(s). Figure 5.8 depicts the knowledge linking pipeline by LILIANA, which we explain step-by-step in the following subsections.

Knowledge Base Alignment

We have chosen the YAGO knowledge base [HSB*11] for semantic enrichment of textual Web (archive) contents. We obtained taxonomy, thesaurus and contents of "Statistics

[2] http://epp.eurostat.ec.europa.eu/statistics_explained/index.php/Main_Page

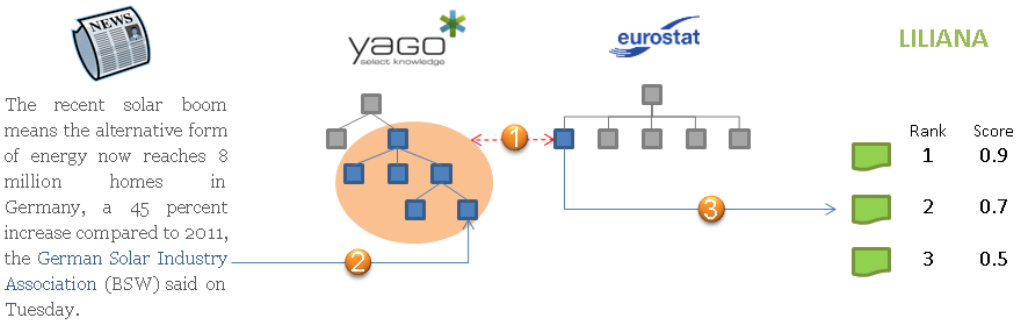

Figure 5.8: Pipeline of knowledge linking for online statistics

Explained" by crawling, thus, creating a replica for indexing and alignment. In particular, we have selected the "Statistical Themes" subsection of the hierarchy as it reflects a taxonomical structure used for classification.

Conceptually, both knowledge bases are organized in a Wikipedia-like structure. Therefore, we have undertaken in a first stage an alignment of YAGO and eurostat's "Statistics explained". However, they differ substantially in size and granularity. For instance, the "Statistical Themes" is fairly small and consists of only 40 categories in total used for classifying almost 2000 statistical articles (English contents only). On the contrary, YAGO contains almost 3 million entities classified by more than 350.000 types/categories derived from Wikipedia[3] and WordNet [Fell98]. Hence, there is no one-to-one correspondence between both knowledge bases.

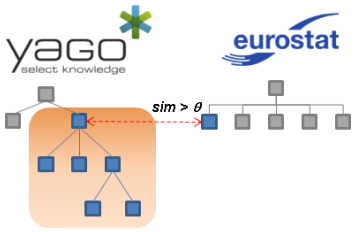

Figure 5.9: Knowledge base alignment

When aligning both knowledge bases, we started with those categories that can be directly mapped based on their textual similarity. This was do-able 12 of the 40 existing "Statistics Themes" categories. All other categories were then mapped onto their textual most similar counterpart, requiring that a user-definable similarity threshold θ is exceeded. In order to allow for the large discrepancy in size between the two knowledge bases, categories of "Statistical Themes" are mapped to a whole sub-hierarchy in YAGO (cf. Figure 5.9).

[3]http://www.wikipedia.org

Formally, the similarity between two categories c_1 and c_2 is defined as follows:

$$sim(c_1, c_2) = \frac{Jaccard\big(root(c_1), c_2\big)}{distance\big(root(c_1), c_1\big)} \tag{5.1}$$

where $Jaccard\big(root(c_1), c_2\big)$ indicates the textual similarity between category c_2 and the root category of the subtree where c_1 is located. The $distance\big(root(c_1), c_1\big)$ shows how far c_1 is from its root in terms of edges. The mapping constructed in this way contains more than 1 million category pairs.

Entity Disambiguation for Semantic Enrichment

In order to semantically enrich textual Web contents, we lift the plain text to the entity-level by detecting named entities and resolving ambiguous names. To this end we employ the AIDA entity disambiguation system [HYB*11] that maps mentions of entities onto canonical entities of the YAGO knowledge base [HSB*11]. AIDA is built on top of the Stanford NER tagger [FGMa05] to identify mentions in the input document. As a result, we obtain a semantically enriched document including entities and their types. For instance, in a document that contains the entity "European Central Bank"[4], we obtain 24 YAGO types it is associated with. Based on the previously described mapping we are able to interpret them indirectly as categories of "Statistical Themes" as well.

Live Linking and Ranking

We introduce in the following three models, which use semantic and textual features for linking and ranking.

TFIDF: Our baseline approach considers only the textual similarity between the query text and the documents in the corpus. Each document is represented as a $TFIDF$ vector in the common feature space. The relevance of a document to the query is defined as a cosine similarity between the corresponding vectors:

$$score(a, q) = cos(\vec{v_a}, \vec{v_q}) \tag{5.2}$$

As outlined in the beginning, this method has several drawbacks. The constructing of a ranked list of recommended articles requires that the entire corpus has to be scanned and the cosine similarity to each document needs to be computed. Moreover, the result can be biased towards the most populated topics the collection.

Voting: This method works entirely on the entity-level. Here, the subset of the relevant articles is defined by means of the precomputed knowledge base alignment. Depending

[4] "European Central Bank" in YAGO

on types derived from the entities in the query text, the relevant documents in "Statistical Themes" are identified. The greater the overlap between entity and document types, the higher the document is scored. Let C be a set of entity classes mentioned in the query text q and C' their counterparts in "Statistical Themes". Then, we define the score of an article a as follows:

$$score(a, q) = |c' \in categories(a)|c' \in C'| \tag{5.3}$$

where $categories(a)$ return all categories of the article a. The voting model is fully semantic. However, if article classification is based on a coarse-grained type system (e.g. "Statistical Themes"), this commonly results in many equally ranked documents.

Voting + TFIDF: This approach is a combination of the previous methods in a two-stage computation. First, we confine the relevant statistical articles based on the semantic **voting** model. In a subsequent step, we then rank the documents based on the textual overlap of **TFIDF**. Thus, this model captures both, semantic and textual similarity. Formally, the score of an article a is defined as follows:

$$score(a, q) = |c' \in categories(a)|c' \in C'| \cdot cos(\vec{v_a}, \vec{v_q}) \tag{5.4}$$

5.3.2 Live Linking – Interconnecting Web (Archive) Contents and Online Statistics

Exploring and analyzing Web (archive) contents includes finding names of people, companies, products, songs, etc. Since names are often ambiguous, disambiguation of named entities in natural language text helps to map mentions onto canonical entities allowing a semantically exploitation Web data. However, raising mentions of people, places, or organizations onto the entity level is only the first step in making use of the raw and often noisy data. Even more, contextual information (e.g. online statistics) are required to understand the complex implications between Web (archive) contents and the underlying societal event being investigated.

Figure 5.10 shows the application of the LILIANA browser plug-in[5]). In the example shown in the screenshot, the user has selected a text fragment of a news article dealing with "Rising Energy Prices – Germans Grow Wary of Switch to Renewables". Upon clicking on the right mouse button she is able to select the option "Link to Online Statistics". When doing so, the plug-in directs her to our disambiguation and link recommendation server (cf. left hand side of Figure 4) at: https://d5gate.ag5.mpi-sb.mpg.de/webliliana. The user interface shown here, comprises the following four key components:

(a) On top of the left panel, there are three buttons that can be selected. Depending on the user's choice, the underlying linking method is being selected. As a default

[5]available for download at: https://addons.mozilla.org/de/firefox/addon/liliana-linking

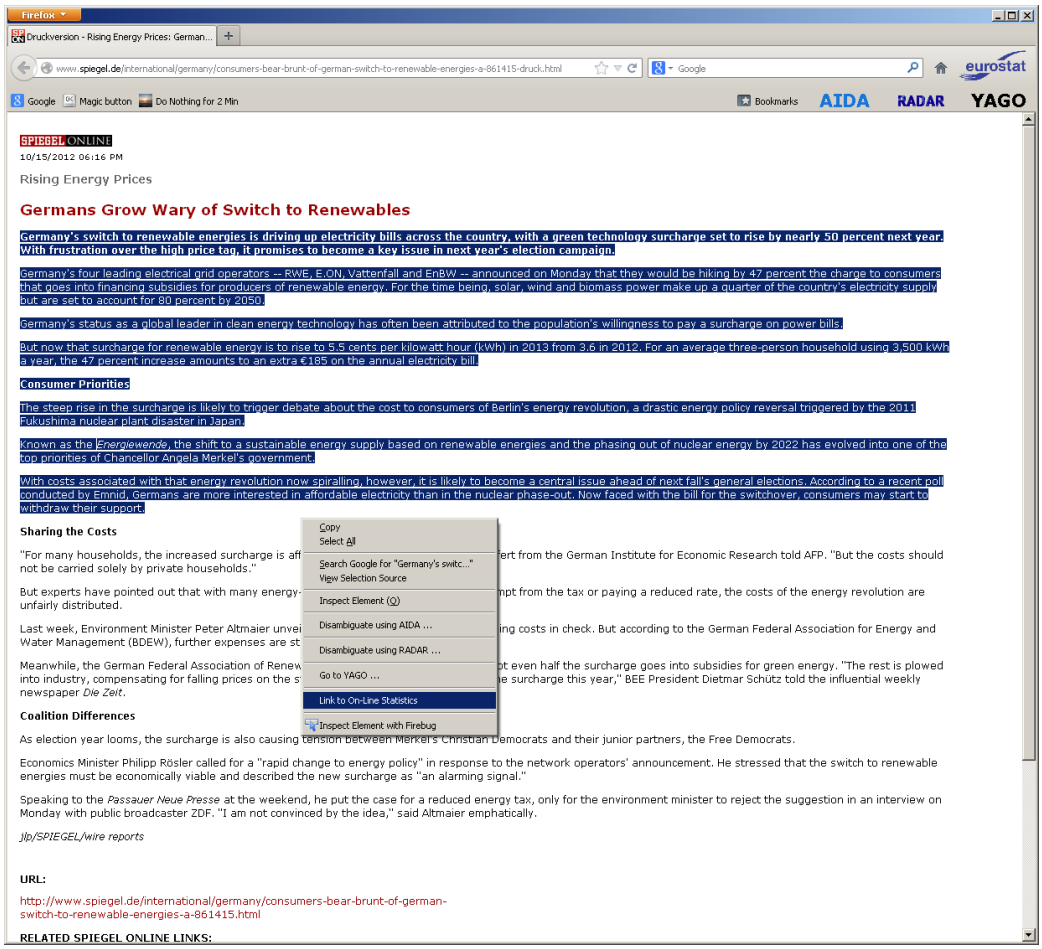

Figure 5.10: Live Linking by the LILIANA browser plug-in

setting, LILIANA employs the before mentioned hybrid approach that combines textual similarity measures with semantics captured in knowledge bases.

(b) The text panel initially contains a copy of the text that has been selected when activating the LILIANA browser plug-in the previous step. However, the user may input any text, e.g. by copy-and-paste from arbitrary Web contents, or even HTML tables. By default, the AIDA entity disambiguation system identifies noun phrases that can be interpreted as entity mentions. As this is potentially error-prone, the user can alternatively flag mentions by putting them in double brackets, e.g.: “Harry is the opponent of [[you know who]]”.

(c) The output in the upper right pane shows for each mention (in blue), the entity that has been assigned it, in the form of a clickable link. The links point to the corresponding Wikipedia articles. Alternatively, they could point to the YAGO knowledge base entries, or any comparable knowledge source in the Linked-Data world.

(d) Finally, the lower right pane contains links to the top ranked statistics articles. In order to help the user in finding the most appropriate article, the article title and the computed confidence score based on the selected linking method are shown.

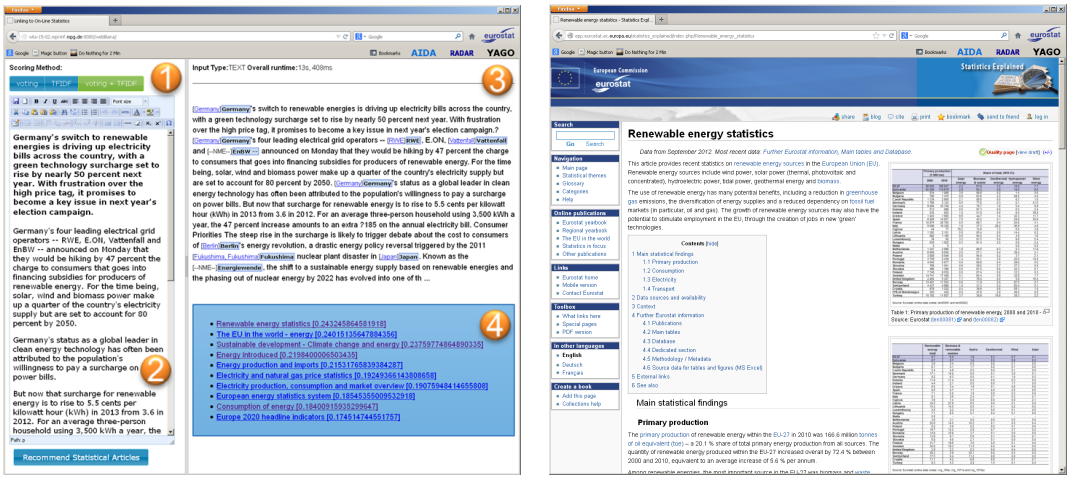

Figure 5.11: Entity-level analytics (left) and linked statistics article (right)

The outcome of live linking for the highest ranked statistical article is shown on the right hand side of Figure 5.11. Based on entity and textual similarity, LILIANA points the user in this case to the article on "Renewable energy statistics"[6].

5.4 Summary

By applying technologies for semantic enrichment and analytics of large-scale Web data, we have presented several scenarios for temporal Web analytics. The implemented services therefore primarily consist of mechanisms to ensure a comprehensive time-travel Web archive access [YHP*12, SpBe13, BRSW13]. For end users this is a great asset, as they can explore - via browsing, search, and discovery tools - Web contents on specific topics at different epochs. Topics of interested can range from broad categories such as politics, business, or sports to fine-grained topics focused on named entities such as politicians, companies or sports teams, and also historic opinions on these.

[6]http://epp.eurostat.ec.europa.eu/statistics_explained/index.php/Renewable_energy_statistics

Chapter 6

Conclusions and Outlook

The Web represents a unique source of information that a growing number of scientific communities, agencies and industries are starting to need to mine at large scale. The ability to acquire, process and mine large scale data from the Web is becoming a strategic advantage in many domains from business intelligence to epidemiological tracking and monitoring. From this perspective, knowledge on archiving, harvesting and semantically large scale Web contents is a strategic infrastructure of tomorrow's research. Even further, this has led to the launch of the Temporal Web Analytics Workshop (series) [BMSp11, BMSp11a, BMSp12, BMSp12a, BMSp13, BMSp13a]. This novel workshop (series) is co-located with the International World Wide Web conference and helped shaping a community of interest on the research challenges and possibilities resulting from the introduction of the time dimension in Web analysis.

By developing technologies for large-scale analytics on a Web data repository, we have demonstrated the proof of concept in temporal Web analytics and introduced an infrastructure for general-purpose Internet data analytics. The developed methods primarily consist of mechanisms to ensure a comprehensive time-travel Web archive access via browsing, search, and discovery tools. Topics of interested can range from broad categories such as politics, business, or sports to fine-grained topics focused on named entities such as politicians, companies or sports teams, and also historic opinions on these.

In order to systematically exploit Web contents along the before mentioned lines, this thesis has introduced a framework for temporal Web analytics. In the following, the key contributions will be summarized and perspectives on future research will be highlighted.

6.1 Contributions

As a basis for all temporal analytics, a **high-quality Web archive** is required. Though, the interpretability of contents is severely threatened by the temporal diffusion resulting

from the archiving process. Consequently, we have investigated at first how to achieve best possible archive coherence. In order to do so, our crawling strategies maximize the amount of contents that are coherent and help in identifying those contents violating coherence. Further, the underlying coherence framework is capable of dealing with proper as well as improper dated contents so that coherence can be achieved under real life conditions.

In a next step, methods are required that allow a **semantic exploitation** of this raw and often "noisy" Web contents. By developing methods that detect named entities in Web pages we have raised the entire analytics to a semantic rather than keyword-level. Difficulties arising from name ambiguities have been resolved by a disambiguation mapping of mentions (noun phrases in the text that can denote one or more entities) onto entities. In addition, we have developed methods that extract temporal facts about the entities and even **identify newly emerging latent concepts** by analyzing news articles (or social media) over time.

Finally, the before mentioned aspects need to be backed by an extensible architecture for Web-scale data analytics. To this end, a reference architecture for Web content analytics has been implemented. Further, a set of services for measuring, mining and classifying Web scale data has been deployed in order to enable **experimentally driven temporal analytics** on large scale Web datasets. Altogether, this framework enables to move up the semantic value chain in temporal Web analytics.

6.2 Future Research

Challenging computational problems arise in the real-time analytics of Big Data extracted and aggregated from heterogeneous Web sources and social media. Moreover, personalized applications are becoming accessible from any mobile device, further increasing the amount of data published on the Web. In order to understand the inherent dynamics and interplay among heterogeneous Web data, the detection and analytics of events, and the tracking of event-related entities and opinions becomes a crucial requirement. To this end, future research perspectives based on the can be classified within the areas of **Web Mining** and **Big Data**. In the following, several research questions in each of the mentioned areas will be introduced.

In the area of **Web Mining** a "natural" next step is to investigate cross-knowledge-base alignment in order to align resources beyond Wikipedia. The task of multilingual alignment is to find the best possible match over the set of instances (articles) and the set of classes (categories) of two multilingual knowledge bases. The multilingualism of sources on one hand constraints existing approaches of knowledge base alignment and provides additional information for construction a coherent alignment, from the other. Further, with respect to entity-level analytics a further study is required to classify out of knowledge base entities. The task will be relevant to "emerging" (e.g. Taifun Haiyan) or "unknown" (e.g. Marc Spaniol) entities. To this end, novel classification methods need to be investigated that can cope with (partial) context information of rarely occurring entities.

In the field of **Big Data**, new research question emerge due to the sheer amount of data. In particular new methods toward (real-time) scalable analytics of linked open data and social media contents will open-up new opportunities in cross-community entity identification. This will also incorporate entity type classification – particular with respect to "unknown" entities – in order to trace entities across various communities. The goal in this scenario will be to reveal the identity of entities independent of their (potentially different) user-name, solely based on entity type classification and context. Hence, research in that direction might greatly benefit from robust type classification of out of knowledge base entities as outlined before.

Bibliography

[ABK*07] AUER, S., C. BIZER, G. KOBILAROV, J. LEHMANN, R. CYGANIAK and Z. G. IVES: *DBpedia: A Nucleus for a Web of Open Data*. In *ISWC/ASWC*, pages 722–735, 2007.

[AGBa07] ALONSO, O., M. GERTZ and R. A. BAEZA-YATES: *On the value of temporal information in information retrieval*. SIGIR Forum, 41(2):35–41, 2007.

[Alle83] ALLEN, J. F.: *Maintaining knowledge about temporal intervals*. Commun. ACM, 26(11):832–843, November 1983.

[AlMa02] ALFONSECA, E. and S. MANANDHAR: *An Unsupervised Method for General Named Entity Recognition And Automated Concept Discovery*. In *In: Proceedings of the 1 st International Conference on General WordNet*, 2002.

[AMRW11] ALVANAKI, F., S. MICHEL, K. RAMAMRITHAM and G. WEIKUM: *EnBlogue: emergent topic detection in web 2.0 streams*. In *Proceedings of the 2011 international conference on Management of data*, SIGMOD '11, pages 1271–1274, New York, NY, USA, 2011. ACM.

[BCS*07] BANKO, M., M. J. CAFARELLA, S. SODERLAND, M. BROADHEAD and O. ETZIONI: *Open information extraction from the web*. In *IJCAI*, pages 2670–2676, 2007.

[BEPS08] BOLLACKER, K., C. EVANS, P. PARITOSH, T. STURGE and J. TAYLOR: *Freebase: a collaboratively created graph database for structuring human knowledge*. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, SIGMOD '08, pages 1247–1250. ACM, 2008.

[BGMl08] BRANK, J., M. GROBELNIK and D. MLADENIĆ: *Predicting Category Additions in a Topic Hierarchy*. In DOMINGUE, JOHN and CHUTIPORN ANUTARIYA (editors): *The Semantic Web*, volume 5367 of *Lecture Notes in Computer Science*, pages 315–329. Springer Berlin-Heidelberg, 2008.

[Blei12] BLEI, DAVID M.: *Probabilistic Topic Models*. Commun. ACM, 55(4):77–84, April 2012.

[BMSp11a] BAEZA-YATES, R., J. MASANÈS and M. SPANIOL: *The 1^{st} Temporal Web Analytics Workshop (TWAW)*. In *WWW (Companion Volume)*, pages 307–308, 2011.

[BMSp11] BAEZA-YATES, R., J. MASANÈS and M. SPANIOL (editors): *Proceedings of the 1^{st} International Temporal Web Analytics Workshop (TWAW), Hyderabad, India*. CEUR Workshop Proceedings, 2011.

[BMSp12a] BAEZA-YATES, R., J. MASANÈS and M. SPANIOL: *The 2^{nd} temporal web analytics workshop (TempWeb 2012)*. In *Proceedings of the 2^{nd} International Temporal Web Analytics Workshop (TempWeb), Lyon, France*, pages i–ii. ACM, 2012.

[BMSp12] BAEZA-YATES, R., J. MASANÈS and M. SPANIOL (editors): *Proceedings of the 2^{nd} International Temporal Web Analytics Workshop (TempWeb), Lyon, France*. ACM, 2012.

[BMSp13a] BAEZA-YATES, R., J. MASANÈS and M. SPANIOL: *The 3^{rd} temporal web analytics workshop (TempWeb 2013)*. In *Proceedings of the 3^{rd} International Temporal Web Analytics Workshop (TempWeb), Rio de Janeiro, Brazil*, pages 1033–1034. ACM, 2013.

[BMSp13] BAEZA-YATES, R., J. MASANÈS and M. SPANIOL (editors): *Proceedings of the 3^{rd} International Temporal Web Analytics Workshop (TempWeb), Lyon, France*. ACM, 2013. pp. 1033-1108.

[BNJo03] BLEI, D. M., A. Y. NG and M. I. JORDAN: *Latent Dirichlet Allocation*. J. Mach. Learn. Res., 3:993–1022, March 2003.

[BrCy00] BREWINGTON, B. E. and G. CYBENKO: *Keeping Up with the Changing Web*. Computer, 33(5):52–58, May 2000.

[BRSp13] BENCZÚR, A., P. RIGAUX and M. SPANIOL: *Report on Advances in Virtual Web Observatory*. http://www.lawa-project.eu/uploads/D5.6.pdf, October 2013, [last access: 14.02.2014].

[BRSW13] BENCZÚR, A., P. RIGAUX, M. SPANIOL, and G. WEIKUM: *LAWA Virtual Web Observatory Reference Architecture*. http://www.lawa-project.eu/uploads/D5.7.pdf, Spetember 2013, [last access: 14.02.2014].

[BuPa06] BUNESCU, R. and M. PASCA: *Using Encyclopedic Knowledge for Named Entity Disambiguation*. In *In EACL*, pages 9–16, 2006.

[CBK*10] CARLSON, A., J. BETTERIDGE, B. KISIEL, B. SETTLES, E. R. HRUSCHKA and T. M. MITCHELL: *Toward an architecture for never-ending language learning*. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*, 2010.

[CBW*10] CARLSON, A., J. BETTERIDGE, R. C. WANG, E. R. HRUSCHKA JR. and T. M. MITCHELL: *Coupled Semi-Supervised Learning for Information Extraction*. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining (WSDM 2010)*, 2010.

[ChGa00] CHO, J. and H. GARCIA-MOLINA: *The Evolution of the Web and Implications for an Incremental Crawler*. In *VLDB '00: Proceedings of the 26th International Conference on Very Large Data Bases*, pages 200–209, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[ChGa03a] CHO, J. and H. GARCIA-MOLINA: *Effective page refresh policies for Web crawlers.* ACM Transactions on Database Systems, 28(4), 2003.

[ChGa03b] CHO, J. and H. GARCIA-MOLINA: *Estimating frequency of change*. ACM Transactionson Internet Technology, 3(3):256–290, August 2003.

[CGPa98] CHO, J., H. GARCIA-MOLINA and L. PAGE: *Efficient crawling through URL ordering*. In *WWW7: Proceedings of the seventh international conference on World Wide Web 7*, pages 161–172, Amsterdam, The Netherlands, The Netherlands, 1998. Elsevier Science Publishers B. V.

[ChJu08] CHAMBERS, N. and D. JURAFSKY: *Jointly Combining Implicit Constraints Improves Temporal Ordering*. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 698–706, 2008.

[ChLi11] CHANG, C.-C. and C.-J. LIN: *LIBSVM: A library for support vector machines*. ACM Transactions on Intelligent Systems and Technology, 2(3):27, 2011.

[Cuce07] CUCERZAN, S.: *Large-Scale Named Entity Disambiguation Based on Wikipedia Data*. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 708–716, 2007.

[Cunn02] CUNNINGHAM, H.: *GATE, a General Architecture for Text Engineering*. Computers and the Humanities, 36(2):223–254, 2002.

[CYCH10] CUI, B., J. YAO, G. CONG and Y. HUANG: *Evolutionary taxonomy construction from dynamic tag space*. In *Proceedings of the 11th international conference on Web information systems engineering*, WISE'10, pages 105–119, Berlin, Heidelberg, 2010. Springer-Verlag.

[DDF*90] DEERWESTER, S., S. T. DUMAIS, G. W. FURNAS, T. K. LANDAUER and R. HARSHMAN: *Indexing by latent semantic analysis*. JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE, 41(6):391–407, 1990.

[DGRV08] DOAN, A.I, L. GRAVANO, R. RAMAKRISHNAN and S. VAITHYANATHAN (editors): *Special Section on Managing Information Extraction*. SIGMOD Record. 37(4), 2008.

[DKM*07] DUBINKO, M., R. KUMAR, J. MAGNANI, J. NOVAK, P. RAGHAVAN and A. TOMKINS: *Visualizing Tags over Time*. ACM Trans. Web, 1(2), August 2007.

[DMSW09] DENEV, D., A. MAZEIKA, M. SPANIOL and G. WEIKUM: *SHARC: Framework for Quality Conscious Web Archiving*. In *Proceedings of the 35th International Conference on Very Large Data Bases, Lyon, France*, pages 586–597. ACM Press, 2009.

[DMSW11] DENEV, D., A. MAZEIKA, M. SPANIOL and G. WEIKUM: *SHARC: Framework for Quality Conscious Web Archiving*. VLDB Journal, 20(2):183–207, 2011.

[ESFP10] EKBAL, A., E. SOURJIKOVA, A. FRANK and S. P. PONZETTO: *Assessing the challenge of fine-grained named entity recognition and classification*. In *Proceedings of the 2010 Named Entities Workshop*, NEWS '10, pages 93–101, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[FCH*08] FAN, R.-E., K.-W. CHANG, C.-J. HSIEH, X.-R. WANG and C.-J. LIN: *LIBLINEAR: A Library for Large Linear Classification*. Journal of Machine Learning Research, 9:1871–1874, 2008.

[Fell98] FELLBAUM, C. (editor): *WordNet An Electronic Lexical Database*. The MIT Press, 1998.

[FGMa05] FINKEL, J. R., T. GRENAGER and C. D. MANNING: *Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling*. In *ACL*, pages 363–370, 2005.

[FlHo02] FLEISCHMAN, M.L and E. HOVY: *Fine grained classification of named entities*. In *Proceedings of the 19th international conference on Computational linguistics - Volume 1*, COLING '02, pages 1–7, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[FSEt11] FADER, A., S. SODERLAND and O. ETZIONI: *Identifying Relations for Open Information Extraction*. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1535–1545, 2011.

[Geor11] GEORGE, L.: *HBase: The Definitive Guide*. O'Reilly Media, 1^{st} edition, 2011.

[Giul09] GIULIANO, C.: *Fine-grained classification of named entities exploiting latent semantic kernels*. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, CoNLL '09, pages 201–209, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[GPSW12] GÖBÖLÖS-SZABÓ, J., N. PRYTKOVA, M. SPANIOL and G. WEIKUM: *Cross-Lingual Data Quality for Knowledge Base Acceleration across Wikipedia Editions*. In *QDB 2012*, 2012.

[Hadoop] APACHE HADOOP. http://hadoop.apache.org, [last access: 14.02.2014].

[HaZh09] HAN, X. and J. ZHAO: *Named Entity Disambiguation by Leveraging Wikipedia Semantic Knowledge*. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 215–224, New York, NY, USA, 2009. ACM.

[HBase] APACHE HBASE. http://hbase.apache.org, [last access: 14.02.2014].

[HSB*11] HOFFART, J., F. M. SUCHANEK, K. BERBERICH, E. LEWIS-KELHAM, G. DE MELO and G. WEIKUM: *YAGO2: exploring and querying world knowledge in time, space, context, and many languages*. In *WWW (Companion Volume)*, pages 229–232, 2011.

[HSBW13] HOFFART, J., F. M. SUCHANEK, K. BERBERICH and G. WEIKUM: *YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia*. Artificial Intelligence, 194:28–61, 2013.

[HYB*11] HOFFART, J., M. AMIR YOSEF, I. BORDINO, H. FÜRSTENAU, M. PINKAL, M. SPANIOL, S. THATER and G. WEIKUM: *Robust Disambiguation of Named Entities in Text*. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 782–792, 2011.

[Karp72] KARP, R. M.: *Reducibility Among Combinatorial Problems*. In MILLER, R. and J. THATCHER (editors): *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[KiLe04] KIM, H.-J. and S.-G. LEE: *An Intelligent Information System for Organizing Online Text Documents*. Knowledge and Information Systems, 6:125–149, 2004.

[Klei03] KLEINBERG, J.: *Bursty and Hierarchical Structure in Streams*. Data Mining and Knowledge Discovery, 7:373–397, 2003.

[KlHa08] KLAMMA, R. and C. HAASLER: *Wikis as Social Networks: Evolution and Dynamics*. In *2nd SNA-KDD Workshop Social Network Mining and Analysis.*, 2008.

[KSRC09] KULKARNI, S., A. SINGH, G. RAMAKRISHNAN and S. CHAKRABARTI: *Collective Annotation of Wikipedia Entities in Web Text*. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 457–466, New York, NY, USA, 2009. ACM.

[LiOz09] LIU, L. and M. T. ÖZSU: *Encyclopedia of database systems*. New York : Springer, 2009.

[LiWe10] LING, X. and D. S. WELD: *Temporal Information Extraction*. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*, 2010.

[LiWe12] LING, X. and D. S. WELD: *Fine-Grained Entity Recognition*. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI)*, 2012.

[LLWL08] LEE, H.-T., D. LEONARD, X. WANG and D. LOGUINOV: *IRLbot: scaling to 6 billion pages and beyond*. In *WWW '08: Proceeding of the 17th International Conference on World Wide Web*, pages 427–436, New York, NY, USA, 2008. ACM.

[LMDK07] LANDAUER, T. K., D. S. MCNAMARA, S. DENNIS and W. KINTSCH: *Handbook of Latent Semantic Analysis*. Lawrence Erlbaum Associates, 2007.

[Lucene] APACHE LUCENE. http://lucene.apache.org, [last access: 14.02.2014].

[MaDa10] MAZUR, P. P. and R. DALE: *WikiWars: A New Corpus for Research on Temporal Expressions*. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 913–922, 2010.

[MAD*09] MAYFIELD, J., D. ALEXANDER, B. J. DORR, J. EISNER, T. ELSAYED, T. FININ, C. FINK, M. FREEDMAN, N. GARERA, P. MCNAMEE, S. MOHAMMAD, D. W. OARD, C. D. PIATKO, A. B. SAYEED, Z. SYED, R. M. WEISCHEDEL, T. XU and D. YAROWSKY: *Cross-Document Coreference Resolution: A Key Technology for Learning by Reading.* In *AAAI Spring Symposium: Learning by Reading and Learning to Read*, pages 65–70. AAAI, 2009.

[MaKo10] MATHIOUDAKIS, M. and N. KOUDAS: *TwitterMonitor: Trend Detection over the Twitter Stream*. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, SIGMOD '10, pages 1155–1158, New York, NY, USA, 2010. ACM.

[Masa06] MASANÈS, J.: *Web Archiving*. Springer, New York, 2006.

[McCa09] MCCARTHY, D.: *Word Sense Disambiguation: An Overview.* Language and Linguistics Compass, 3(2):537–558, 2009.

[MiCs07] MIHALCEA, R. and A. CSOMAI: *Wikify!: Linking Documents to Encyclopedic Knowledge*. In *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management*, CIKM '07, pages 233–242, New York, NY, USA, 2007. ACM.

[MDSW10] MAZEIKA, A., D. DENEV, M. SPANIOL and G. WEIKUM: *The SOLAR System for Sharp Web Archiving*. In *Proceedings of the* 10^{th} *International Web Archiving Workshop (IWAW), Vienna, Austria, September 22 - 23, 2010.*, pages 24 – 30, 2010.

[MHGo10] MCCANDLESS, M., E. HATCHER and O. GOSPODNETIC: *Lucene in Action, Second Edition: Covers Apache Lucene 3.0*. Manning Publications Co., Greenwich, CT, USA, 2010.

[MiWi08] MILNE, D. N. and I. H. WITTEN: *Learning to link with wikipedia*. In *CIKM*, pages 509–518, 2008.

[MKSR04] MOHR, G., M. KIMPTON, M. STACK and I. RANITOVIC: *Introduction to Heritrix, an archival quality Web crawler*. In *4th International Web Archiving Workshop (IWAW'04)*, 2004.

[MoMo65] MOON, J. and L. MOSER: *On cliques in graphs*. Israel Journal of Mathematics, 3(1):23–28, March 1965.

[MRSp10] MASANÈS, J., A. RAUBER and M. SPANIOL (editors): *Proceedings of the* 10^{th} *International Web Archiving Workshop (IWAW), Vienna, Austria, September 22 - 23, 2010.*, 2010.

[MSAi11] MICHEL, J.-B., Y. K. SHEN and A. P. AIDEN ET AL.: *Quantitative analysis of culture using millions of digitized books.* Science (New York, N.Y.), 331(6014):176–182, 2011.

[MVW*06] MANI, I., M. VERHAGEN, B. WELLNER, C. M. LEE and J. PUSTEJOVSKY: *Machine Learning of Temporal Relations*. In *ACL*, pages 17–18, Stroudsburg, PA, USA, 2006. The Association for Computer Linguistics.

[Navi09] NAVIGLI, R.: *Word Sense Disambiguation: A Survey*. ACM Comput. Surv., 41(2):10:1–10:69, February 2009.

[NaWi01] NAJORK, M. and J. L. WIENER: *Breadth-first search crawling yields high-quality pages*. In *In Proceedings 10th International World Wide Web Conference*, pages 114–118, 2001.

[NBGr11] NAAMAN, M., H. BECKER and L. GRAVANO: *Hip and Trendy: Characterizing Emerging Trends on Twitter*. Journal of the American Society for Information Science and Technology (JASIST), 62(5):902–918, May 2011.

[NgCa08] NGUYEN, H. T. and T. H. CAO: *Named entity disambiguation on an ontology enriched by Wikipedia*. In *RIVF*, pages 247–254. IEEE, 2008.

[NTWe11] NAKASHOLE, N., M. THEOBALD and G. WEIKUM: *Scalable knowledge harvesting with high precision and high recall*. In *WSDM*, pages 227–236, 2011.

[OlPa08] OLSTON, C. and S. PANDEY: *Recrawl scheduling based on information longevity*. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 437–446. ACM, 2008.

[PCI*03] PUSTEJOVSKY, J., J. M. CASTAÑO, R. INGRIA, R. SAURI, R. J. GAIZAUSKAS, A. SETZER, G. KATZ and D. R. RADEV: *TimeML: Robust Specification of Event and Temporal Expressions in Text*. In *New Directions in Question Answering*, pages 28–34, 2003.

[PSMe04] PANT, G., P. SRINIVASAN and F. MENCZER: *Crawling the Web*. In *Web Dynamics - Adapting to Change in Content, Size, Topology and Use*, pages 153–178. Springer, 2004.

[PSWe12] PRYTKOVA, N., M. SPANIOL and G. WEIKUM: *Predicting the Evolution of Taxonomy Restructuring in Collective Web Catalogues*. In *WebDB 2012*, 2012.

[RaNg10] RAHMAN, M. A. UR and V. NG: *Inducing Fine-Grained Semantic Classes via Hierarchical and Collective Classification*. In *COLING*, pages 931–939, 2010.

[RHC*12] RUIZ, E. J., V. HRISTIDIS, C. CASTILLO, A. GIONIS and A. JAIMES: *Correlating financial time series with micro-blogging activity*. In *Proceedings of the fifth ACM international conference on Web search and data mining*, WSDM '12, pages 513–522, New York, NY, USA, 2012. ACM.

[RMSp09] RAUBER, A., J. MASANÈS and M. SPANIOL (editors): *Proceedings of the 9^{th} International Web Archiving Workshop (IWAW), Corfu, Greece, September 30 - October 1, 2009.*, 2009.

[RoYi04] ROTH, D. and W.-T. YIH: *A Linear Programming Formulation for Global Inference in Natural Language Tasks*. In *Proceedings of the 2004 Conference on Computational Natural Language Learning (CoNLL-2004)*, pages 1–8, 2004.

[SBVW12] SPANIOL, M., A. BENCZÚR, Z. VIHAROS and G. WEIKUM: *Big Web Analytics: Toward a Virtual Web Observatory*. ERCIM News, 2012(89):23–24, April 2012.

[ScPe95] SCHÜTZE, H. and J. O. PEDERSEN: *Information Retrieval Based on Word Senses*. In *In Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, 1995.

[ScSp06] SCHULT, R. and M. SPILIOPOULOU: *Discovering Emerging Topics in Unlabelled Text Collections*. In MANOLOPOULOS, YANNIS, JAROSLAV POKORNÝ and TIMOS K. SELLIS (editors): *ADBIS*, volume 4152 of *Lecture Notes in Computer Science*, pages 353–366. Springer, 2006.

[SDM*09] SPANIOL, M., D. DENEV, A. MAZEIKA, P. SENELLART and G. WEIKUM: *Data Quality in Web Archiving*. In *Proceedings of WICOW, Madrid, Spain, April 20, 2009*, pages 19 – 26. ACM Press, 2009.

[SKWe07] SUCHANEK, F. M., G. KASNECI and G. WEIKUM: *YAGO: A Core of Semantic Knowledge - Unifying WordNet and Wikipedia*. In *16th International World Wide Web Conference (WWW 2007)*, pages 697–706. ACM, 2007.

[SMDW09] SPANIOL, M., A. MAZEIKA, D. DENEV and G. WEIKUM: *Catch me if you can: Visual Analysis of Coherence Defects in Web Archiving*. In *Proceedings of the 9^{th} International Web Archiving Workshop (IWAW), Corfu, Greece, September 30 - October 1, 2009*, pages 27 – 37, 2009.

[SpBe13] SPANIOL, M. and A. BENCZÚR: *Report on Web Analytics Technology V2*. http://www.lawa-project.eu/uploads/D4.5.pdf, September 2013, [last access: 14.02.2014].

[SPNT13] SFAKIANAKIS, G., I. PATLAKAS, N. NTARMOS and P. TRIANTAFILLOU: *Interval indexing and querying on key-value cloud stores*. 2013 IEEE 29th International Conference on Data Engineering (ICDE), pages 805–816, 2013.

[SpWe12] SPANIOL, M. and G. WEIKUM: *Tracking Entities in Web Archives: The LAWA Project*. In *WWW (Companion Volume)*, pages 287–290, 2012.

[SPWe13] SPANIOL, M., N. PRYTKOVA and G. WEIKUM: *Knowledge Linking for Online Statistics*. In *Proc. of the 59^{th} World Statistics Congress (WSC 2013), Hong Kong, SAR, China, August 22-30, 2013*, 2013.

[SSWe09] SUCHANEK, F. M., M. SOZIO and G. WEIKUM: *SOFIE: A Self-Organizing Framework for Information Extraction*. In QUEMADA, J., G. LEÓN, Y. S. MAAREK and W. NEJDL (editors): *International World Wide Web conference (WWW 2009)*, New York, NY, USA, 2009. ACM Press.

[StGe10] STRÖTGEN, J. and M. GERTZ: *TimeTrails: A System for Exploring Spatio-Temporal Information in Documents*. PVLDB, 3(2):1569–1572, 2010.

[TaCr09] TALUKDAR, P. P. and K. CRAMMER: *New Regularized Algorithms for Transductive Learning*. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part II*, ECML PKDD '09, pages 442–457, Berlin, Heidelberg, 2009. Springer-Verlag.

[TRNa09] TANG, L., S. RAJAN and V. K. NARAYANAN: *Large scale multi-label classification via metalabeler*. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 211–220. ACM, 2009.

[TWMi12] TALUKDAR, P. P., D. WIJAYA and T. MITCHELL: *Coupled Temporal Scoping of Relational Facts*. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining (WSDM)*, New York, NY, USA, February 2012. ACM.

[TZZh12] TSOUMAKAS, G., M.-L. ZHANG and Z.-H. ZHOU: *Introduction to the special issue on learning from multi-label data*. Machine Learning, 88(1-2):1–4, 2012.

[VGS*09] VERHAGEN, M., R. GAIZAUSKAS, F. SCHILDER, M. HEPPLE, J. MOSZKOWICZ and J. PUSTEJOVSKY: *The TempEval challenge: identifying temporal relations in text*. Language Resources and Evaluation, 43:161–179, 2009.

[VMS*05] VERHAGEN, M., I. MANI, R. SAURI, R. KNIPPEN, S. B. JANG, J. LITTMAN, A. RUMSHISKY, J. PHILLIPS and J. PUSTEJOVSKY: *Automating temporal annotation with TARSQI*. In *ACL '05: Proceedings of the ACL 2005 on Interactive poster and demonstration sessions*, pages 81–84, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[WCRC09] WICK, .L L., A. CULOTTA, K. ROHANIMANESH and A. MCCALLUM: *An Entity Based Model for Coreference Resolution.* In *SDM*, pages 365–376. SIAM, 2009.

[WDR*12] WANG, Y., M. DYLLA, Z. REN, M. SPANIOL and G. WEIKUM: *PRAVDA-live: Interactive Knowledge Harvesting*. In *Proceedings of the* 21^{st} *ACM Conference on Information and Knowledge Management (CIKM), Maui, Hawaii, October 29 - November 2, 2012*, pages 2674–2676, 2012.

[WDSW12] WANG, Y., M. DYLLA, M. SPANIOL and G. WEIKUM: *Coupling Label Propagation and Constraints for Temporal Fact Extraction*. In *Proceedings of the* 50^{th} *Annual Meeting of the Association for Computational Linguistics, Jeju, Republic of Korea, July 8-14, 2012*, pages 233–237, 2012.

[WHN*12] WEIKUM, G., J. HOFFART, N. NAKASHOLE, M. SPANIOL, F. M. SUCHANEK and M. A. YOSEF: *Big Data Methods for Computational Linguistics*. IEEE Data Engineering Bulletin, 35(3):46–55, September 2012.

[WLWZ11] WU, W., H. LI, H. WANG and K. Q. ZHU: *Probase: A Probabilistic Taxonomy for Text Understanding*. In *Proceedings of the VLDB Endowment*, VLDB '11, 2011.

[WNS*11] WEIKUM, G., N. NTARMOS, M. SPANIOL, P. TRIANTAFILLOU, A. BENCZÚR, S. KIRKPATRICK, P. RIGAUX and M. WILLIAMSON: *Longitudinal Analytics on Web Archive Data: It's About Time!* In *Proceedings of the* 5^{th} *biennial Conference on Innovative Data Systems Research (CIDR), Asilomar, CA, USA, January 9-12*, pages 199–202, 2011.

[WRCh97] WACHOLDER, N., Y. RAVIN and M. CHOI: *Disambiguation of proper names in text*. In *Proceedings of the fifth conference on Applied natural language processing*, ANLC '97, pages 202–208, Stroudsburg, PA, USA, 1997. Association for Computational Linguistics.

[WYQ*11] WANG, Y., B. YANG, L. QU, M. SPANIOL and G. WEIKUM: *Harvesting Facts from Textual Web Sources by Constrained Label Propagation*. In *Proceedings of the* 20^{th} *ACM Conference on Information and Knowledge Management (CIKM), Glasgow, Scotland, UK, October 24-28, 2011*, pages 837–846, 2011.

[WYZ*11] WANG, Y., B. YANG, S. ZOUPANOS, M. SPANIOL and G. WEIKUM: *Scalable Spatio-temporal Knowledge Harvesting*. In *Proceedings of the* 20^{th} *World Wide Web Conference (WWW), Bangalore, India, March 28 - April 1*, pages 143–144, 2011.

[WYZW09] WU, B., S. YANG, H. ZHAO and B. WANG: *A Distributed Algorithm to Enumerate All Maximal Cliques in MapReduce*. In *Frontier of Computer Science and Technology, 2009*, pages 45 –51, dec. 2009.

[WZQ*10] WANG, Y., M. ZHU, L. QU, M. SPANIOL and G. WEIKUM: *Timely YAGO: Harvesting, Querying, and Visualizing Temporal Knowledge from Wikipedia*. In *Proceedings of the* 13^{th} *Intl. Conference on Extending Database Technology (EDBT), Lausanne, Switzerland, March 22-26*, pages 697–700, 2010.

[YBH*12] YOSEF, M. A., S. BAUER, J. HOFFART, M. SPANIOL and G. WEIKUM: *HYENA: Hierarchical Type Classification for Entity Names*. In *Proc. of the* 24^{th} *Intl. Conference on Computational Linguistics (Coling 2012), December 8-15, Mumbai, India*, pages pp. 1361–1370, 2012.

[YBH*13] YOSEF, M. A., S. BAUER, J. HOFFART, M. SPANIOL and G. WEIKUM: *HYENA-live: Fine-Grained Online Entity Type Classification from Natural-language Text*. In *Proc. of the* 51^{st} *Annual Meeting of the Association for Computational Linguistics (ACL 2013), Sofia, Bulgaria, August 4-9, 2013*, pages 133–138, 2013.

[YCHZ12] YAO, J., B. CUI, Y. HUANG and Y. ZHOU: *Bursty event detection from collaborative tags*. World Wide Web, pages 171–195, 2012.

[YHB*11] YOSEF, M. A., J. HOFFART, I. BORDINO, M. SPANIOL and G. WEIKUM: *AIDA: An Online Tool for Accurate Disambiguation of Named Entities in Text and Tables*. In *Proc. of the* 37^{th} *Intl. Conference on Very Large Databases (VLDB 2011), August 29 - September 3, Seattle, WA, USA*, pages 1450–1453, 2011.

[YHP*12] YOSEF, M. A., J. HOFFART, N. PRYTKOVA, M. SPANIOL, G. WEIKUM, N. NTARMOS and A. BENCZÚR: *Report on Web Analytics Technology V1*. http://www.lawa-project.eu/uploads/D4.3.pdf, April 2012, [last access: 14.02.2014].

[YRAM09] YOSHIKAWA, K., S. RIEDEL, M. ASAHARA and Y. MATSUMOTO: *Jointly identifying temporal relations with Markov Logic*. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, ACL '09, pages 405–413, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[ZNL*09] ZHU, J., Z. NIE, X. LIU, B. ZHANG and J.-R. WEN: *StatSnowball: A Statistical Approach to Extracting Entity Relationships*. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 101–110, New York, NY, USA, 2009. ACM.

[ZSWe08] ZHANG, Q., F. SUCHANEK and G. WEIKUM: *TOB: Timely Ontologies for Business Relations*. In *11th International Workshop on Web and Databases 2008 (WebDB 2008)*, New York, NY, USA, 2008. ACM.

List of Figures

List of Tables

List of Abbreviations

Acronym	Description	first on
ACL	Conference of the Association for Computational Linguistics	p. 6
AIDA	Accurate Online Disambiguation of Named Entities	p. 25
API	Application Programming Interface	p. 68
BBC	British Broadcasting Corporation	p. 51
BFS	Breadth-First-Search	p. 19
BFS-FIFO	Breadth-First-Search First-In First-Out	p. 19
BFS-LIFO	Breadth-First-Search Last-In First-Out	p. 19
CALIPSO	Cloud-Aerosol Lidar and Infrared Pathfinder Satellite Observations	p. 63
CEO	Chief Executive Officer	p. 1
CIKM	Conference on Information and Knowledge Management	p. 6
CoNLL	Conference on Natural Language Learning	p. 41
Coling	International Conference on Computational Linguistics	p. 6
CoTS	Coupled Temporal Scoping	p. 54
CRF	Conditional Random Field	p. 42
DAG	Directed acyclic graph	p. 56
DFG	Deutsche Forschungsgemeinschaft (German science foundation)	p. 2
DMOZ	Directory Mozilla (aka Open Directory Project [ODP])	p. 54
DNS	Domain Name System	p. 9
ECDL	European Conference on Digital Libraries	p. 6
EDBT	International Conference on Extending Database Technology	p. 6
EMNLP	Conference on Empirical Methods in Natural Language Processing	p. 6
ERCIM	European Research Consortium for Informatics and Mathematics	p. 6

ETA	Extract-Transform-Annotate	p. 68
EU	European Union	p. 2
FIFA	Fédération Internationale de Football Association	p. 46
FIRE	Future Internet Research & Experimentation	p. 3
GB	Gigabyte	p. 73
HAC	Hierarchical Agglomerative Clustering	p. 56
HTML	Hypertext Markup Language & Experimentation	p. 9
HTTP	Hypertext Transfer Protocol	p. 70
HYENA	Hierarchical tYpe classification for Entity NAmes	p. 31
ID	Identifier	p. 68
IE	Information Extraction	p. 45
IEEE	Institute of Electrical and Electronics Engineers	p. 6
IIPC	International Internet Preservation Consortium	p. 7
ILP	Integer Linear Programming	p. 30
IP	Integrated Project	p. 3
IT	Information Technology	p. 2
JSON	JavaScript Object Notation	p. 70
LAWA	Longitudinal Analytics of Web Archive data	p. 3
LDA	Latent Dirichlet Allocation	p. 64
LILIANA	LIve LInking for online statistic ANAlytics	p. 77
LiWA	Living Web Archives	p. 2
LP	Linear Programming	p. 30
LP	Label Propagation	p. 46
LSI	Latent Semantic Indexing	p. 64
L3S	Learning Lab Lower Saxony	p. 2
MAD	Modified ADsorption	p. 46
MAP	Maximum A Posteriori	p. 30
MB	Megabyte	p. 73
MI	Mutual Information	p. 29
MIME	Multipurpose Internet Mail Extensions	p. 9
MPI-INF	Max-Planck-Institute for Informatics	p. 2
NEC	Named Entity Classification	p. 31
NED	Named Entity Disambiguation	p. 30
NELL	Never-Ending Language Learning	p. 53
NER	Named Entity Recognizer/Recognition	p. 26

NLP	Natural Language Processing	p. 29
NP-hard	Non-deterministic Polynomial-time hard	p. 30
NYT	New York Times	p. 55
PDF	Portable Document Format	p. 9
PIWO	Predicting evolution In Web catalOgues	p. 55
PRAVDA	label Propagated fAct extraction on Very large DAta	p. 46
RAM	Random Access Memory	p. 73
RDF N3	Resource Description Framework Notation3	p. 27
REST	REpresentational State Transfer	p. 70
RWTH	Rheinisch-Westfälische Technische Hochschule	p. 2
SARS	Severe Acute Respiratory Syndrome	p. 6
SOFIE	Self-Organizing Framework for Information Extraction	p. 30
STREP	Specific Targeted Research Project	p. 2
SVD	Singular Value Decomposition	p. 42
SVM	Support Vector Machine	p. 72
SQL	Structured Query Language	p. 68
TIE	Temporal information extraction	p. 53
TOB	Timely Ontologies for Business relations	p. 54
T-YAGO	Temporal YAGO	p. 25
UEFA	Union of European Football Associations	p. 29
URI	Uniform Resource Identifier	p. 69
URL	Uniform Resource Locator	p. 23
VLDB	International Conference on Very Large Databases	p. 6
VLDBJ	International Journal on Very Large Databases	p. 6
VWO	Virtual Web Observatory	p. 67
WebDB	International Workshop on the Web and Databases	p. 6
WWW	International World Wide Web Conference	p. 6
XML	Extensible Markup Language	p. 27
YAGO	Yet Another Great Ontology	p. 25