Chapter 3
Spam, Opinions, and Other Relationships: Towards a Comprehensive View of the Web Knowledge Discovery

Bettina Berendt

Abstract "Web mining" or "Web Knowledge Discovery" is the analysis of web resources with data-mining techniques such as classification, clustering, association-rule or graph-structure methods. Its applications pervade much of the software web users interact with on a daily basis: search engines' indexing and ranking choices, recommender systems' recommendations, targeted advertising, and many others. An understanding of this fast-moving field is therefore a key component of digital information literacy for everyone and a useful and fascinating extension of knowledge and skills for Information Retrieval researchers and practitioners. This chapter proposes an integrating model of learning cycles involving data, information and knowledge, explains how this model subsumes Information Retrieval and Knowledge Discovery and relates them to one another. We illustrate the usefulness of this model in an introduction to web content/text mining, using the model to structure the activities in this form of Knowledge Discovery. We focus on spam detection, opinion mining and relation mining. The chapter aims at complementing other books and articles that focus on the computational aspects of web mining, by emphasizing the often-neglected context in which these computational analyses take place: the full cycle of Knowledge Discovery, which ranges from application understanding via data understanding, data preparation, modeling and evaluation to deployment.

3.1 Introduction

Most web users are in daily contact with web mining: They profit from search engines' analyses of web pages' texts and multimedia materials for indexing and ranking these resources with respect to their relevance for users. They also see the results of these search engines taking link structure into account to determine which sites are more "authoritative" than others as evidenced by what incoming links they receive. Users receive (and often follow) buying and viewing recommendations for books, music, films, and many other items based on an analysis of their

B. Berendt (✉)
Department of Computer Science, K.U. Leuven, Celestijnenlaan 200A, 3001 Heverlee, Belgium
e-mail: bettina.berendt@cs.kuleuven.be
url: http://people.cs.kuleuven.be/~bettina.berendt/

M. Melucci, R. Baeza-Yates (eds.), *Advanced Topics in Information Retrieval*,
The Information Retrieval Series 33,
DOI 10.1007/978-3-642-20946-8_3,

own and other people's web-viewing, rating, tagging and buying behaviors. They receive "targeted advertising" based on their own and their friends' behaviors, search queries, or mail contents.

Thus, the scope of web mining—the analysis of web resources with data-mining techniques such as classification, clustering, association-rule or graph-structure methods—has by now become vast. Web mining is a fascinating approach to making sense of the Web and how people use and build it, and the results of web mining in turn shape the Web and its usage. Therefore, a basic understanding of this field is a key component of digital information literacy for everyone, and a deeper understanding is a recommendable extension to the knowledge and skills for Information Retrieval researchers and practitioners. The purpose of this chapter is to provide an introduction to the field, especially but not exclusively for readers with (some) background in Information Retrieval.

Very good textbooks now exist on various aspects of web mining. For excellent general overviews of web mining, see (Liu 2007; Baldi et al. 2003). Highly recommendable overviews of its three key subfields, content, structure and usage mining, are, respectively (Feldman and Sanger 2007; Chakrabarti 2003; Mobasher 2007).

The present chapter aims to complement these texts, which focus on the computational aspects of web mining, by emphasizing the oft-neglected context in which these computational analyses take place: the full cycle of Knowledge Discovery. Thus, we attempt to pay as much attention to the phases of application and data understanding, evaluation and deployment, as to the more technical ones of data preparation and modeling. The purpose of this chapter is to show general lines of current developments in web mining in a restricted space; we therefore strove for a choice of literature that exemplifies these well, rather than for a comprehensive survey. (Yet, we have aimed at pointing to classical papers, overviews, and other good starting points for further reading.) For reasons of space, we restrict attention to web mining that operates on texts found on the Web ("web content/text mining").

Within web content/text mining, we focus again on three application areas: spam detection, opinion mining and the extraction of relational information. These were chosen because they (a) represent important current foci of research and real-world tools and (b) illustrate the use of and design choices in key mining techniques. In addition, (c) spam detection is one of the areas of today's web mining where the three subareas meet and complement each other in innovative ways: The content of a site is obviously an important cue as to whether this site is spam or not. In addition, link structures can reveal auxiliary structures designed specifically to boost a spam site's ranking in search engines, and therefore the effective visibility of this site to the target groups. Last but not least, whether and how people query and use sites gives rich information about sites' value; for example, the large majority of people immediately recognize something as spam when they see it and do not explore such sites further.[1]

[1]Two other application areas of web mining that have received a lot of attention recently are the mining of news and the mining of social media such as blogs; for overviews of their specifics,

The chapter is structured as follows: Section 3.2 explains why we use "Knowledge Discovery" (KD) and "data mining" synonymously (but prefer the former term). It situates the field in relation to Information Retrieval and proposes an integrating model of learning cycles involving data, information and knowledge. It also gives an overview of KD phases, of modeling tasks and structures, and of the three areas of web mining: content, structure and usage mining. Section 3.3 illustrates the phases of web content mining with reference to the three example application areas. Section 3.4 focuses on a currently heavily discussed challenge to web mining (and in fact web activities as a whole): the dangers to privacy inherent in the large-scale collection, dissemination and analysis of data relating to people. Section 3.5 closes with an outlook on four specific challenges for web mining as viewed from the big-picture perspective of this chapter: context, the pervasiveness of learning cycles and prior knowledge, the question of definitional power and viewpoints, and the importance of accessible tools.

3.2 Basics and Terminology

3.2.1 From Information Retrieval to Knowledge Discovery

The classical notion of Information Retrieval (IR) assumes a person with an—often underspecified—information need, a query this person (or a helper) formulates to express this information need, and a document store towards which an IR system issues this query. The IR system is good to the extent that it returns a selection of these documents: those that are *relevant* to the information need. We shall refer to documents as *information*.[2] Thus, IR systems take as input information and the process data of the query (which is today generally expressed in free text), perform IR, and return information. This mode of interacting with human users is today extremely popular and powerful, as witnessed by the huge success of search engines.

Knowledge Discovery (KD), also known as data mining, attempts to go a step further. The aim is not to return (existing) documents and therefore information, but new *knowledge*: *valid* and *interesting* statements about the domain being described.[3] This is a semi-automatic process, in which humans interpret the *patterns* that the KD process has generated from its inputs. The inputs are typically data from (e.g., relational) databases, but also semi-structured and unstructured data/information including documents as used in IR, or knowledge as stored in knowledge bases. Patterns

see for example the proceedings of the International Conference on Weblogs and Social Media at http://www.icwsm.org and Berendt (2010).

[2]There are various concepts of "data vs. information vs. knowledge". The notions we use are designed to be maximally consistent with the uses of the term in the databases, Information Retrieval, and Knowledge Discovery literatures. For a summary, see Fig. 3.1 for details.

[3]The classical definition is "the nontrivial process of identifying valid, previously unknown, and potentially useful patterns" (Fayyad et al. 1996).

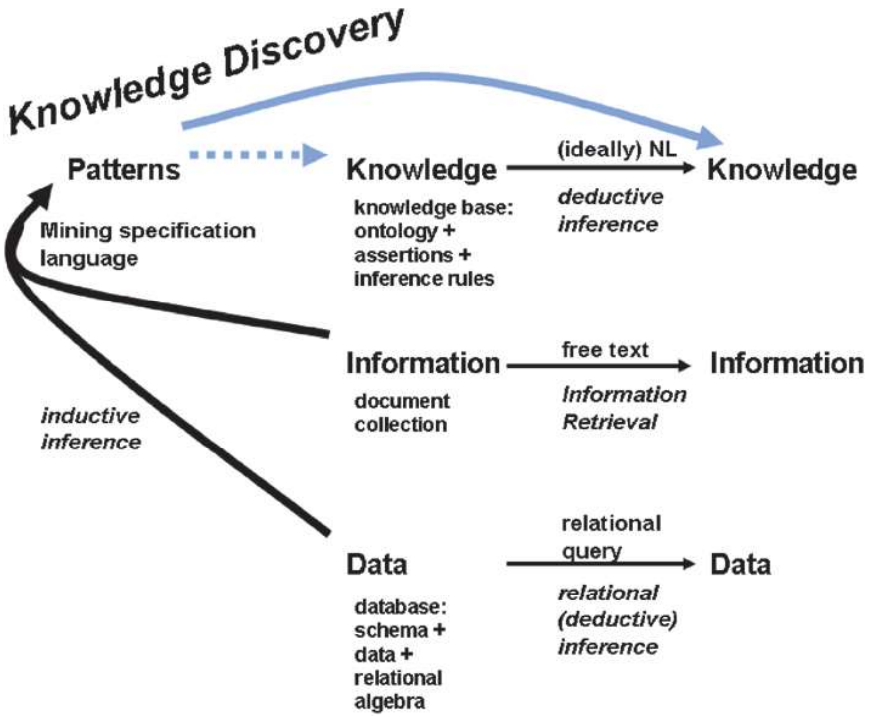


Fig. 3.1 From information retrieval to knowledge discovery

may be global over the whole investigated dataset (e.g. clusterings or classifiers) or local (e.g. association rules). In line with KD's focus on *new* patterns, it relies on *inductive* inference as opposed to data and knowledge bases whose inference is deductive.[4]

IR and KD are compared and related to each other as well as to the more fundamental database retrieval in Fig. 3.1.[5] In the interest of clarity, typical forms and terms were chosen (relational databases, ontologies, ...). Querying the stores in the middle and their content (data, information or knowledge) yields the corresponding type of output. Captions above arrows show the process-data type, captions below arrows show the retrieval/inference type. NL stands for "Natural Language". *Knowledge Discovery* is the process depicted by thick arrows at the left and top of the figure. Input are typically data or information plus the mining specification as process data. These process data are generally specified interactively and therefore often ephemeral; emerging standards aim at machine-readable, declarative, interoperable and re-usable specifications.[6] From the output patterns, people produce knowledge; this semi-automatic step is shown in grey. Often, they only produce it for one-off use; more rarely, and therefore shown by a dotted line, do they feed it directly into a knowledge base.

[4]The association of induction/abduction with new knowledge goes back to Peirce, cf. the collection of relevant text passages at http://www.helsinki.fi/science/commens/terms/abduction.html.

[5]Thanks to Ricardo Baeza-Yates for the ideas and discussions that led to this figure.

[6]See http://www.ecmlpkdd2007.org/CD/tutorials/KDUbiq/kdubiq_print.pdf, retrieved on 2010-04-07.

Today, many confluences exist between IR and KD. For example, IR systems like web search engines use KD methods such as PageRank in order to find knowledge about which web pages or sites are more "authoritative" than others (see Chakrabarti 2003 for an overview and details). The intuition is that the more authoritative pages hyperlink to a page, the more authority this page gets. This recursive notion can be assessed by a graph algorithm that returns a score for each page (the authority score, a new piece of knowledge). This score can then be used as an input to the ranking function on the pages' system-computed relevance, and this ranking function determines which pages (information) the search engine returns and in which order. Conversely, the fundamental vector-space model of text known from IR is often used to derive the word features that represent a text and from which, for example, a spam classifier is learned whose output is the new knowledge that a given document is probably spam (or not). Some analysis techniques support a range of analyses from IR to KD. For example, clustering documents may be considered mainly an interface-presentation question for IR results (cf. www.clusty.com), it may be used for extracting keyword patterns that characterize clusters (Fortuna et al. 2005), or even for semi-automatically learning a new descriptive model ("ontology") of the domain described in the documents (Fortuna et al. 2006; Berendt et al. 2010). The increasing closeness of the two areas can, for example, be seen in the contents of publications at major IR and KD conferences such as SIGIR, ECIR, SIGKDD or PKDD.

Due to these differences and commonalities, KD is an inspiring and relevant topic for everyone interested in IR, and *web mining*, i.e., KD applied to web resources, is relevant for those interested in web IR.

3.2.2 Knowledge Discovery Phases

KD rests on the application of automated methods from areas such as statistics, machine learning, graph/network theory, and visualization. For example, the applications discussed at the end of the previous section utilize clustering, classifier learning, the solving of an eigenvalue problem, and multidimensional scaling. However, KD is not a blackbox into which data, texts, etc. can be fed such that knowledge automatically emerges. Rather, it is a process which requires extensive human intellectual effort, especially in the first phases that can be described as "application, context and question understanding" and "input understanding", and in the last phases that can be described as "evaluating the results" and "acting on them, for example by deploying changes in business processes, software, and/or actions towards users/customers".

In between are phases whose problems are more well-defined and whose solution approaches therefore more formalized, such that these phases are wholly or mostly automatic: "data preparation" and "modeling". It is also by now established that the evaluation of patterns that leads to the selection of the *interesting* patterns can rely on subjective, but also on many objective criteria, which makes the "evaluation"

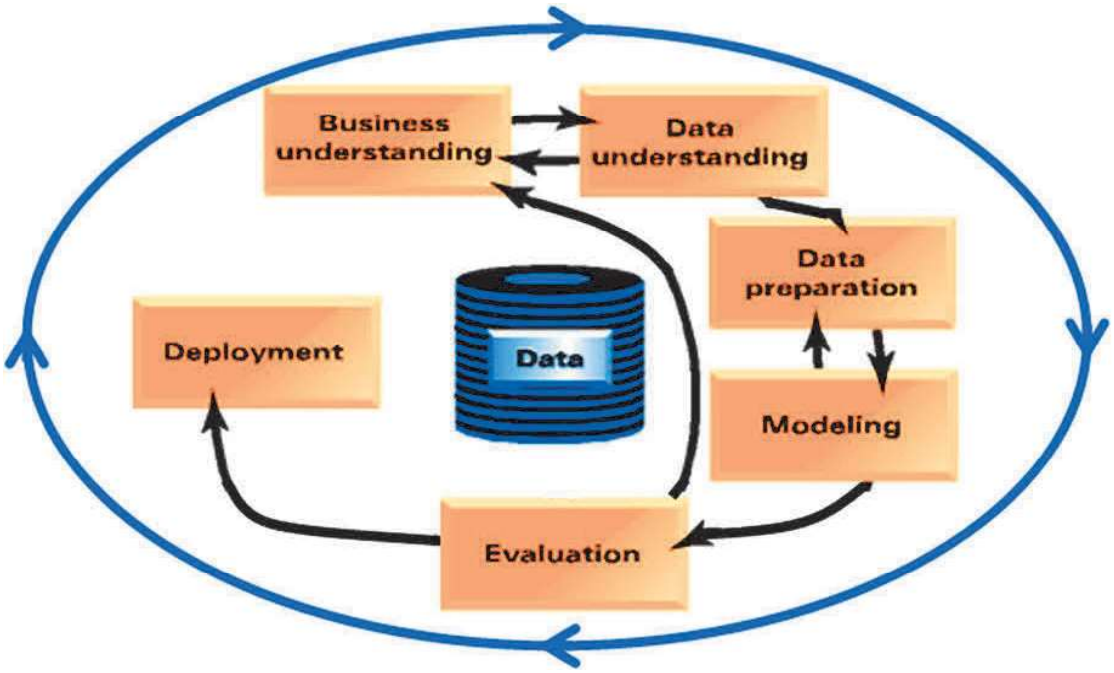


Fig. 3.2 The CRISP-DM process model

phase partially automatic (McGarry 2005). These objective interestingness criteria include extensions of the criterion of statistical significance of a result.

This common structure of the typical KD process has given rise to a process model that is widely used in the KD community: CRISP-DM (Cross-Industry Standard Process for Data Mining) (Shearer 2000), whose standard graphical summary is shown in Fig. 3.2. The phase names reflect the business focus of the authors; they may be adapted/generalized to reflect other application contexts, such as done in the previous paragraph. The diagram shows the central role of the analysed material (due to the roots of KD in the database community and the original term "knowledge discovery in databases/KDD", all input is considered as "data"). It also shows the iterative nature of the task; typically, the feedback loops ensure that several iterations of the whole process are needed to fully profit from one's data and the implicit knowledge in it.

The inner loop of the diagram corresponds to the depiction of KD in Fig. 3.1: the arrows pointing to the left there correspond to the phases from business understanding through modeling; the arrows pointing to the right correspond to the phases evaluation and increased business understanding. The pragmatic phase of deployment is abstracted away. Conversely, in Fig. 3.2, the discovered "knowledge" remains implicit in actions taken (deployment) and understanding gained.[7]

The reason is probably the business rather than knowledge-base focus of CRISP-DM. The gained and not necessarily formalized understanding may also be thought of as "knowledge that resides only in a person's head"; we will return to this in Sect. 3.2.4. This comprehensive view of KD is important because results and their

[7] Diagram adapted from http://www.crisp-dm.org/Process/index.htm.

interpretation will always depend on the context created by decisions made in all of these phases. For example, concentrating too much on methodological details of the modeling phase may lead to oversights regarding the consequences of data preparation choices.

Some people use the term "data mining" to refer only to the modeling phase, and in addition identify it with certain tasks and methods (such as association-rule discovery) used for modeling, but not others (such as classifier learning). We do not follow this terminological usage here and instead regards "data mining" as synonymous with "KD". This ensures that activities such as "web mining" or "news mining" or "multimedia mining" are understood as intended: encompassing also specific activities of data understanding, data preparation, etc. A standard book on data preparation is (Pyle 1999); specifics of data preparation for web materials will be described below. Overviews of issues in evaluating patterns by interestingness can be found in (McGarry 2005). Business understanding and deployment issues tend to be much more strongly covered in business science, e.g. (Berry and Linoff 2002, 2004), and (generally non-published) company practices—after all, KD is a core part of the business model of companies building search engines or offering personalization.

3.2.3 Modeling Tasks and Model/Pattern Structures

There are many classifications of the tasks and the methods of the modeling phase. We follow the classification proposed by Hand et al. (2001, pp. 9–15), with minor modifications.

The first distinction is that between the *kinds of representations* sought during modeling. These representations may be *global models* or *local patterns*. A *model structure* is a global summary of a data set; it makes statements about any point in the full measurement space. In contrast, *pattern structures* make statements only about restricted region of the space spanned by the variables. This structure consists of constraints on the values of the variables. Thus, in contrast to global models, a local pattern describes a structure relating to a relatively small part of the data or the space in which data could occur. Both global models and local patterns need to be fitted with the help of parameters.

The following *data-mining tasks* and associated *model or pattern structures* can be distinguished:

- exploratory data analysis with interactive, often visual methods;[8]
- descriptive modeling (density estimation, cluster analysis and segmentation, dependency modeling);

[8]While the exploration of data is often considered but one and the first step of data-mining modeling, it is also common to regard the whole of data mining (modeling) as exploratory data analysis. The reason is that in contrast to confirmatory methods, one usually does not test a previously specified hypothesis, does not collect data only for this purpose, and performs an open-ended number of statistical tests.

- predictive modeling (classification and regression): the aim is to build a model with which the value of one variable can be predicted from the known values of other variables. In classification, the predicted variable is categorical; in regression, it is quantitative;
- discovering (local) patterns and rules: typical examples are frequent patterns such as sets, sequences or subgraphs, and rules derived from them (e.g., association rules).

Finally, the data mining algorithms have *components*:

- model or pattern structure;
- score function(s) for judging the quality of a fitted model;
- optimization and search methods, to optimize the score function and search over different model and pattern structures;
- a data management strategy.

3.2.4 Learning Cycles and Knowledge Discovery

The input materials for activities such as IR or KD have no independent existence. Data, information and knowledge are typically created by people. They create this from the 'knowledge in their heads', which is the meaning of the term "knowledge" outside of Computer Science (e.g. in Psychology). To differentiate this from the CS term that denotes externalized knowledge, we refer to it as $\mathsf{Knowledge}_\mathsf{P}$. $\mathsf{Knowledge}_\mathsf{P}$ is needed to create (most) data, information or knowledge; it is needed when people embark on querying activities, and it is created or (re)shaped from the results of these information-related behaviors. These activities form a cycle that has been described by many theories of learning from Psychology, the Social Sciences, and—within computer science—knowledge management theory (Nonaka and Takeuchi 1995).

In Fig. 3.3a, this cycle is embedded into our basic model from Fig. 3.1. Note that many knowledge production and consumption activities are social, thus Figs. 3.3a and 3.3b make no commitment to an agent or bearer of $\mathsf{Knowledge}_\mathsf{P}$. Social aspects of knowledge creation are discussed in depth in knowledge management theory and other fields; they have entered web mining through the increasing relevance of social media as a source of data and an application type.

New human knowledge shows itself not only in the production of new manifestations of knowledge (such as documents) or new ways of dealing with it (such as queries), but also in new ways of creating it. As but one example of this, the left-pointing arrow at the bottom of Fig. 3.3a shows how new knowledge may change knowledge-discovery processes. This arrow completes the correspondence between our basic model and CRISP-DM: it closes the feedback loop of KD. Graphically, the outer cycles of Figs. 3.2 and 3.3a correspond to each other. Learning cycles are also at the heart of many machine learning methods within the CRISP-DM modeling phase (not shown in the figure). For example, supervised neural network training by back-propagation (Hand et al. 2001) tries one mapping from inputs to outputs, obtains corrections from the teacher, tries the next mapping, and so on.

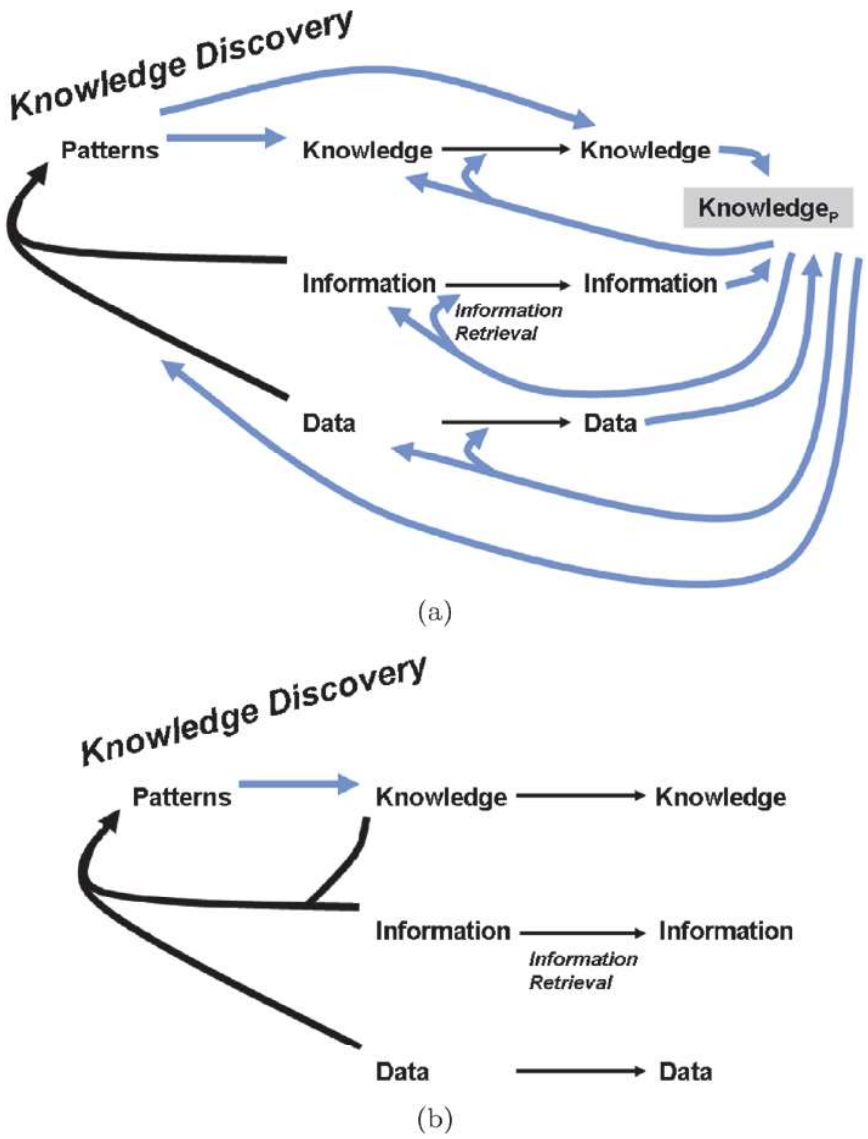


Fig. 3.3 The creation of (new) data, information and knowledge

Another learning cycle is behind the vision of *Semantic (Web) Mining* (Stumme et al. 2006), shown in Fig. 3.3b: The basic (human) knowledge creation and consumption cycle (a), Semantic Web Mining (b). All captions from Fig. 3.1 still apply, but have been omitted for clarity. Here, the left side of Fig. 3.1 is closed into a loop (rather than the right side that describes human learning). Semantic Web Mining "aims at combining the two areas Semantic Web and Web Mining. This vision follows our observation that trends converge in both areas: Increasing numbers of researchers work on improving the results of Web Mining by exploiting (the new) semantic structures in the Web, and make use of Web Mining techniques for building the Semantic Web. Last but not least, these techniques can be used for mining the Semantic Web itself" (Stumme et al. 2006, p. 124). Some current forms of Semantic Web Mining are described below, see in particular Sect. 3.3.3 on relation mining.

3.2.5 Web Mining: Content, Structure and Usage

As mentioned above, web mining is the analysis of web resources with data-mining techniques such as classification, clustering, association-rule or graph-structure methods. Technically, this means that all kinds of data/information/knowledge related to the Web may be mined. The traditional distinction between web content mining, web structure mining and web usage mining (Zaïane 1998) is based on a typology of these inputs. The distinction is still applicable today, even if the data investigated by these three areas have become more diverse.

Web content mining operates on all 'content' of the Web, i.e., that which any web user (or authorized user in the case of restricted-access resources) can access. The resources are web pages, the analysed content is the text, the multimedia elements, the metadata (such as the URL) and/or the layout of these resources. These pages can be of very different genres, which has given rise to new subfields such as blog mining or news mining. Web content mining has seen a large surge in activity in recent years, partially because rich data sources are available for everyone—often, corpora are generated by downloading a dump (e.g. of Wikipedia), crawling, or querying an archive such as news.google.com or www.digg.com.

Web structure mining operates on 'structures' found on the Web. The classic structures were those between pages (hyperlinks) and within pages (DOM tree). Increasingly, structures between people are now also being analysed. One research direction derives networks between people from transformations on (usually bipartite) graphs; an example is co-authorship analysis on the co-author links derived from author-paper links in literature databases. Another research direction utilizes the relational self-profiling of the users of social-network sites, expressed for example in "friend links". To the extent that these structures are generally visible, data are as easy to procure as for content mining.

Web usage mining operates on data describing 'usage' of the Web. The research focus changed with the design of typical sites: from click data on generally static pages, through the selection or specification of a choice from a menu on form-generated pages and sites for ranking movies etc., to free-form textual queries in an era dominated by Web-wide and internal search engines. In addition, the action type of a click is no longer only that of moving through a space of given options ("navigation"), but is indicative of a range of actions concerning the creation of content (e.g. the upload of a new or edited page in Wikipedia) and the use of content (querying it, downloading it, printing it, tagging it, ...). Data describing usage are typically recorded in web server logs, application-server logs, or program-specific logs (an example of the latter is the "history" functionality of Wikipedia). This also means that most usage data are not open; in fact, many businesses regard this as sensitive business knowledge about reactions to their Web offers. In addition, research and practice have shown that usage data such as queries or ratings cannot effectively be anonymized and therefore remain *personal data* that may threaten privacy. As a consequence, only aggregate data (if any) tend to be made public. Examples of aggregations are http://www.google.com/trends or http://buzz.yahoo.com/ for search

queries, visited sites, or search queries and resources associated with them and discussed by users in various forms. (Wikipedia content creation rests on an entirely different 'business model' than search-engine usage; it can therefore afford to be an exception).

Content mining draws on text-analysis models of IR, on (computational) linguistics, and on fields such as image analysis where non-textual content is concerned. Structure and content mining (can) draw on bibliometrics for methods of analyzing the graphs induced by people authoring papers and citing other papers, and to some extent on the textual analysis of those papers. Increasingly, methods for social network analysis from other areas such as sociology or complex-systems theory, are becoming relevant too. Usage mining analyses behavioral data and therefore draws on HCI, business studies, user research in digital libraries, and similar fields. Conversely, mining methods feed back into those disciplines and research areas.

3.3 A Short Overview of Web Content Mining on Text

The typical content to be mined is the textual content of web pages. This includes classical material such as academic or commercial web pages, pages created in social media such as fora, blogs, or social networks, and all other content that is distributed over the Web such as scientific publications in online digital libraries. The generic KD steps are adapted to the characteristics that texts in general have (text mining), to the characteristics that texts on web pages have (web text mining) and optionally to further specific characteristics induced by the material and/or questions (e.g., in blog mining).

3.3.1 Application and Data Understanding

Application understanding requires that one considers the environment(s) in which authors and readers operate, the intentions they harbor when authoring or reading, and how they produce and consume content in response to these. This can best be understood with reference to our three examples.

3.3.1.1 Spam

Spam was originally a phenomenon of email, but soon spread on the Web. Spammers want to convince gullible people to buy something.[9] They operate in an environment that is wary of spammers and that has mechanisms in place to counter

[9]Other spammers want to convince gullible people to disclose their passwords (phishing). For reasons of space, we do not investigate this further here.

the spamming methods identified so far. They face readers who in principle want to buy things, but who oppose to being defrauded (= to the intentions of the spammers) and also generally oppose to spamming in general. Thus, the problem to be solved is that of *spam detection*: determine whether something is spam or not. However, one person's spam may be another person's appreciated recommendation, cf. the continuing debate over web advertisements personalised based on someone's mailbox (Gmail), social network (Facebook Beacon), etc.

These characteristics of the application have effects on the data: Spam data on the Web tend to be (a) similar in topics and wording to non-spam commercial content, (b) appealing directly to popular instincts of bargain hunting, (c) sites that in fact are non-popular or even unknown to a general public, but that trick search engines into believing they are, in order to be found by users. Strategies for achieving (c) depend on, and change with, overarching search-engine strategies.

In the early Web, the reliance on textual content led spammers to embed popular search terms (e.g., "free") that were unrelated to the real content invisibly in the HTML body (e.g., with textcolor = backgroundcolor) or in the HTML head (`<meta>` tags). In the social web with its reliance on linkage-induced authority, a popular spamming strategy uses program-generated "link farms" (large numbers of fake, identical pages all pointing to the spam site) and, often also program-generated, incongruous contributions to fora containing these links. Further strategies are explained with reference to technical details in Sect. 3.3.3. Due to the adversarial setting, an 'arms race' has developed: continuously changing spamming and spam-detection methods.

3.3.1.2 Opinions

Opinions of web users concern, for example, movies or other products, and they are most often voiced in dedicated sites such as movielens.umn.edu or www.ciao.com or in generic social-media sites such as blogging sites, which are then indexed by systems such as www.technorati.com. Opinions are voiced in different ways, including "positive" or "negative" (or 1–5 out of five stars) concerning a product, binary or graded judgments of features/aspects of a product (such as price vs. energy efficiency), several dimensions such as "polarity" and "strength" of the opinion, or even unforeseen groupings of utterances. Opinion-giving users operate in an environment that is generally welcoming: other users are grateful for helpful opinions, for recommendations based on them, and companies view reviews and ratings as a fast and cheap way of doing market research. For various reasons, including a wish to help other users, many authors of reviews write understandably, use standard terminology and straightforward language. On the other hand, voiced opinions may not be representative due to a self-selection of authors and an often stronger tendency to write something when angry or particularly happy. Many authors are not particularly good writers, write in a hurry, or for other reasons use non-standard language. Depending on their literary abilities and preferences as well as on their opinion, authors may use irony and other stylistic means.

These characteristics of the application lead to a broad range of mining tasks, which are described in Sect. 3.3.3.

3.3.1.3 Relations

Relational assertions like "Paris is a city", "cameras are a type of consumer electronics", or "antibiotics kill bacteria" capture 'what the web community thinks about the world' or, with some liberty of generalization, 'what mankind knows'. The idea of *open information extraction from the Web* (as Banko et al. 2007 called it) is to extract all such assertions from the Web, weigh them by plausibility of being true, and from the reliable ones construct a huge and comprehensive knowledge base. The application, its setting and the approach to text data share many characteristics with opinion mining: generally well-intentioned authors and readers, the locus of information generally being at sentence rather than document level, operands of the statements that are not known a priori ("Paris", "camera", "antibiotics", "bacteria" all need to be extracted from the data), and aggregation problems. Due to the much wider semantic and conceivably also syntactic range of utterances, as well as the much more open-ended problems of aggregating inconsistencies, the problem is however more difficult.

3.3.1.4 Web Documents in General

Most web resources are published in the setting of an attention economy (Davenport and Beck 2001) and a specific aesthetics. This means that web documents are not only marked up in HTML, but that they are also designed with a view to retrievability and indexing by current-day search engines, that they present themselves not as stand-alone documents but parts of a bigger system (company pages, information space, community, ...), that they often contain advertising that finances the production of the core content, and that core content, navigation and search elements as well as advertising may be closely intertwined in graphically determined DOM-tree encodings. All these characteristics have implications on data preparation (e.g., the need to remove advertisement noise) and modeling.

3.3.2 Data Preparation

The goal of data preparation is to produce records for the chosen data model. The basic model in text mining is one in which the instances are documents, the features are terms (= words or multiword compounds), and a matrix with numerical weights that characterize each instance in terms of these features. This is the traditional vector-space model also used in IR.

To produce such a database of records, the following steps need to be taken. First, HTML and other markup as well as embedded non-textual content such as pictures needs to be removed, and the content-bearing parts need to be selected with the help of a template. Often, HTML markup gives valuable clues as to where the content-bearing parts are. A simple heuristic for such template recognition is that given a

site, the part of the DOM tree that varies across pages is content-bearing, while the parts of the DOM tree that are identical or similar across pages contain navigation bars, banner ads, etc. and can therefore be ignored. Alternatively, templates can be learned from annotated examples ("wrapper induction").

In some applications, HTML markup can be useful input for weighting or otherwise interpreting features. For example, the words of a title (marked by an HTML heading) may be given more weight for describing the topical content than other words, and text rendered invisible by HTML markup may be an indicator of spam. Table markup may be interpreted as listing relations (one row = one tuple), with the attribute names given by the column headings and the attribute values given by the table cells.

After this data cleaning, further steps are applied to obtain processable lexical/syntactical units that can be interpreted as features, and to concentrate on the relevant ones and create features that are more 'semantic' than mere word tokens that vary strongly with surface features of the text. Typical steps include the following; they are not always all performed, and they typically require some others to be completed previously.

Tokenisation consists of separating strings by word boundaries (such as spaces in Western languages). *Stopword removal* deletes words that are frequent but generally not content-bearing, such as articles and conjunctions or terms that are meaningless in the given application (for example, "Java" in a corpus of documents all dealing with this programming language). *Lemmatization* and *stemming* reduce words to a base form, eliminating differences in number, case, tense or even word class (e.g. noun vs. verb). Various forms of (usually shallow) parsing classify words into word classes (*part-of-speech tagging*: nouns, verbs, adjectives, ...) or perform *semantic role labeling* (roles include Agent, Patient or Instrument; they are labelled along with their adjuncts, such as Locative, Temporal or Manner). *Named-entity recognition* unifies occurrences with different names for the same person, place, company, etc. *Word-sense disambiguation* is one method for resolving synonymy and homonymy. This requires a lexical resource such as WordNet (Fellbaum 1998) that models that the term "key" has senses key#1 (metal device shaped in such a way that when it is inserted into the appropriate lock the lock's mechanism can be rotated) and key#4 (any of 24 major or minor diatonic scales that provide the tonal framework for a piece of music), and that the term "tonality", in its first (and only) sense. tonality#1, is synonymous with key#4.

All these steps may be used for further filtering (e.g., only nouns, and of these, only lemmata). Many steps are language-dependent, and the coverage of languages other than English varies widely. Free and/or open-source tools or web APIs exist for most steps. The result is then processed to obtain the feature weights; where a typical weight is TF.IDF: term frequency multiplied by inverse document frequency. In the basic vector-space model, a text is regarded as a bag of words, i.e., without regard to the order in which the terms occur. This lends itself most easily to the standard relational-table model of data.

Popular variations of the basic model include (i) a different unit for instances, for example, a site (instead of a document/web page) may be a spam site, or a sentence (instead of a document) may contain an opinion about a product feature, or a

blog (> document) or blogpost (< document) may be the appropriate unit for content classification; (ii) structure on terms, such as sequential order; (iii) transformed features, such as the latent "topics" of word co-occurrence in LSI (Deerwester et al. 1990), LDA (Blei et al. 2003) and similar methods; and (iv) additional features, such as linkage. The basic idea of latent topic models is to use the statistics of word co-occurrences in a corpus to detect latent "topics" that are indicated by the presence of various words without necessarily requiring the presence of each of them. Depending on the goal of the analysis, topic modeling may be used for pre-processing or modeling. Linkage as the basis of additional features also leads to a different data model: instances become nodes in a graph model.

A more extended overview and further reading on preprocessing for text mining can be found in (Feldman and Sanger 2007).

3.3.3 Modeling

In this section, several solutions to the problems introduced in Sect. 3.3.1 are described briefly.

3.3.3.1 Classification

The task of classification consists of deciding, on the basis of a description of an *instance* (e.g., a document), to which *class* (e.g., "documents about nature") it belongs. The description is in terms of *features* (e.g., the words in the English language) and the *value* of each feature for an instance (e.g., the TF.IDF weight of this word in the document). The classifier is therefore a function mapping from the space of features (e.g., the vector space spanned by the words and populated by document vectors) into the set of classes. This function is learned from a training set of instances, often in *supervised* fashion (the class value of all training-set instances is known and input/fed back to the learner). The goodness-of-fit is measured by metrics such as accuracy (percent correct) on a test set. The function is then applied to new instances, under the assumption that they have the same patterns of features and feature-class relationships. The classifier thus *predicts* the most likely class for each unseen instance. It is important that the classifier is able to generalize beyond the instances seen during training and to avoid overfitting to them. The quality of a classifier therefore hinges critically on (a) the right choice of features, (b) the type of function that the classifier can learn to separate between the classes, and (c) the set of assumptions that the learner uses to predict outputs given inputs that it has not encountered. Here, we will show three examples of (b) that have a straightforward geometrical interpretation and mention typical choices of (a) for the problem of spam detection. For a discussion of (c), the "inductive bias", we refer the reader to (Gordon and des Jardins 1995).

Three prominent types of web content mining applications classify by topical content, by opinion (∼ emotional content), or by a higher-level classification such

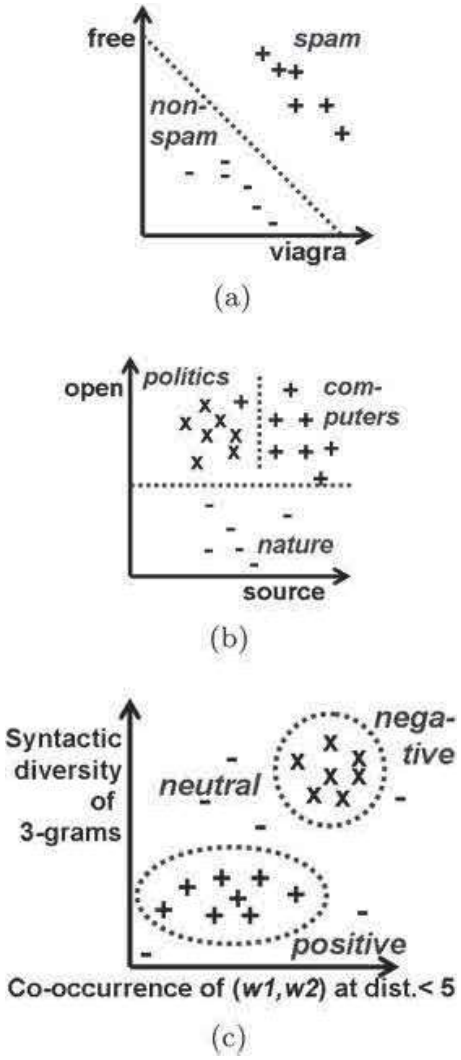


Fig. 3.4 Classification: geometric interpretation and the importance of feature and learning-algorithm choice

as "spam or not". The third is a two-class problem and can therefore be used to explain the simplest form of classifiers. In Fig. 3.4a, the fictitious feature space of a spam detector consists of the TF.IDF weights of two typical spam words; all spam instances in the training set have high values on one or both of the dimension; while all non-spam instances have lower values. The former are shown by "+" tick marks; the latter, by "−" tick marks. Thus, the function "all instances to the right of the dotted grey line are spam, all instances to the left of it are not" perfectly classifies the training instances. This type of function, a separating hyperplane, can be learned by one of the simplest learning techniques, a perceptron classifier.

Topical content classifiers (Mladenic 1998) map documents to a keyword or class name from a set or hierarchy of classes, which are often disjoint (as in the example shown in the figure), but may also overlap (this is better addressed with probabilistic learning schemes, which we do not discuss here). The fictitious feature space of a content classifier in Fig. 3.4b again consists of the TF.IDF weights of two words with good discriminatory power for the three classes of interest, but here, high values on both dimensions indicate one class, while high values on only one dimension indicate other classes. A decision-tree learner like C4.5 can learn an arbitrary tiling of the feature space into (hyper)rectangular areas each mapped to one class. The

figure also shows an example of a document misclassification incurred by restricting the boundaries to be parallel to one of the axes.

Opinion-mining classifiers of whole documents generally map documents to a small number of classes such as "positive" (opinion about the common topic, previously identified for example by topic classification), "negative" (opinion) and "neutral" (all other documents). Figure 3.4c shows a fictitious space of complex features whose values are computed from documents and auxiliary resources such as large corpora for language statistics and grammatical parsers for assigning syntactical categories to words in a sentence. These are inter-term features computed not per word/term, but on larger parts of the texts; examples of other features are given in the application examples below. In the fictitious example, instances of one class occur in the bounds of more complex geometrical shapes such as ellipses; such bounds can be learned by techniques such as Support Vector Machines.

Note that classification illustrates an interesting meeting point of Information Retrieval and Knowledge Discovery: viewed from the latter perspective, classification/classifier learning finds a global model of a space of items and a function mapping these items' features into one of n classes. The application of this classifier enables a program such as a search engine to retrieve/return only those documents that belong to the class considered "relevant" to the current user or query, which is a basic decision of an Information Retrieval process.

3.3.3.2 Spam-Detection Modeling

Spam detection is a two-class classification problem. Classifiers are trained in a supervised-learning setting that uses pages that were manually labelled as spam or non-spam as a training set. These techniques may be enhanced by semi-supervised learning (which requires fewer manually labelled examples), e.g. (Tian et al. 2007), or active learning, e.g. (Katayama et al. 2009). Content-based features have been shown to work well for detecting web spam (and also email spam). Early spam pages simply repeated attractive keywords that are not frequent in general documents ("free", "viagra", etc.) to respond to the weighting by TF.IDF that was applied by early search engines. Classifiers could therefore rely on words-as-features, or later features based on checksums and more sophisticated word-weighting techniques (Drost and Scheffer 2005). Many standard classifier-learning techniques like C4.5, Naïve Bayes or Support Vector Machines have been applied.

However, search engines soon increased the sophistication of their spam filtering, identifying statistical properties of natural human text such as individual or bigram word frequencies, and singling out outliers as possible spam. Spammers increased their linguistic sophistication by techniques such as weaving (copying an existing body of text and inserting various terms to be spammed throughout the text) or phrase stitching (gluing together individual phrases or sentences from a large corpus and inserting the spam terms). The results comply with general language statistics and therefore escape basic detection schemes.

Figure 3.5 shows an example, which combines phrase stitching (to generate content that fooled the search engine, see Fig. 3.5a); corruption of a legitimate URL

Fig. 3.5 Web spam example (page excerpts)

(the URL indexed by the search engine is the domain of a food cooperative in North America plus a script request, with the script presumable placed there by hackers); and redirection to another script-generated page that at first sight resembles a legitimate online pharmacy, but upon closer inspection reveals suspicious elements such as machine-translated text (see Fig. 3.5b). The corrupted URL was ranked 10th of several millions of results to the search query highlighted in (a) by a major search engine on 10th April 2010; Figure 3.5 (a) shows parts of the cache contents and (b) shows the result of clicking on the search result link. Thus, presumably, (a) is what the search engine sees and (b) what the user sees.

In response, new features have been proposed and successfully applied to current corpora, including inter-term features such as the distributions of term co-occurrence at various distances (Attenberg and Suel 2008) or linguistic features like the diversity of grammatical categories (parts-of-speech) in n-grams (Piskorski et al. 2008). Text-mining techniques that transform words-as-features into fewer features to capture the "latent semantics" of a corpus of texts have also been applied for spam detection, e.g. (Bíró et al. 2008). Current research also investigates HTML features, e.g. (Urvoy et al. 2006), and link features such as link farms, e.g. (Wu and Davison 2005; Drost and Scheffer 2005).

The fundamental setting of spamming is at the same time a boon and a curse for spam detection techniques: Spam only works, economically, if *huge* volumes of messages (web pages/sites, emails) are being generated; therefore most spam is generated by programs. On the one hand, such programs are easier to reverse-engineer than the unfathomable processes that lead real humans to creative writing. A reverse-engineered spamming technique, in turn, is comparatively easy to transform into an accurate classifier. On the other hand, the "adversarial" setting of spam and the arms race between spammers and spam detectors requires an ever-changing and increasing sophistication of effective techniques.[10] Insight into a range of current

[10] "New web spam techniques are introduced every 2–3 days." (Liverani 2008).

approaches to spam detection can be obtained from the proceedings of the AIRWeb ("Adversarial information retrieval on the Web") workshops.[11]

3.3.3.3 Opinion-Mining Modeling

As discussed above with reference to Fig. 3.4, opinion mining may be just another example of learning how to assign whole documents to one of a set of classes. This has many applications and can deliver important application information, cf. the study of 'what makes bloggers happy' (Mihalcea and Liu 2006) that found that the word "shopping" was the best predictor of a blog labelled by the blogger with the information that s/he was in a happy mood.

However, more often, whole documents are not the right unit of analysis for opinion mining. A typical interest in opinions voiced on the Web is that of market research formerly (and still today) done with the help of focus groups: What do (potential) customers think about a given product? Typically, this will be differentiated judgements, for example, people may like the price but not the design or usability of a product. Often, interest then focuses not only on a retrieval of all users' fine-grained opinions on all possible aspects, but also on a summary of these. Opinion mining is therefore a prime application area of analysis at a level below full documents (often sentences) and of summarization techniques. To illustrate these ideas, we will concentrate on examples from "review mining" for opinions. The *extraction* of such information is a move from the global models of classification to local patterns. In the remainder of this section, we will concentrate on local patterns and methods for discovering them.

The extraction of users' opinions on specific aspects from sentences involves two parts: the extraction of aspects that are associated with an opinion (e.g., product features) and the extraction of the opinions on these aspects. Through part-of-speech tagging, a simple form of grammatical analysis of sentences, one can determine the word classes of the words in a sentence such as noun, verb or adjective. Candidates for aspects are generally nouns or noun phrases found by a simple grammatical analysis of the sentences (e.g., "the lens" of a camera) (Hu and Liu 2004) or with the help of typical natural-language phrases indicating that the noun phrase is indeed a part of the object under discussion (e.g., "of camera", "camera has", "camera comes with", etc. for the Camera class) (Popescu and Etzioni 2005). Opinions are often expressed by adjectives ("good"), but also by verbs ("like"/"hate"). Lower bounds for frequency are used to concentrate on more important/more likely candidates. To establish that an opinion relates to an aspect, one can rely on the co-occurrence of aspect candidates with opinion candidates close to each other (in a window of a few words) (Hu and Liu 2004) or on extraction patterns modeled on the syntactic dependencies in natural language. For example, "* product" can help extract "expensive" as an opinion word via texts containing "expensive camera"; "aspect has *" can yield "problems" via "lens has problems"; "I * this product" can yield

[11] See, e.g. AIRWeb 2009 at http://airweb.cse.lehigh.edu/2009.

(a) http://www.amazon.com/Samsung-SL420-Digital-Stabilized-Black/product-reviews/B001PKTRA8

While other large manufacturers are starting to talk about launching mirrorless systems, Samsung has become the third manufacturer to actually to turn talk into tangible product. However, while Samsung is only the third party to enter the fray, enough time has passed for the other mirrorless makers to have moved on to their second-generation of cameras, including the newly-launched Panasonic G2 and G10. Between them these two cameras (which like the NX take many of their styling ideas from DSLR designs) are likely to make life pretty difficult for the Samsung. The G10 doesn't match the NX's spec but is aggressively priced while the G2 offers smarter video compression and touch-screen cleverness, which will be attractive to some. And they have the advantage of being second-generation products, with the enhanced level of refinement that this tends to bring.

(b) http://www.dpreview.com/reviews/samsungnx10/

McNally's dialogue provides clear motivation for most of the actions and sets up DAVID YAZBEK's punchy songs. Yazbek's pop score deserves closer examination. He wisely avoids the inanities of hard rock that has for the most part been rejected by theater-going audiences until now. His melodies are like pop jingles, light on tune. But his lyrics are gems, some with intricate and unexpected rhymes, as in the standout "Big-Ass Rock."

(c) http://www.theatrereviews.com/fullmonty.html

Fig. 3.6 Different reviews containing opinions (excerpts)

"hate" via "I hate this camera'. Extraction can also be helped by background knowledge about the topics (e.g., WordNet (Fellbaum 1998) assertions about properties vs. parts) (Popescu and Etzioni 2005). Further knowledge can be integrated into the process of mining via sentiment-related lexicons that list, for example, the polarities of words.

Neither the non-linguistic heuristics based on frequency and co-occurrence nor the linguistic heuristics based on parsing nor the use of lexical resources are straightforward and error-proof. Reasons include: (i) words mean different things in different contexts, which includes different polarities (e.g., "not bad", "terrific", "a . . . to die for" may all be strongly positive); (ii) language choices may depend on the domain; (iii) grammatical constructions may be very complex (cf. the many ways of expressing negation); (iv) often, authors are very creative in utilizing the generativity of language. Figure 3.6 shows examples:[12] Figure 3.6a shows customer reviews of cameras at shopping site, enhanced by author global rating of product (stars) and reader rating of the helpfulness of the review. Figure 3.6b is a part of an expert review of a camera having a total length of 30 pages. Figure 3.6c is a part of a review of a theatre performance. (The choice of domains is inspired by Williams and Anand

[12] All retrieved on 2010-04-10.

(2009); the reader is encouraged to consider the likely success of various extraction rules on these texts.)

If polarities are associated with opinions, a summarization over different sentences and/or reviews may be done. Simple counting of positive vs. negative opinions may reveal that a given text is "more positive than negative" or that most of a set of texts (e.g. those that discuss some new camera model A) are positive concerning some aspect (the price) and more so than those about a competing product B, while concerning another aspect (the lens), they are more often negative than those of B.

The extractions that are involved in opinion mining of this type can each be regarded as classification tasks: a word or group of words is to be classified as a product aspect or not, an opinion word or not, etc. However, the identification of candidates follows typical unsupervised and local mining techniques such as association-rule/co-occurrence analysis. This type of mining also shows the advantages of mining with background knowledge (see Sect. 3.2.4 on Semantic Web Mining in general and (Popescu and Etzioni 2005) for a comparison of extraction quality between methods with and without background knowledge) such as grammatical and lexical knowledge about the natural language of the texts. Finally, the examples illustrate the importance of using heuristics because these knowledge sources can never be exhaustive and/or because deep processing would be computationally inefficient.

A comprehensive recent survey of opinion mining can be found in (Pang and Lee 2008).

3.3.3.4 Relation-Mining Modeling

In a linguistically comparatively circumscribed area such as review mining, certain relations play a key role. As the examples above have shown, products *have* aspects/product features, and these *are (assessed to be)* of a certain quality. This puts a focus on two relation types—meronymy and attribution of value—that occur frequently throughout conceptual (semantic) models as well as the syntax of natural language. Mining is successful to the extent that these relation types can be comprehensively and correctly extracted from language found on the Web.

This idea lends itself to straightforward generalization: Much of linguistically expressed human knowledge takes the form of assertions, and among these, binary relations are a very common form. Common forms of relations are the aforementioned meronymy (part-whole relation), hyponymy/hyperonymy (subclass-superclass relation), and various relations expressed by transitive verbs (e.g., "eats" relates living beings to food items). Such relations are also a key building block of knowledge models such as logics (e.g. predicate logic) or ontologies. Thus, forms of mining that successfully extract correct relational assertions from a wide range of (web) sources hold the promise of 'learning all the knowledge that the world has' and converting it to machine-readable and therefore highly generative knowledge structures.

Web sources from which such relational knowledge can be learned vary widely, and current research aims at leveraging as many different forms as possible. Before investigating free text as input, we need to step back and consider more structured sources. One form are relational databases.[13] Many schemas lend themselves to straightforward extraction: For example, assume that the review site referred to in Fig. 3.6b maintains a database with the schema CAMERA (NAME, MANUFACTURER, PRICE). From this, one can learn that cameras have-a name, have-a manufacturer, and are-sold-at a price. Moreover, one can learn that a specific camera has name NX10 and manufacturer Samsung. Here, "learning" is done by a direct mapping from schema to knowledge structure ("ontology") and from record contents to "instances" of the concepts in this ontology and their properties.

This approach is limited: It has to be done separately for each database, and many databases are neither open to such knowledge extraction nor interoperable with other sources. Also, it is unclear whether the entity called "Samsung" in database A is the same as that with the same name in database B, and whether these are different from the entity "Samsung Semiconductor Inc."? The *Linked Data* (also known as Linked Open Data on the Web) initiative (for an overview, see (Bizer et al. 2009)) goes one step further: (a) Each data source provides a set of relational statements (RDF triples); as the previous paragraph has suggested, mapping existing databases to this structure is feasible; (b) these statements are interoperable by the use of the "Semantic Web stack"; (c) disambiguation of entities is performed through globally unique identifiers (URIs), jointly used namespaces, and the assertion of `<sameAs>` relations between different identifiers for one entity.[14] In principle, this type of "knowledge learning" is as simple as that from individual databases: the "learning" consists only of the combination of knowledge from different sources and the deductive inferences afforded by this combination. Major open issues are data quality, conflict resolution, and the validity of de-contextualising and re-combining atomic pieces of knowledge (the RDF triples), see Sect. 3.5.

Much relational knowledge is not expressed, or not accessible, in one of these forms, but in their equivalent in text tables. (Lists work analogously; for clarity, in the following all explanations refer to tables.) The semi-structured nature of HTML gives clear cues for information extraction from tables. For example, `<th>` cells typically contain attribute names, `<td>` cells contain attribute values (in the same order), and a caption or title (marked as a heading, in bold font, etc.) immediately preceding or following the table the relation name or description. In real-world examples of HTML, not all tables represent relations, since tables are also used for formatting, and not all relations are represented in tables. Yet, as long as a sufficient

[13] These are typical examples of humans having fed their knowledge into machine-readable *data* as described by the left-pointing arrows at the bottom of Fig. 3.3.

[14] RDF triples (just like database content) do not need to be authored by technology-savvy users: Web forms are a convenient way to collect structured data from laypeople. Thus, for example, social networks generate and hold masses of personal data in table/RDF form and accessible over the Web. Examples are the FOAF export of Livejournal (http://www.livejournal.com/bots/) and exporter tools for Facebook (http://www.dcs.shef.ac.uk/~mrowe/foafgenerator.html), Twitter (http://sioc-project.org/node/262) or Flickr (http://apassant.net/home/2007/12/flickrdf).

number of pages follow the same conventions (this is the case for most content management systems pages), wrapper induction can learn a mapping from structuring (XML) or formatting (HTML/CSS) elements to relational structure (Kushmerick et al. 1997).

Tables tend to be produced when the author wants to express knowledge about a larger number of instances of one relation (e.g., the name, director, release date, etc. of a collection of films), but rarely when only one or a small number of instances are described. Particularly in the latter case, free text is the more common medium of expression.[15] Again, natural language shows many regularities, and text mining approaches for leveraging these can be grouped by the type of regularity.

The first is purely statistical and based on co-occurrence information. Thus, the observation that most sentences (or passages or documents) that mention "Samsung NX10" also contain "camera", while only a fraction of text units that mention "camera" also contain "Samsung NX10", and that the same pattern holds for "Canon EOS", can be used to infer that both Samsung NX10 and Canon EOS are a subclass or instance of camera.[16] Such *association rules* however need not express taxonomical information; they may also express other types of conceptual associations (such as "people who liked this also liked that").

A second type of regularity therefore leverages typical ways of expressing relations in a given natural language. Examples have been given in the section on opinion mining; probably the best-known of these lexico-syntactic patterns (or templates) are the "Hearst patterns": two noun-phrases A and B linked by "such A as B" or "B and other A" often signal that B is a kind of A (Hearst 1992). Patterns such as "A such as B, C and D" identify several relation instances at once.

Such patterns are *lexico*-syntactic, i.e., they rely on the lexicon of a given language and its meaning (i.e., special relations like hyponymy/hyperonymy). A third type of regularity rests only on the grammar of a given natural language and can therefore capture different types of relations. Thus, the above-mentioned parts-of-speech tagging label constituents of a sentence with (e.g.) noun1, verb and noun2, and a simple grammatical parser can identify that the sentence is in the active voice. This can be interpreted as asserting the relational information verb(noun1, noun2). An example is "[the] camera takes pictures".

Templates can be hand-crafted inputs to the mining process, or they can be learned from corpora in supervised fashion (if sentences are labelled) or with less than full supervision (see below). Statistical patterns and natural-language templates (with hand-crafted extraction patterns) are used in *ontology learning from text* (Maedche and Staab 2001; Buitelaar et al. 2005).

---

[15] These are typical examples of humans having fed their knowledge into machine-readable *information* as described by the left-pointing arrows in the middle of Fig. 3.3.

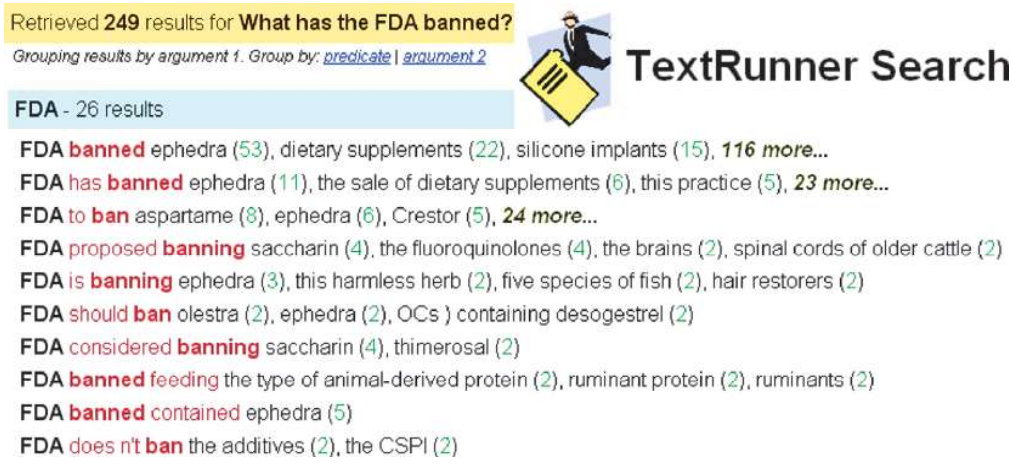
[16] The decision whether to treat something as a concept (standing in a subclass relation to another concept) or as an instance (standing in an instance-of relation) is not always straightforward, handled differently by different extraction methods, and even treated differently by different logics and ontology formalisms. For reasons of space, we will therefore not investigate this differentiation.

The transition from *text* to *text on the Web* implies a number of new challenges and new opportunities: First, corpora need to be created; for this, many current approaches utilize general-purpose search engines queried with seed patterns. Second, the magnitude of the Web means that there will be redundant and also conflicting information. The usual response to this is to rely on the massive redundancy of the Web and assume that frequently asserted statements are correct and infrequent statements contradicting them can be ignored. The huge size of the Web also means that extraction techniques can successfully operate in a *self-supervised* fashion. For example, candidates for A, B, C and D standing in a hyponomy relation as stated above may be found by a search-engine query with a seed Hearst pattern ("A such as B, C and D"). To (a) validate these candidates as "true" examples and (b) obtain more extraction patterns, further steps of the process can investigate frequency statistics of the candidates and use them in a new query (e.g. "+A +B +C +D": enforcing the presence of all four in the results). Such self-supervised learning is the core of "open information extraction from the Web" as performed by, for example (Etzioni et al. 2004; Banko et al. 2007). Two examples of live systems that learn arbitrary relations from free texts (both currently displaying the results of pre-indexed fixed corpora) are shown in Fig. 3.7.[17] In Fig. 3.7a, Textrunner (operating on a web corpus) is shown with a query specifying the first argument and the relation, and results showing different instantiations of the second argument and different forms of the relation. Figure 3.7b shows Yahoo! Quest (operating on a corpus of questions and answers by users), a query and resulting clustered entities. Clustered entities are associated as follows with the query: (i) kinds of the query class, (ii) verbs (relations) associated with it, or (iii) nouns (often the other argument of the relation) associated with it. Yahoo! Quest is a good example of how a browsing-oriented display of various extracted relations invites users to explore a corpus.

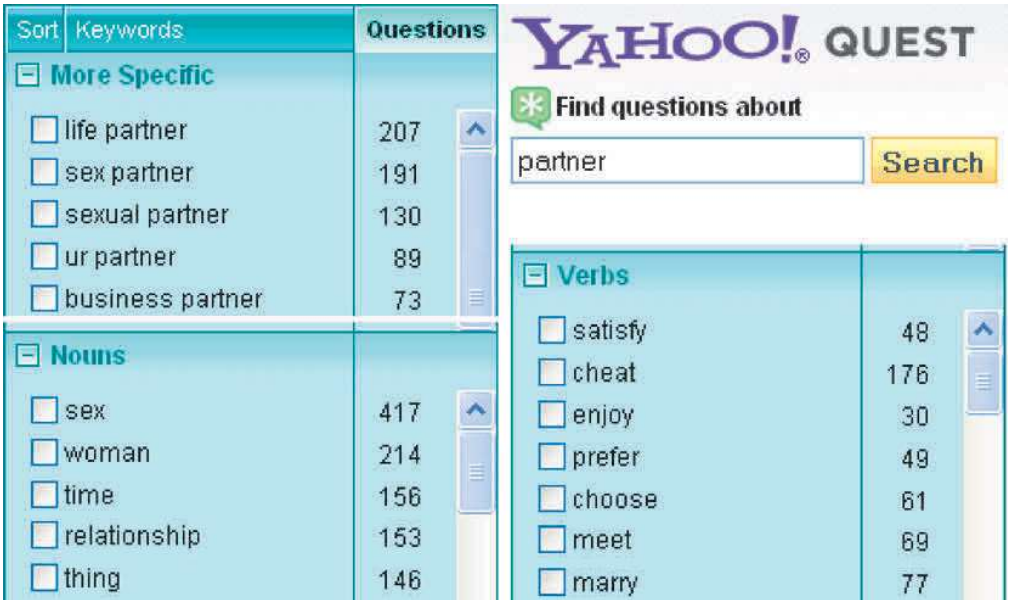
Mining the Web in this frequency- and feedback-driven way may be said to profit from the *constraints* that each document implicitly defines on the others (since documents confirm or do not confirm others, and since some documents determine the extraction patterns for others). Not only can different documents be combined such that the knowledge mined from them is 'self-constraining', the same can also be done with different formats (for example, (Cafarella 2009) combines mining hyponymy relations (Etzioni et al. 2004), other relational triples (Banko et al. 2007) and HTML tables, and a purely statistical association miner). Constraints can also come from existing knowledge bases; for example, the knowledge that a digital camera has exactly one "number of megapixels", but may have several lenses is readily expressible in an OWL ontology and can help evaluate multiple findings from free-text mining, and coupling the knowledge from different (parts of) ontologies may provide further constraints for consistency checking, cf. for example (Matuszek et al. 2005; Carlson et al. 2010).[18] A promising idea is to leverage the

[17] Both retrieved on 2010-04-10.

[18] These are typical examples of humans having fed their knowledge into machine-readable *knowledge* as described by the left-pointing arrows at the top of Fig. 3.3a, and into the form that can be used for automatic consistency checking in the sense of Fig. 3.3b.



(a) http://turingc.cs.washington.edu:7125/TextRunner



(b) http://quest.sandbox.yahoo.net

Fig. 3.7 Interfaces for relation extraction (excerpts)

knowledge inherent in very large and highly quality-controlled resources such as Wikipedia[19] and Cyc[20] (whose modes of quality control are very different, with one relying on distributed editing and "the wisdom of the crowds" and the other on highly skilled knowledge engineers), e.g. (Sarjant et al. 2009).

In sum, current approaches that couple mining from different resources (data, information, knowledge) and that use different feedback loops for validity checking and evaluating, together are beginning to realize the vision of Semantic Web Mining (see Figs. 3.3a and 3.3b).

[19] See http://www.wikipedia.org.

[20] See http://www.cyc.com.

3.3.4 Evaluation and Deployment

The typical deployment scenario for spam detection methods is close to the user: in the case of email, at the mailserver of an organization and more typically operating on individual users' mailboxes (to avoid problems of false positives, i.e., users losing legitimate mail). In the case of web spam, good deployment environments are again mail programs (to warn of sites/pages announced in emails that are spam, or to classify emails as spam based on them containing links to web spam), search engines (to avoid indexing spam), and possibly special-purpose programs that identify, for example, online-banking spam.

The preponderance of spam and the large consensus over most instances of spam has led to filters in email readers and search-engine indexers that automatically classify something as a spam candidate. However, in the environment of email readers (which is more personal than that of indexers), the subjectivity of the notion of spam makes many spam filter developers include "active learning" elements into the spam classification: a checkbox that allows users to file, delete, and report something as spam (or not) and to remove automatically generated spam flags.

Yet, for all deployment scenarios, evaluation needs to ask "A key objective is to determine if there is some important business issue that has not been sufficiently considered [even if the model is of high quality from a data analysis perspective]."[21] A good example can again be found in the web spam domain. As mentioned above, including hidden or invisible text that is unrelated to the spam site's real content, but is judged as attractive for searchers, is a popular strategy for spammers. Text can be hidden in HTML code such as "no frame" or "no script" sections or ALT attributes. However, hidden text is not always a sign of spamming; it can also be used to enhance accessibility. In fact, ALT text equivalents for images, frames, scripts, etc. are *prescribed* by the W3C's accessibility guidelines (W3C 2000). Thus, a classifier that relies heavily on such attributes for detecting spam—whether it be prescribed as a heuristic or machine-learned—is likely to produce false positives, which may result in the non-indexing of legitimate, accessible sites. Such tradeoffs and their (financial, societal, etc.) costs have to be considered in order to evaluate whether and how learned models or patterns should be deployed.

The deployment of results may range from (annotated) trend meters such as those by www.blogpulse.com, where the mere mention of something, i.e., a topical analysis, is already interpreted as 'the opinion that something is interesting/hot', to detailed positive/negative opinions-on-product-features barometers. Opinion spam, designed to push or damage brand reputations, may be a severe problem whose frequency is only very partially known. Another problem may be privacy issues. For example, people may object to the re-purposing, for marketing purposes, of content that they wrote as a forum post designed to help other users. Similar issues may arise in the deployment of relation-mining results. In addition, the de-contextualisation and re-combination of atomic relational statements may endanger consistency and

[21] See http://www.crisp-dm.org/Process/index.htm, retrieved on 2010-04-10.

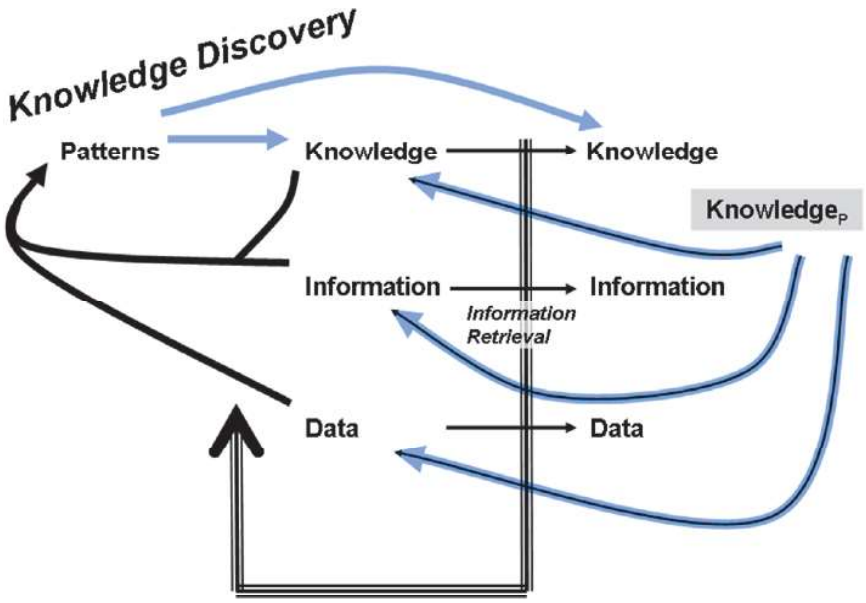


Fig. 3.8 Full web mining or "You're a document too" (Berendt 2008)

validity. Questions of privacy and context are further investigated in the following two sections.

3.4 New Challenges: Web Mining and Privacy

As Sect. 3.2.5 described in general form and Sect. 3.3 illustrated for specific examples, modern mining in general and web mining in particular operate not only on data/information/knowledge that have been put into some repository consciously and intentionally, often for the purpose of being findable/queryable. They also operate on documents that may not have been put online for the purpose of being analysed, on structural characteristics of these such as hyperlink structure between web documents, and on data arising from usage activities such as querying, browsing/navigating, or creating content. These are intellectual activities, of which measurable outcomes such as clicks, issued queries, uploads or edits can become input for KD. These intellectual activities and measurable outcomes are shown as grey and black arrows, respectively, in Fig. 3.8.

The main outcome of this is that the Web contains increasingly large amounts of "personal data", data that relate to an identified or identifiable individual. This concerns such things as behavior (which sites were visited, for how long, etc.), interests and questions (those sites' contents as well as queries posed, self-profiling on social networks, etc.), social relationships ("friendship" links in social networks, blogrolls, etc.), plus a wealth of self-reported or third-party-reported views, attitudes, etc. (all authored content). This is increasingly viewed as a threat to privacy, where privacy can refer to the interest in a protected sphere of personal and intimate life, the ability to keep different social identities separate and also to otherwise know and control

what happens to data about oneself, and to (re-)negotiate the boundaries between these private and public spheres in social and societal processes (Phillips 2004).

Data protection legislation limits the permitted uses of data in an attempt to protect individuals and help them to control the use of their data, and privacy-enhancing technologies aim at helping people keep data hidden from unwanted accesses or uses. However, it is increasingly becoming clear that current provisions are reaching their limits. For example, many planning activities—whether it be for research, marketing, urban planning, or other uses—require data, in particular data about people. A common solution to avoid infringing on individuals' private spheres has been to anonymize such data, and anonymous data are not protected by data protection laws. However, with even a small amount of background knowledge from other sources, it is possible to de-anonymize people, for example with the help of basic demographic data (Sweeney 2002), queries (Barbaro and Zeller 2006), social-network relationships (Backstrom et al. 2007), self-profiling information in e-Commerce websites (Owad 2006) or ratings and user-generated content such as forum posts (Frankowski et al. 2006; Narayanan and Shmatikov 2008). Many such sources of background knowledge are freely accessible.

The data mining and security/privacy communities have addressed this issue by extensive research on "privacy-preserving data mining", "privacy-preserving data publishing", "private information retrieval" and similar families of techniques. However, a number of findings shed doubt on the effectiveness of these techniques in real-world scenarios. A first concern is that some of these techniques may protect the privacy of different groups of people (data subjects, data owners, or data users), but not of others (Domingo-Ferrer 2007). A second concern is that the full theoretical and practical consequences of the de-anonymization results mentioned are only beginning to become clear. A third concern is that current accounts of privacy in computer science and law are disregarding the issues arising from relational information (that pertains not only to one individual, but to several, like the "friendship" links in social networks) and the many external effects that one person's dealings with their data have on others (Gürses and Berendt 2010; Berendt 2007).

A fourth concern is the focus of data protection laws and current privacy-enhancing technologies on the individual. This disregards the privacy-relevant effects of social categorizations (classifications). To understand this concern, one needs to consider the uses of classification in web mining for classifying people.

Classification is used for a wide range of personalisation services. The goal is to predict what the current user may find most interesting, attractive, etc., based on what the service knows about this user and what regularities have been observed in the past between these features and "liked items". These regularities may have been observed in this particular user's behavior or, more often, in larger populations of all users. Features may be the words (or other text-derived attributes) of previously read and therefore presumably liked texts; the prediction may be the degree of personal relevance of a new search result. Outcomes may range from recommendations via filtering, re-ranking, to a denial of service because the current user is classified as "belonging to an undesirable group" (such as a group of people with a high risk

of defaulting on a loan). The function learned may map directly from the user-describing feature space into the space of items; this however often suffers from the sparseness of typical datasets used as the training-set universe.

Often, therefore, there is a modeling step in the middle: First, the current user is mapped into a previously defined set of classes of users; second, the new item to be evaluated is scored with respect to this user class. These classes include such criteria as gender or age bracket (Liu and Mihalcea 2007; Hu et al. 2007), and socioeconomic or other categories deemed important by the provider of the personalisation service. For example, many marketing people in Germany work with the category of *DDR-Nostalgiker*, people who are "nostalgic for the former East German State", because their consumption patterns and attitudes differ from those of other target groups.[22] Marketing also has a tradition of classifying by ethnicity, skin colour, or sexual orientation (Wardlow 1996). Criticism of such classes is being voiced especially in the literature on social sorting from surveillance studies,[23]and it is also being discussed in the privacy literature at the interface between surveillance studies and privacy-enhancing technologies (Phillips 2004).

At the moment, it appears that in addition to adapting privacy legislation and legal practice to the changing realities of today's Web and other data spaces, in addition to educational and computational initiatives with various goals (confidentiality, data control, etc.), new methods for structured negotiations between data subjects and service providers on the one hand and between data subjects among themselves on the other will need to be developed (Preibusch 2006; Gürses 2010). These are likely to have major impacts on business models and related high-level application strategies, and therefore also on mining phases like business/application understanding.

3.5 Conclusions

In this article, we have touched upon several current application areas of web (content/text) mining. It has become clear that while much progress has been made, problems remain hard[24]—because the "target" keeps moving (spammers and their strategies for fooling search engines and users), because the target is, even in a non-adversarial setting, intrinsically changing (natural language evolves continually), because the target is not fully automatable (natural language understanding), or because pursued goals are contradictory (data analysis needs data with a high

[22] See http://www.sociovision.de/loesungen/sinus-milieus.html, retrieved on 2010-04-10.

[23] A cross-disciplinary initiative to understand the ways in which personal details are collected, stored, transmitted, checked, and used as means of influencing and managing people and populations; for an overview, see Lyon (2007).

[24] We have deliberately not discussed any accuracies, F measure values, or other absolute numbers here, in order to concentrate on the big picture. However, the reader is encouraged to consult original articles, investigate the reported quality values closely, and consider what for example a 20% misclassification rate or an unknown recall rate may mean in practice.

signal-to-noise ratio, privacy is about restricting data or their information content). This implies a wide range of research challenges on both computational and wider information-systems-related questions.

As an outlook, we want to point to some specific lessons learned and their implications for future research.

3.5.1 Context

The first challenge is the increasing realization that *context* is highly important for intelligent behavior (including inferences) of all kinds. This has been discussed (not only) in the privacy literature for some time. As pointed out above, one important part of many notions of privacy is the ability and right to keep different social identities apart. This means that a given piece of data/information/knowledge may not be taken out of its "context" and re-used somewhere else. For example, photographs may be meant for one's circle of friends or family, but not for one's employer. "Contextual integrity" (Nissenbaum 2004) ties adequate protection for privacy to norms of specific contexts, demanding that information gathering and dissemination be appropriate to that context and obey the governing norms of distribution within it. A formal model of contextual integrity has been proposed by Barth et al. (2006).

Context may affect not only the legitimacy, but also the validity of data-processing operations. Asserting that two social roles of one "physical individual" refer to "the same" individual and can therefore be combined ad lib may be wrong in a similar way as asserting that two URIs are `<sameAs>`, refer to the same entity, and that therefore all statements about them may be atomized and recombined ad lib. An example of the latter is provided by Hayes (2009): Two pieces of knowledge about sodium, one referring to the 3D model of this chemical, the other to the 4D model. Combining statements from these two universes of discourse is bound to lead to incorrect conclusions. Some logics provide for such contexts of conceptualizations (Cycorp 2001); applications in Linked Data and web mining are a promising application area and testing ground for such constructs.

3.5.2 Learning Cycles

The second lesson learned is the basic observation that all learning rests on iterations of a basic cycle such as that shown in Fig. 3.3b: All new knowledge is obtained on the basis of some previous knowledge and conceptualizations. This does not mean that such existing knowledge cannot also lead the learner astray. The argument between those who argue for the importance of background knowledge for reasoning and other intelligent activities, and those who regard it as cumbersome, misleading, etc. is at least as old as Artificial-Intelligence research as such. In mining, one strategy for avoiding background knowledge is to search for "domain-independent"

and/or "language-independent" techniques. However, such independence may be illusory. The critical reader should always closely examine all heuristics, extraction patterns, enumerations of classes, data preparation choices, etc. in order to find possibly hidden assumptions that do hinge on language, domain, or other properties of the mined materials.

3.5.3 Definitional Power and Viewpoints

The example of a multitude of contexts raises another question: Who may decide on contexts, and is this agent in possession of an absolute truth? The pragmatic answer to the second part is obviously "no", and results from semiotics and informatics suggest that it is not even possible to find such an agent or even a complete and expressive system of assertions. A structuralist approach in which people are enabled to dig deeper when they so desire appears to be superior. This makes *sourcing* a key element of knowledge processing: the indication of where a statement comes from—and thus what the inferences based on it rest on. The Cyc inference engine, for example, allows for such source-enhanced chains of reasoning (Baxter et al. 2005). In web/text mining, the search for viewpoints or biases in news reporting, user-generated content, etc. has recently received a lot of attention and new techniques for identifying common elements (to identify a common 'story') and differentiating elements (to identify diversity and bias), e.g. (Fortuna et al. 2009; Lin et al. 2008; Nakasaki et al. 2009). Sourcing is related to the modeling and processing of data *provenance*, which has been researched so far especially with a view to data quality and trustworthiness, cf. (Hartig 2009) for references and the relation to Linked Data.

3.5.4 Tools and Access

Finally, all these ideas will only be able to live up to their full potential if they are deployed in tools that are being used, tools for end users and not only skilled mining experts. More and more tools that marry data-analysis functionality with user-friendly interfaces are becoming available, see for example the development sections of major search engines, research-led initiatives like MovieLens,[25] or other tools like the Wikiscanner.[26] However, most tools are rather fixed in their functionality and/or data sources; thus, they support generativity (Zittrain 2008) at most at a content level, while generativity at a program level remains reserved for informatics-savvy elites. It is an open question whether this is the most we can get, whether we can and want to raise information-literacy levels to enable everyone to be a web miner, and what compromises this would entail.

[25] See http://movielens.umn.edu.

[26] See http://wikiscanner.virgil.gr.

Acknowledgements I thank my students and colleagues from various Web Mining classes for many valuable discussions and ideas. In particular, I thank the members of the European Summer School in Information Retrieval ESSIR 2007, the members of the Information Retrieval and Data Mining course at Universitat Pompeu Fabra in Barcelona 2010, and the members of the Advanced Databases and Current Trends in Databases courses at K.U. Leuven/U. Antwerp/U. Hasselt.