Web Mining

Petar Ristoski

1 Introduction

The World Wide Web (WWW) [1] emerged in the early 1990s, providing means to publish and access documents online. The WWW is based on the Hypertext Transfer Protocol (HTTP), used to transfer hypermedia documents between servers and clients, and the Hypertext Markup Language (HTML), which is the standard markup language for creating documents to be viewed in a Web browser. This allows users and organizations to easily and instantly publish information and documents on the Web. Since the beginning, a vast amount of information has been published on the Web, covering any imaginable topic and domain. Currently, the Web contains several billions of Web pages and several hundreds of billions of links between those pages, and it continuously expands. To be able to cope with such a large amount of data, especially being able to identify relevant information in it, *Web mining* approaches are being used. Web mining is the process of performing data mining on the Web, including Web documents, Web graph, and Web usage data. The main goal of Web mining is to identify and extract relevant information and knowledge hidden in Web data. Web mining approaches follow the standard knowledge discovery process, which consists of five steps: data collection, preprocessing, transformation, pattern discovery, and pattern analysis [2]. Web mining is a multidisciplinary research area involving approaches and techniques from many other research areas, e.g., databases, data mining, machine learning, information retrieval, information extraction, natural language processing, statistics, etc.

Web mining is commonly divided into three sub-areas:

P. Ristoski (✉)
IBM Research, San Jose, CA, USA
e-mail: petar.ristoski@ibm.com

L. Rokach et al. (eds.), *Machine Learning for Data Science Handbook*,
https://doi.org/10.1007/978-3-031-24628-9_20

1. Web Content Mining: The process of mining the content of the Web documents in order to identify useful information, using data mining and machine learning. This commonly includes mining unstructured or semi-structured text documents, images, audio and video files extracted from Web documents.
2. Web Structure Mining: The process of mining the graph structure of the Web to identify structural and connection information about the nodes in the graph. This is usually done using graph theory and graph mining techniques.
3. Web Usage Mining: The process of mining the Web server logs, in order to extract patterns and information about the user interactions with the Web servers of one or more Web sites.

There are several textbooks [3, 4, 5] that cover Web mining in details, as well as several surveys [6, 7, 8, 9]. For surveys on Web content mining, we refer to [10, 11, 12, 13], surveys for Web structure mining can be found in [14, 15, 16], and surveys for Web usage mining can be found in [17, 18, 19, 20, 21].

An additional sub-area of Web mining is Semantic Web mining, which is concerned with the application of Web mining approaches for the Semantic Web. The Semantic Web is an extension of the standard Web in which data is structured using well established formats and technologies, making it machine-readable and machine-understandable [22]. For surveys on Semantic Web mining, we refer to [23, 24, 25, 26, 27, 28, 29].

The remaining of the chapter is organized as follows. Section 2 gives an introduction to the graph structure and properties of the Web. Section 3 gives an overview of approaches used for text mining and information extraction on the Web. Section 4 surveys approaches for Web structure mining for information retrieval and social network analysis. Section 5 gives an overview of Web usage mining and query log mining approaches. Finally, Sect. 6 gives an introduction to Semantic Web and Semantic Web Mining.

2 The Graph Structure of the Web

The Web is a network, or a directed graph, where the documents are the nodes and the hyperlinks between the documents are the edges of the graph. The graph structure carries important information about the Web documents and the information contained in them, thus the Web structure becomes crucial for Web mining. As shown by Hall et al. [30], the Web and the underlying Web graph are evolving throughout time. The Web started as a collection of documents, also known as "the Web of documents," when in the 2000s the users started to be more involved and chaining the Web to "the Web of people," evolving to the present state called "the Web of data and social networks."

Throughout the years, many studies have tried to analyze the structure and the properties of the continuously evolving Web graph. One of the biggest challenges for analyzing the whole Web graph and Web mining, in general, is downloading the whole Web. To do so, there are special systems called *Web crawlers*, also known as *robots* or *spiders*, which are able to download Web pages in bulk. The architecture of a Web crawler is quite simple, i.e., given a set of seed *Uniform Resource Locators* (URLs), the crawler downloads and indexes the corresponding Web pages, then extracts the outgoing hyperlinks from those pages and iteratively downloads the new Web pages. The process continues until no new hyperlinks are discovered. While the algorithm looks trivial, there are many challenges to build an efficient crawler that can capture a significant portion of the Web and keep it up to date [31]. Many different approaches and strategies have been proposed for broad crawling of the Web [31, 32, 33], as well as focused crawls [34, 35], which focus on downloading Web pages that cover a certain topic.

Several broad crawls have been used to perform in-depth analysis of the whole Web graph. The first analysis on the whole Web was performed by Broder et al. [36]. They used two AltaVista crawls from 1999, each with more than 200 million URLs and 1.5 billion links. The main finding of this work was that the Web is structured as a giant bow tie, with a Strongly Connected Core component (SCC) of 56 million pages in the middle, and two side components with 44 million pages each. The side components are called IN and OUT, where IN consists of pages that can reach the SCC but cannot be reached from it; while OUT consists of pages that are accessible from the SCC but do not link back to it. Furthermore, an additional component called TENDRILS contains pages that cannot reach the SCC and cannot be reached from the SCC. The analysis on the whole graph showed that the diameter of the central core (SCC) is at least 28, and that the diameter of the graph as a whole is over 500. The bow tie structure describes a macroscopic structure of the Web. Donato et al. [37] study the inner part of the bow tie structure, revealing that each component actually has a significantly different structure and propose replacing the bow tie structure with a daisy structure, where the IN and OUT components are attached like petals to the core SCC. A similar study by Jonathan et al. [38] suggest that the structure of the Web is more like a teapot than a bow tie, indicating that the IN component is much bigger than the OUT component. However, later research show that most of these findings heavily depend on the crawl and the crawling process [39, 40, 41]. Given these new findings Meusel et al. [41, 42] and Lehmberg et al. [43] perform analysis on a 2012 crawl from the Common Crawl Foundation,[1] which contained 3.5 billion Web pages and 128 billion links. The analysis confirmed the existence of a giant connected core component; however, there are different proportions of nodes that can reach or that can be reached from the giant component, suggesting that the bow tie structure is dependent on the crawling process and is not a structural property of the Web. Furthermore, the analysis shows that the graph has a diameter of at least 5282. Such studies confirm that the Web continuously evolves, and it is not trivial to analyze the whole structure of the Web graph but remains crucial for successful Web mining.

[1] http://commoncrawl.org/.

3 Web Content Mining

Web content mining approaches extract or mine useful information and knowledge from Web documents. Although there are means to publish structured data (see Sect. 6), most of the data published on the Web remains in unstructured format, i.e., text. To be able to extract useful knowledge from the vast amount of text data published on the Web, text mining approaches are used [44, 45, 46]. This includes: information retrieval, document classification, information extraction, text generation, text summarization, opinion mining and sentiment analysis. Web content mining includes image, video, and multimedia processing as well; however, the main focus of Web content mining is text mining, therefore in this chapter we focus only on text mining.

Information Retrieval Information retrieval is the study concerned with representing, searching, and manipulating large collections of text [49]. As mentioned in the introduction, one of the biggest challenges for consuming the Web is identifying the small pieces of relevant information in the vast amount of data published on the Web. To do so, Web search engines are being used. Search engines allow the users to enter a set of keywords or a query, for which the engine will return a ranked list of Web pages relevant for the query. Some of the most popular search engines are Google, Bing, Yahoo Search, Yandex, and Baidu. Web search engines can be seen as traditional information retrieval system working on Web data [47, 48].

Search engines consist of 3 main components: crawl, index, and search. Crawling is the process of downloading Web documents in bulk, as explained in Sect. 2. Indexing is the process of efficiently and effectively storing the crawled Web pages in a structure that would allow efficient retrieval of the Web pages. Searching is the process of matching the user query with the indexed Web pages and retrieving the matched results ordered by relevance.

Once all the Web documents are collected, before the documents are passed to the indexer, the documents go through a document pre-processor. This usually includes: lexical analysis (including tokenization and token normalization), token weighing, stopword removal, stemming, phrase processing, and hyperlinks processing [50, 51]. After this step, each Web document is represented as a set of search terms (tokens or a set of tokens) with meta data for each token. There are several information retrieval models or document representation techniques used, starting from the more trivial Boolean and the vector space models [52], to more sophisticated probabilistic and language models [53, 54]. Web documents are indexed in an index structure, such as *inverted index* structure. For every search term the inverted index contains a list of Web documents that contain that search term. This allows the search engine to quickly identify all documents that contain a term from a user query. For a survey of more sophisticated indexing techniques we refer to the survey by Gani et al. [55]. During the indexing, usually terms are assigned relevance weight, which indicates the importance of the term in the current document. One of the most used term weighing techniques is *term frequency—inverse document frequency* (tf-idf).

Once the index is built, the search engine can process user queries. Usually the user queries consist of one or several search terms. When the user query is received, the search engine first pre-processes the query using the same pipeline used to pre-process the documents for indexing. Then, for each of the search terms the relevant documents with their weights are retrieved from the inverted index. Finally, for each retrieved document a rank score is calculated using a relevance ranking function. Relevance ranking functions usually consist of 2 parts: (i) content document relevance score based on the matched terms, which depends on the used retrieval model; (ii) document relevance score based on the graph structure (see Sect. 4). There are many content relevance ranking functions, e.g., cosine similarity, BF25 [56], relevance based on language models [53, 57], or using machine learning for information retrieval, called *learning to rank* [58]. Learning to rank is the task to automatically construct a ranking machine learning model using training data, such that the model can sort new objects according to their degrees of relevance, preference, or importance [59]. Commercial search engines use combinations of many relevance functions in order to get the best results. In many cases the user query might be very narrow and the search will not return any relevant results. To alleviate this problem, a *query expansion* approaches are used. Query expansion approaches modify the input query by adding additional search terms with similar semantic meaning [60, 61, 62].

While standard Web search engines deal only with keyword-based queries, with the advances of machine learning and knowledge representation, many systems now offer natural language *question answering*. In such systems, the users can ask a question in a natural language form, which the system then processes and matches to the internal knowledge, and produces an answer in natural language format. Many Web sites adapt such systems to develop *chatbots* and *conversational assistants* to ease the interaction with the users [63, 64]. With the rapid advancements of deep machine learning and knowledge graphs (see Sect. 6), such systems are becoming more popular [65, 66].

Document Classification Document classification is the task of sorting documents into a given set of categories. Categorizing Web documents into a predefined taxonomy made searching and browsing the Web easier [67]. The Open Directory Project (DMOZ) was the first Web topic directory that organized Web pages in a hierarchical ontology. Such categorization was used by search engines to improve the search results. While today such topic directories do not play big role in the search engine ranking, Web document classification is crucial for organizing and maintaining content on the Web. There are many supervised machine learning approaches for document classification, e.g., Naive Bayes, Support Vector Machines, Logistic Regression, Decision Trees, etc. With the advance of word embeddings (e.g., word2vec [68], GloVe [69]) and deep learning approaches (e.g., CNN [70], RNN [71], LSTM [72], BERT [73]) document classification achieves even better results.

When the set of categories is not known upfront, unsupervised methods can be used, such as *clustering* [74] and *topic modeling* [75].

Information Extraction Information extraction approaches automatically extract structured information from unstructured or semi-structured text. Some of the typical tasks in web information retrieval include *named entity recognition* (e.g., identifying mentions of people, places, organization, products, etc.) and *relationship extraction* between such named entities [76]. Named entity recognition is one of the most important tasks of natural language processing, and therefore there is a plethora of work in the literature [77, 78, 79]. Approaches range from simple techniques using external vocabularies and knowledge bases [80], unsupervised methods [81, 82], sophisticated supervised solutions with advanced feature engineering based on Hidden Markov Models [83], Conditional Random Fields [84] and Support Vector Machines [85], and lately advanced deep neural networks [86, 87, 88, 89, 79].

Once the named entities are identified in text, usually the next task is to identify the relation between them. Popular systems for relation extraction range from early solutions based on SVMs and tree kernels [90, 91, 92, 93, 94] to most recent ones exploiting neural architectures [95, 96, 97]. Many approaches exploit large knowledge bases for distant supervision [98, 99, 100, 101, 102], as well as human-in-the-loop approaches [103]. Furthermore, several works have tackled the problem of open information extraction [104, 105, 106] where the focus is on general understanding of text rather than a focused extraction of named entities and specific relations. Furthermore, recent deep neural network approaches solve both tasks of named entity recognition and relation extraction as a single task [107].

In most Web sites, the Web pages are dynamically generated using similar templates, using data from the same database. *Wrapper induction* approaches use supervised machine learning approaches to learn patterns and rules for generating dynamic Web pages in order to automatically extract structured information from them [108, 109].

Besides mining unstructured text from the Web, a lot of work goes into mining semi-structured data, such as lists and tables. WebTables was the first project to extract content tables from the whole Web [110, 111]. The project shows how to extract useful information from Web tables, such as entities, attributes, and relations between entities, which can be used in different applications. Several approaches have been introduced for parsing and semantically annotating Web tables [112, 113, 114], which then can be used for improved search [115], query table extension [116, 117, 118] and knowledge base completion [119, 120]. One survey of recent approaches for Web table mining can be found in [121].

Text Generation and Summarization While text generation was introduced in the early 1990s [122], it became more popular in the recent years with the advance of neural networks [123]. Currently, *generative adversarial neural networks* with reinforcement learning [122, 123, 124, 125] and transformer-based neural networks [126] show state-of-the-art results. Such systems have been successfully used in many applications: generating product descriptions [127], generating product reviews [128, 129], weather forecast [130], and many others. Such systems change the way new content is being created and published to the Web.

On the other hand, *text summarization* approaches try to shorten existing text documents. Such approaches generate a concise summary of large texts, focusing on the pieces of text that provide the most useful information without losing the overall meaning [131]. While in the past most of the approaches were based on topic words, frequencies, and latent semantic analysis, in the recent years *seq2seq* neural networks [132] have become state-of-the-art solution [133]. Such systems allow users to faster identify and consume relevant information on the Web.

Opinion Mining and Sentiment Analysis Web users use the available content from social media platforms, forums, and blogs for their decision making, e.g., the decision to buy a product offered on an e-shop heavily depends on the product reviews written by other users. Thus, it is crucial to be able to automatically identify the opinion and sentiments in user generated text on the Web. Since 2000, sentiment analysis has grown to be one of the most important research areas in natural language processing. Thus, plethora of approaches for opinion mining and sentiment analysis have been proposed [134, 135, 136].

Similar to opinion mining and sentiment analysis approaches, social media mining approaches [137] identify patterns and trends from textual data published on social media, which later can be used in different applications. The most common use of social media mining is for advertising, i.e., identifying the user's interests based on the content they generate to identify the best advertisements for them. Besides advertisement, social media mining has been used for identifying incidents and crisis in real time [138, 139, 140, 141], sports analysis, political analysis, trends, and more [142]. An interesting problem in social media mining is identifying fake news [143], which with the advance of generative neural networks becomes a real problem.

4 Web Structure Mining

As described in Sect. 2, the Web is a directed graph, which structure intrinsically carries important information about Web pages and the type and the quality of data in them. Using graph theory and graph mining approaches such information can be extracted and used in different applications, such as information retrieval and different social network analysis on the whole Web graph or separate Web sites.

Information Retrieval Early Web information retrieval systems used only the content Web data to retrieve the most relevant Web pages for a given user search query, as shown in Sect. 3. However, with the fast growth of the Web, the size of returned search results for any search query became rather large. Analyzing a large number of search results is not convenient for the users and it is costly. To cope with the large amount of Web pages, Web information retrieval systems started using the Web graph structure to identify the *popularity* of the Web pages. Such a popularity score is then used in the relevance ranking function in information retrieval systems. The popularity score of a Web page is calculated directly from the Web graph structure, i.e., by examining how many hyperlinks point to the Web page and the popularity of the Web pages pointing to it. One naive solution is to use the number of incoming links to the Web page, also known as in-degree of the Web page, as a popularity score. However, this score can easily be manipulated by setting hyperlinks to Web pages that are not relevant, known as *spamming*, which would significantly decrease the performance of the search engines. In 1998 and 1999, the two most important Web page ranking algorithms were introduced, *PageRank* [144, 145] and *HITS* [146]. Both of these algorithms originate from social network analysis [147, 148].

The HITS (hyperlink-induced topic search) algorithm [146], also known as *hubs and authorities*, was introduced by Kleinberg in 1999. For a given search query, the algorithm identifies two types of Web pages: (i) *authorities* are pages that contain relevant information for the query; (ii) *hubs* are pages that contain hyperlinks to good sources. The main idea of HITS is that there is a mutual reinforcement relationship between authorities and hubs, i.e., good hubs link to many good authorities, and a good authority is linked to by many good hubs. Given a search query, the algorithm first extends the set of search result pages by adding all the pages that point to any of the pages in the result list, or pages that are pointed to by any of the pages in the result list. In the next step, each page in the expanded list is assigned authority score and a hub score. The authority score of a Web page is calculated as the sum of the hub scores of all Web pages pointing to it. The hub score of a Web page is calculated as the sum of the authority scores of all Web pages that the current page is pointing to. The scores are computed iteratively, and after each iteration the values are normalized between 0 and 1. Kleinberg proved that the algorithm will always converge within couple of iteration, which has been shown many times in practical experiments. While the algorithm has shown great performance to identify relevant pages for a given query, the algorithm has two main drawbacks that make it unusable in modern search engines: (i) it is sensitive to spamming, i.e., the hub score can be manipulated by adding a lot of outgoing links to good authorities; (ii) the scores are query specific and must be calculated during the search, which significantly increases the search time.

On the other hand, PageRank [144, 145] is a static Web page ranking algorithm, i.e., the PageRank value for each Web page in the Web graph is calculated offline and it is not query dependent. The PageRank algorithm was introduced by Brian and Page in 1988 and it is used in the popular *Google Search*. The underlying assumption of PageRank is similar to the one of HITS, i.e., more important Web pages are more likely to be linked to buy more important Web pages. The PageRank of a Web page is calculated as the sum of the PageRank values of all the Web pages pointing to it, where the PageRank value of each Web page is divided by the total number of links going out of that page. Furthermore the algorithm introduces a damping factor, which simulates the probability of a random surfer continuing to click on links as they browse the Web. Then the PageRank value of a Web page A is calculated as:

$$PR(A) = \frac{1-d}{N} + d \sum_{(A,B)} \frac{PR(B)}{O_B} \tag{1}$$

where d is the damping factor, which is set between 0 and 1, N is the total number of pages in the Web graph, and O_B is the number of outgoing links for any Web page B that links to A. The PageRank value is calculated iteratively.

The main strengths of the PageRank algorithm are being query independent and unaffected by spam. Although Google Search currently uses many other signals and features for ranking Web pages, it is noteworthy that even after many years of existence, PageRank is still being used in Google Search. There are several extensions of the PageRank algorithm. Xing and Ghorbani proposed the weighted PageRank algorithm [149], which takes into account the importance of both the incoming and the outgoing links of the pages and distributes rank scores based on the popularity of the pages. Li and Liu introduce the TS-Rank algorithm [150] to address the issue of assigning low PageRank scores to new Web pages, which might contain high quality content.

Social Network Analysis Social network analysis is the study of social entities, such as people, organizations, groups, Web pages, or any knowledge entities, and the relationships between them [147, 148]. Such analysis can infer interesting properties, roles, and relationships between nodes or group of nodes in the network. For example, HITS and PageRank are being used for identifying the popularity of different Web pages in the whole Web graph. An important social network study is *community detection* in networks [151]. A community represents a group of entities that cover a same topic, share a same interest, or are involved in an event. Identifying such communities can give interesting insights on the structure of the network and the function of different communities, which could be used in various applications, e.g., topic-based content clustering, recommender systems, advertising, etc. Several approaches have been proposed for community identification on the whole Web [152, 153], to identify different clusters of Web pages. However, most of the recent approaches focus on community detection in social media platforms on the Web, which connect millions of users and groups [154, 155].

With the advance of deep learning and representation learning in the recent years, network representation learning approaches are introduced [156]. Network representation learning approaches learn latent, low-dimensional representation of network nodes, while preserving the network structure and node content and attributes. Such representation of nodes in a network can be used in many social network analysis on the Web, such as classification [157, 158], link prediction [159], clustering [160], recommender systems [161], visualization [162], and search [163]. Several recent surveys give an overview of such approaches and their applications on different networks on the Web [156, 164, 165].

5 Web Usage Mining

Web usage mining is the process of identifying and analyzing patterns in Web server logs, including clickthrough, clickstream, user transactions, and other data about user interactions with the Web server, which are then used to improve the performance of different services on the Web [17, 166]. The main data source for Web usage mining are server logs, which include web server logs and application server logs. From such data, information about each visit and interaction with the Web server is recorded. The first step of web usage mining is processing the raw log files and extract structured information [167], which typically includes: (i) user identification—identifying all actions that belong to the same user; (ii) Pageview identification—identifying which Web pages are being visited including the attributes of the Web pages; (iii) Sessionization—identifying a set of pages visited by the same user over one visit to a Web site; (iv) Episode identification—identifying a set of Web pages visited by the user that are semantically or functionally related. In the next step, these data are transformed into a data model, which can be used in different data mining algorithms to identify useful patterns from the log files. There are several common mining approaches used for improving the content, structure, and design of Web sites, including session and user analysis, which gives basic visit and popularity statistics of different Web pages within a Web site; classification and clustering of users to identify user communities (see Sect. 4); identifying associated Web pages, or products and services that are commonly bought together, using association rule mining. More advanced applications of Web usage mining involve recommender systems on the Web, and applications of query log mining.

Recommender Systems Recommender systems are machine learning models that predict a rating score or a binary preference a user would give to an item [168]. Recommender systems are the core component of every e-commerce Web site, used for recommending products and services to users based on their preferences, interests and previous interactions with the Web site. Since the performance of the recommender systems directly affects the success and the profit of e-shops, plethora of approaches have been proposed in the literature [169, 170]. There are three general types of recommender systems: (i) *Collaborative filtering* methods recommend items that are liked/bought by users with similar interests and similar past behavior as the current user; (ii) *Content-based filtering* methods use content attributes of the liked/bought items to recommend similar items to the users; (iii) *Hybrid* recommender systems combine collaborative and content-based filtering methods to achieve better performance and circumvent the drawbacks of the two when used separately.

Query Log Mining Query Log Mining (QLM) is a special type of Web usage mining [17], focused on mining log files from Web search engines. QLM techniques are used in variety of applications [171, 172], predominantly being used to improve the ranking and the runtime efficiency of search engines. For example, QLM approaches can be used to discover patterns and knowledge that can be used for future query refinement and expansion [173], personalized query recommendation and suggestion [174, 175], resolving ambiguity and intent [176], and removing spam. Another important application of QLM is in *search advertising*, also known as sponsored search, where QLM methods are being used to deliver the best matching advertisements to the users based on their search queries and preferences, thus maximizing the revenue of the search engines [177, 178]. *Contextual advertising* is another type of advertising, which tries to match ads with the context displayed in the current Web page [177], for which methods from Web content mining and Web usage mining are used. Furthermore, search queries can be seen as signals in a domain over time and can be represented as time-series. Thus, time-series analysis approaches can be applied on query logs to identify new trends, events, interests, and preferences [179], e.g., political events [180] or virus epidemics [181].

6 The Semantic Web and Semantic Web Mining

While the current Web is intended to be human readable, the Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries. Semantic Web technologies facilitate building a large-scale Web of machine-readable and machine-understandable knowledge and thus facilitate data reuse and integration. The basics of the Semantic Web were set by Tim Berners-Lee in 2001 [22], which later lead to the creation of the Linked Open Data [182].[2] Linked Open Data (LOD) is an open, interlinked collection of datasets on the Web in machine-interpretable form, covering many domains [183]. Currently, more than 1000 datasets are interlinked in the Linked Open Data cloud.[3] Since the beginning, the Semantic Web has promoted a graph-based representation of knowledge, e.g., using the Resource Description Framework (RDF).[4] In general, RDF is a framework which provides capabilities to describe information about resources. The core structure of RDF is a set of triples, each consisting of a subject, a predicate, and an object, e.g., db:Berlin dbo:capitalOf db:Germany represents a triple. A set of such triples is called an RDF graph. The term used to describe such RDF graphs has been evolving through the years, i.e., in the beginning they were called *Ontologies* [184], while currently are known as *Semantic Web Knowledge Graphs* or simply *Knowledge Graphs* [185].

In the last decade, a vast amount of approaches have been proposed which combine methods from data mining and knowledge discovery with Semantic Web knowledge graphs. The goal of those approaches is to support different data mining tasks, or to improve the Semantic Web itself. All those approaches can be divided

[2] https://www.w3.org/DesignIssues/LinkedData.html.

[3] https://lod-cloud.net/.

[4] https://www.w3.org/RDF/.

into three broader categories [28, 29]: (i) Using machine learning techniques to create and improve Semantic Web data; (ii) Using data mining techniques to mine the Semantic Web, also called Semantic Web Mining; (iii) Using Semantic Web based approaches, Semantic Web Technologies, and Linked Open Data to support the process of knowledge discovery and data mining.

In the very beginning, a lot of effort was put into building the Semantic Web and Semantic Web datasets, which required use of data mining and machine learning, e.g., extracting structured data from text or Web pages, which is also known as ontology learning [23, 24]. Semantic Web mining became very popular, allowing for formal querying and reasoning over ontological knowledge bases [25, 26, 27]. Because of its structured nature, Semantic Web data has been heavily used as a background knowledge in various machine learning tasks and applications [28, 29]. With the rapid development of deep learning the most popular consumption of Semantic Web Knowledge graphs in data mining is by using *Knowledge Graph Embeddings* [186, 187]. Such algorithms embed the entities and relations, in some cases even bigger components, into a continuous vectors space, where each component is represented with an n-dimensional vector while preserving the information and structure from the graph. Such representation allows easy use of the knowledge graph in various tasks and applications [188, 189].

Besides Linked Open Data and Semantic Web Knowledge Graphs, semantic annotations in HTML pages are another realization of the Semantic Web. Semantic annotations are integrated into the code of HTML pages using one of the four markup languages Microformats,[5] RDFa,[6] Microdata[7] and JSON-LD.[8] Such markup languages extend the standard HTML markup with additional set of attributes and can be automatically recognized, e.g., by a machine. Such semantic annotations are used by search engine companies, such as Bing, Google, Yahoo!, and Yandex. They use semantic annotations from crawled Web pages to enrich the presentation of search results and to complement their knowledge bases [190]. There are several initiatives to extract such data from the whole Web and make it publicly available, such as the *Web Data Commons*.[9] Such data has been used in many e-commerce applications, such as product matching and product categorization [191, 192, 193].

[5] http://microformats.org/.

[6] https://www.w3.org/TR/rdfa-core/.

[7] https://www.w3.org/TR/microdata/.

[8] https://www.w3.org/TR/json-ld11/.

[9] http://webdatacommons.org/.

References

1. T. Berners-Lee, R. Cailliau, A. Luotonen, H. F. Nielsen, A. Secret, The world-wide web, Communications of the ACM 37 (8) (1994) 76–82.
2. U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, From data mining to knowledge discovery in databases, AI magazine 17 (3) (1996) 37–37.
3. S. Chakrabarti, Mining the Web: Discovering knowledge from hypertext data, Elsevier, 2002.
4. H. Chen, M. Chau, Web mining: Machine learning for web applications, Annual review of information science and technology 38 (1) (2004) 289–329.
5. B. Liu, Web data mining: exploring hyperlinks, contents, and usage data, Springer Science & Business Media, 2011.
6. R. Kosala, H. Blockeel, Web mining research: A survey, ACM SIGKDD Explorations Newsletter 2 (1) (2000) 1–15.
7. Q. Zhang, R. S. Segall, Web mining: a survey of current research, techniques, and software, International Journal of Information Technology & Decision Making 7 (04) (2008) 683–720.
8. B. Singh, H. K. Singh, Web data mining research: a survey, in: 2010 IEEE International Conference on Computational Intelligence and Computing Research, IEEE, 2010, pp. 1–10.
9. K. Sharma, G. Shrivastava, V. Kumar, Web mining: Today and tomorrow, in: 2011 3rd International Conference on Electronics Computer Technology, Vol. 1, IEEE, 2011, pp. 399–403.
10. F. Johnson, S. K. Gupta, Web content mining techniques: a survey, International Journal of Computer Applications 47 (11).
11. C. E. Dinucă, D. Ciobanu, Web content mining, Annals of the University of Petrosani. Economics 12 (2012) 85–92.
12. A. Herrouz, C. Khentout, M. Djoudi, Overview of web content mining tools, arXiv preprint arXiv:1307.1024.
13. M. O. Samuel, A. I. Tolulope, O. O. Oyejoke, A systematic review of current trends in web content mining, in: Journal of Physics: Conference Series, Vol. 1299, IOP Publishing, 2019, p. 012040.
14. J. Fürnkranz, Web structure mining, Exploiting the Graph Structure of the World-Wide Web, Österreichische Gesellschaft für Artificial Intelligence (ÖGAI) (2002) 17–26.
15. P. R. Kumar, A. K. Singh, Web structure mining: exploring hyperlinks and algorithms for information retrieval, American Journal of applied sciences 7 (6) (2010) 840.
16. R. Jain, D. G. Purohit, Page ranking algorithms for web mining, International journal of computer applications 13 (5) (2011) 22–25.
17. J. Srivastava, R. Cooley, M. Deshpande, P.-N. Tan, Web usage mining: Discovery and applications of usage patterns from web data, ACM SIGKDD Explorations Newsletter 1 (2) (2000) 12–23.
18. J. Vellingiri, S. C. Pandian, A survey on web usage mining, Global Journal of Computer Science and Technology.
19. T. Hussain, S. Asghar, N. Masood, Web usage mining: A survey on preprocessing of web log file, in: 2010 International Conference on Information and Emerging Technologies, IEEE, 2010, pp. 1–6.
20. L. Grace, V. Maheswari, D. Nagamalai, Analysis of web logs and web user in web mining, arXiv preprint arXiv:1101.5668.
21. V. Chitraa, D. Davamani, A. Selvdoss, A survey on preprocessing methods for web usage data, arXiv preprint arXiv:1004.1257.
22. T. Berners-Lee, J. Hendler, O. Lassila, et al., The semantic web, Scientific American 284 (5) (2001) 28–37.
23. V. Tresp, M. Bundschus, A. Rettinger, Y. Huang, Towards machine learning on the semantic web, in: Uncertainty reasoning for the Semantic Web I, Springer,
24. A. Rettinger, U. Lösch, V. Tresp, C. d'Amato, N. Fanizzi, Mining the semantic web, Data Mining and Knowledge Discovery 24 (3) (2012) 613–662 2006, pp. 282–314.

25. Q. K. Quboa, M. Saraee, A state-of-the-art survey on semantic web mining, Intelligent Information Management 5 (01) (2013) 10.
26. D. Dou, H. Wang, H. Liu, Semantic data mining: A survey of ontology-based approaches, in: Proceedings of the 2015 IEEE 9th international conference on semantic computing (IEEE ICSC 2015), IEEE, 2015, pp. 244–251.
27. K. Sridevi, D. R. UmaRani, A survey of semantic based solutions to web mining, International Journal of Emerging Trends and Technology in Computer Science (IJETTS) 1.
28. P. Ristoski, H. Paulheim, Semantic web in data mining and knowledge discovery: A comprehensive survey, Web semantics: science, services and agents on the World Wide Web 36 (2016) 1–22.
29. P. Ristoski, Exploiting semantic web knowledge graphs in data mining, Vol. 38, IOS Press, 2019.
30. Wendy Hall and Thanassis Tiropanis. Web evolution and web science. *Computer Networks*, 56(18):3859–3865, 2012.
31. Christopher Olston, Marc Najork, et al. Web crawling. *Foundations and Trends® in Information Retrieval*, 4(3):175–246, 2010.
32. SM Pavalam, SV Kashmir Raja, Felix K Akorli, and M Jawahar. A survey of web crawler algorithms. *International Journal of Computer Science Issues (IJCSI)*, 8(6):309, 2011.
33. Manish Kumar, Rajesh Bhatia, and Dhavleesh Rattan. A survey of web crawlers for information retrieval. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 7(6):e1218, 2017.
34. Blaž Novak. A survey of focused web crawling algorithms. *Proceedings of SIKDD*, 5558:55–58, 2004.
35. Yong-Bin Yu, Shi-Lei Huang, Nyima Tashi, Huan Zhang, Fei Lei, and Lin-Yang Wu. A survey about algorithms utilized by focused web crawler. *Journal of Electronic Science and Technology*, 16(2):129–138, 2018.
36. Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the web. *Computer networks*, 33(1-6):309–320, 2000.
37. Debora Donato, Stefano Leonardi, Stefano Millozzi, and Panayiotis Tsaparas. Mining the inner structure of the web graph. In *WebDB*, pages 145–150. Citeseer, 2005.
38. Jonathan JH Zhu, Tao Meng, Zhengmao Xie, Geng Li, and Xiaoming Li. A teapot graph and its hierarchical structure of the Chinese web. In *Proceedings of the 17th international conference on World Wide Web*, pages 1133–1134, 2008.
39. M Ángeles Serrano, Ana Maguitman, Marián Boguñá, Santo Fortunato, and Alessandro Vespignani. Decoding the structure of the www: A comparative analysis of web crawls. *ACM Transactions on the Web (TWEB)*, 1(2):10–es, 2007.
40. Dimitris Achlioptas, Aaron Clauset, David Kempe, and Cristopher Moore. On the bias of traceroute sampling: or, power-law degree distributions in regular graphs. *Journal of the ACM (JACM)*, 56(4):1–28, 2009.
41. Robert Meusel, Sebastiano Vigna, Oliver Lehmberg, and Christian Bizer. Graph structure in the web—revisited: a trick of the heavy tail. In *Proceedings of the 23rd international conference on World Wide Web*, pages 427–432, 2014.
42. Robert Meusel, Sebastiano Vigna, Oliver Lehmberg, and Christian Bizer. The graph structure in the web–analyzed on different aggregation levels. *The Journal of Web Science*, 1, 2015.
43. Oliver Lehmberg, Robert Meusel, and Christian Bizer. Graph structure in the web: aggregated by pay-level domain. In *Proceedings of the 2014 ACM conference on Web science*, pages 119–128, 2014.
44. R. Feldman, I. Dagan, Knowledge discovery in textual databases (KDT)., in: KDD, Vol. 95, 1995, pp. 112–117.
45. C. C. Aggarwal, C. Zhai, Mining text data, Springer Science & Business Media, 2012.
46. M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, K. Kochut, A brief survey of text mining: Classification, clustering and extraction techniques, arXiv preprint arXiv:1707.02919.

47. S. Büttcher, C. L. Clarke, G. V. Cormack, Information retrieval: Implementing and evaluating search engines, MIT Press, 2016.
48. W. B. Croft, D. Metzler, T. Strohman, Search engines: Information retrieval in practice, Vol. 520, Addison-Wesley Reading, 2010.
49. C. D. Manning, P. Raghavan, H. Schütze, Introduction to information retrieval, Cambridge university press, 2008.
50. G. Miner, J. Elder IV, A. Fast, T. Hill, R. Nisbet, D. Delen, Practical text mining and statistical analysis for non-structured text data applications, Academic Press, 2012.
51. A. K. Uysal, S. Gunal, The impact of preprocessing on text classification, Information Processing & Management 50 (1) (2014) 104–112.
52. R. Baeza-Yates, B. Ribeiro-Neto, et al., Modern information retrieval, Vol. 463, ACM press New York, 1999.
53. J. M. Ponte, W. B. Croft, A language modeling approach to information retrieval, in: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, 1998, pp. 275–281.
54. G. Amati, Information Retrieval Models, Springer New York, New York, NY, 2018, pp. 1976–1981.
55. A. Gani, A. Siddiqa, S. Shamshirband, F. Hanum, A survey on indexing techniques for big data: taxonomy and performance evaluation, Knowledge and information systems 46 (2) (2016) 241–284.
56. S. E. Robertson, Overview of the okapi projects, Journal of documentation.
57. C. Zhai, Statistical language models for information retrieval, Synthesis Lectures on Human Language Technologies 1 (1) (2008) 1–141.
58. T.-Y. Liu, et al., Learning to rank for information retrieval, Foundations and Trends® in Information Retrieval 3 (3) (2009) 225–331.
59. T.-Y. Liu, Learning to Rank for Information Retrieval., Springer, 2011.
60. C. Carpineto, G. Romano, A survey of automatic query expansion in information retrieval, ACM Computing Surveys (CSUR) 44 (1) (2012) 1–50.
61. J. Ooi, X. Ma, H. Qin, S. C. Liew, A survey of query expansion, query suggestion and query refinement techniques, in: 2015 4th International Conference on Software Engineering and Computer Systems (ICSECS), IEEE, 2015, pp. 112–117.
62. H. K. Azad, A. Deepak, Query expansion techniques for information retrieval: A survey, Information Processing & Management 56 (5) (2019) 1698–1735.
63. R. Dale, The return of the chatbots, Natural Language Engineering 22 (5) (2016) 811–817.
64. A. Følstad, P. B. Brandtzæg, Chatbots and the new world of HCI, interactions 24 (4) (2017) 38–42.
65. D. Diefenbach, V. Lopez, K. Singh, P. Maret, Core techniques of question answering systems over knowledge bases: a survey, Knowledge and Information systems 55 (3) (2018) 529–569.
66. S. Vakulenko, Knowledge-based conversational search, arXiv preprint arXiv:1912.06859.
67. I. Russell, Z. Markov, T. Neller, Web document classification, Jun 3 (2005) 1–19.
68. T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in neural information processing systems, 2013, pp. 3111–3119.
69. J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.
70. Y. Kim, Convolutional neural networks for sentence classification, arXiv preprint arXiv:1408.5882.
71. Y. Bengio, P. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult, IEEE transactions on neural networks 5 (2) (1994) 157–166.
72. S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber, et al., Gradient flow in recurrent nets: the difficulty of learning long-term dependencies (2001).
73. J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805.

74. P. Berkhin, A survey of clustering data mining techniques, in: Grouping multidimensional data, Springer, 2006, pp. 25–71.
75. M. Steyvers, T. Griffiths, Probabilistic topic models, Handbook of latent semantic analysis 427 (7) (2007) 424–440.
76. L. Chiticariu, M. Danilevsky, H. Ho, R. Krishnamurthy, Y. Li, S. Raghavan, F. Reiss, S. Vaithyanathan, H. Zhu, Web information extraction. (2018).
77. D. Nadeau, S. Sekine, A survey of named entity recognition and classification, Lingvisticae Investigationes 30 (1) (2007) 3–26.
78. G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, Neural architectures for named entity recognition, arXiv preprint arXiv:1603.01360.
79. V. Yadav, S. Bethard, A survey on recent advances in named entity recognition from deep learning models, arXiv preprint arXiv:1910.11470.
80. I. Segura Bedmar, P. Martínez, M. Herrero Zazo, Semeval-2013 task 9: Extraction of drug-drug interactions from biomedical texts (DDIExtraction 2013), Association for Computational Linguistics, 2013.
81. M. Collins, Y. Singer, Unsupervised models for named entity classification, in: 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, 1999.
82. S. Zhang, N. Elhadad, Unsupervised biomedical named entity recognition: Experiments with clinical and biological texts, Journal of biomedical informatics 46 (6) (2013) 1088–1098.
83. G. Zhou, J. Su, Named entity recognition using an hmm-based chunk tagger, in: proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, 2002, pp. 473–480.
84. S. Liu, B. Tang, Q. Chen, X. Wang, Effects of semantic features on machine learning-based drug name recognition systems: word embeddings vs. manually constructed dictionaries, Information 6 (4) (2015) 848–865.
85. Y. Li, K. Bontcheva, H. Cunningham, SVM based learning system for information extraction, in: International Workshop on Deterministic and Statistical Methods in Machine Learning, Springer, 2004, pp. 319–339.
86. R. Collobert, J. Weston, A unified architecture for natural language processing: Deep neural networks with multitask learning, in: Proceedings of the 25th international conference on Machine learning, 2008, pp. 160–167.
87. R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, Natural language processing (almost) from scratch, Journal of machine learning research 12 (Aug) (2011) 2493–2537.
88. Z. Huang, W. Xu, K. Yu, Bidirectional LSTM-CRF models for sequence tagging, arXiv preprint arXiv:1508.01991.
89. Y. Kim, Y. Jernite, D. Sontag, A. M. Rush, Character-aware neural language models, in: Thirtieth AAAI Conference on Artificial Intelligence, 2016.
90. R. C. Bunescu, R. J. Mooney, A shortest path dependency kernel for relation extraction, in: HLT/EMNLP, ACL, 2005, pp. 724–731.
91. A. Culotta, J. Sorensen, Dependency tree kernels for relation extraction, in: ACL, ACL, 2004, p. 423.
92. R. J. Mooney, R. C. Bunescu, Subsequence kernels for relation extraction, in: NIPS, 2006, pp. 171–178.
93. D. Zelenko, C. Aone, A. Richardella, Kernel methods for relation extraction, Journal of machine learning research 3 (2003) 1083–1106.
94. S. Zhao, R. Grishman, Extracting relations with integrated information using kernel methods, in: ACL, ACL, 2005, pp. 419–426.
95. T. H. Nguyen, R. Grishman, Relation extraction: Perspective from convolutional neural networks., in: VS@ HLT-NAACL, 2015, pp. 39–48.
96. D. Zeng, K. Liu, S. Lai, G. Zhou, J. Zhao, et al., Relation classification via convolutional deep neural network, in: COLING, 2014, pp. 2335–2344.

97. N. T. Vu, H. Adel, P. Gupta, et al., Combining recurrent and convolutional neural networks for relation classification, in: NAACL-HLT, 2016, pp. 534–539.
98. I. Augenstein, D. Maynard, F. Ciravegna, Distantly supervised web relation extraction for knowledge base population, Semantic Web 7 (4) (2016) 335–349.
99. A. L. Gentile, Z. Zhang, I. Augenstein, F. Ciravegna, Unsupervised wrapper induction using linked data, in: K-CAP, ACM, 2013, pp. 41–48.
100. G. Ji, K. Liu, S. He, J. Zhao, Distant supervision for relation extraction with sentence-level attention and entity descriptions, in: AAAI, 2017, pp. 3060–3066.
101. A. J. Ratner, C. D. Sa, S. Wu, D. Selsam, C. Ré, Data programming: Creating large training sets, quickly, in: NIPS, 2016, pp. 3567–3575.
102. B. Roth, T. Barth, M. Wiegand, D. Klakow, A survey of noise reduction methods for distant supervision, in: AKBC, ACM, 2013, pp. 73–78.
103. P. Ristoski, A. L. Gentile, A. Alba, D. Gruhl, S. Welch, Large-scale relation extraction from web documents and knowledge graphs with human-in-the-loop, Journal of Web Semantics (2019) 100546.
104. M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, O. Etzioni, Open information extraction from the web, in: Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI'07, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2007, pp. 2670–2676.
105. O. Etzioni, A. Fader, J. Christensen, S. Soderland, M. Mausam, Open information extraction: The second generation, in: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence—Volume One, IJCAI'11, AAAI Press, 2011, pp. 3–10.
106. V. Presutti, A. G. Nuzzolese, S. Consoli, A. Gangemi, D. Reforgiato Recupero, From hyperlinks to semantic web properties using open knowledge extraction, Semantic Web 7 (4) (2016) 351–378.
107. Q. Li, H. Ji, Incremental joint extraction of entity mentions and relations, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2014, pp. 402–412.
108. N. Kushmerick, D. S. Weld, R. Doorenbos, Wrapper induction for information extraction, University of Washington Washington, 1997.
109. N. Dalvi, R. Kumar, M. Soliman, Automatic wrappers for large scale web extraction, Proceedings of the VLDB Endowment 4 (4) (2011) 219–230.
110. M. J. Cafarella, A. Halevy, D. Z. Wang, E. Wu, Y. Zhang, Webtables: exploring the power of tables on the web, Proceedings of the VLDB Endowment 1 (1) (2008) 538–549.
111. M. Cafarella, A. Halevy, H. Lee, J. Madhavan, C. Yu, D. Z. Wang, E. Wu, Ten years of WebTables, Proceedings of the VLDB Endowment 11 (12) (2018) 2140–2149.
112. G. Limaye, S. Sarawagi, S. Chakrabarti, Annotating and searching web tables using entities, types and relationships, Proceedings of the VLDB Endowment 3 (1-2) (2010) 1338–1347.
113. P. Venetis, A. Y. Halevy, J. Madhavan, M. Pasca, W. Shen, F. Wu, G. Miao, Recovering semantics of tables on the web.
114. Z. Zhang, Effective and efficient semantic table interpretation using TableMiner+, Semantic Web 8 (6) (2017) 921–957.
115. M. J. Cafarella, A. Halevy, N. Khoussainova, Data integration for the relational web, Proceedings of the VLDB Endowment 2 (1) (2009) 1090–1101.
116. X. Zhang, Y. Chen, J. Chen, X. Du, L. Zou, Mapping entity-attribute web tables to web-scale knowledge bases, in: International Conference on Database Systems for Advanced Applications, Springer, 2013, pp. 108–122.
117. C. S. Bhagavatula, T. Noraset, D. Downey, Methods for exploring and mining tables on wikipedia, in: Proceedings of the ACM SIGKDD Workshop on Interactive Data Exploration and Analytics, 2013, pp. 18–26.
118. O. Lehmberg, D. Ritze, P. Ristoski, R. Meusel, H. Paulheim, C. Bizer, The mannheim search join engine, Journal of Web Semantics 35 (2015) 159–166.
119. B. Kruit, P. Boncz, J. Urbani, Extracting novel facts from tables for knowledge graph completion, in: International Semantic Web Conference, Springer, 2019, pp. 364–381.

120. O. Lehmberg, Web table integration and profiling for knowledge base augmentation, Ph.D. thesis (2019).
121. S. Zhang, K. Balog, Web table extraction, retrieval, and augmentation: A survey, ACM Transactions on Intelligent Systems and Technology (TIST) 11 (2) (2020) 1–35.
122. K. McKeown, Text generation, Cambridge University Press, 1992.
123. S. Lu, Y. Zhu, W. Zhang, J. Wang, Y. Yu, Neural text generation: Past, present and beyond, arXiv preprint arXiv:1803.07133.
124. K. Lin, D. Li, X. He, Z. Zhang, M.-t. Sun, Adversarial ranking for language generation, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30, Curran Associates, Inc., 2017, pp. 3155–3165.
125. Y. Zhang, Z. Gan, K. Fan, Z. Chen, R. Henao, D. Shen, L. Carin, Adversarial feature matching for text generation, in: Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR. org, 2017, pp. 4006–4015.
126. A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language models are unsupervised multitask learners, OpenAI Blog 1 (8) (2019) 9.
127. T. Zhang, J. Zhang, C. Huo, W. Ren, Automatic generation of pattern-controlled product description in e-commerce, in: The World Wide Web Conference, 2019, pp. 2355–2365.
128. L. Dong, S. Huang, F. Wei, M. Lapata, M. Zhou, K. Xu, Learning to generate product reviews from attributes, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, 2017, pp. 623–632.
129. J. Ni, J. McAuley, Personalized review generation by expanding phrases and attending on aspect-aware representations, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2018, pp. 706–711.
130. H. Mei, M. Bansal, M. R. Walter, What to talk about and how? selective generation using LSTMs with coarse-to-fine alignment, arXiv preprint arXiv:1509.00838
131. A. Nenkova, K. McKeown, A survey of text summarization techniques, in: Mining text data, Springer, 2012, pp. 43–76.
132. I. Sutskever, O. Vinyals, Q. V. Le, Sequence to sequence learning with neural networks, in: Advances in neural information processing systems, 2014, pp. 3104–3112.
133. R. Nallapati, B. Zhou, C. Gulcehre, B. Xiang, et al., Abstractive text summarization using sequence-to-sequence RNNs and beyond, arXiv preprint arXiv:1602.06023.
134. B. Liu, L. Zhang, A survey of opinion mining and sentiment analysis, in: Mining text data, Springer, 2012, pp. 415–463.
135. K. Ravi, V. Ravi, A survey on opinion mining and sentiment analysis: tasks, approaches and applications, Knowledge-Based Systems 89 (2015) 14–46.
136. L. Zhang, S. Wang, B. Liu, Deep learning for sentiment analysis: A survey, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8 (4) (2018) e1253.
137. R. Zafarani, M. A. Abbasi, H. Liu, Social media mining: an introduction, Cambridge University Press, 2014.
138. S. Vieweg, A. L. Hughes, K. Starbird, L. Palen, Microblogging during two natural hazards events: what twitter may contribute to situational awareness, in: Proceedings of the SIGCHI conference on human factors in computing systems, 2010, pp. 1079–1088.
139. O. Okolloh, Ushahidi, or ‘testimony’: Web 2.0 tools for crowdsourcing crisis information, Participatory learning and action 59 (1) (2009) 65–70.
140. R. Goolsby, Lifting elephants: Twitter and blogging in global perspective, in: Social computing and behavioral modeling, Springer, 2009, pp. 1–6.
141. A. Schulz, P. Ristoski, H. Paulheim, I see a car crash: Real-time detection of small scale incidents in microblogs, in: Extended semantic web conference, Springer, 2013, pp. 22–33.
142. D. E. O’Leary, Twitter mining for discovery, prediction and causality: Applications and methodologies, Intelligent Systems in Accounting, Finance and Management 22 (3) (2015) 227–247.

143. K. Shu, A. Sliva, S. Wang, J. Tang, H. Liu, Fake news detection on social media: A data mining perspective, ACM SIGKDD Explorations Newsletter 19 (1) (2017) 22–36.
144. S. Brin, L. Page, The anatomy of a large-scale hypertextual web search engine.
145. L. Page, S. Brin, R. Motwani, T. Winograd, The PageRank citation ranking: Bringing order to the web., Tech. rep., Stanford InfoLab (1999).
146. J. M. Kleinberg, Authoritative sources in a hyperlinked environment, Journal of the ACM (JACM) 46 (5) (1999) 604–632.
147. S. Wasserman, K. Faust, et al., Social network analysis: Methods and applications, Vol. 8, Cambridge university press, 1994.
148. D. Knoke, S. Yang, Social network analysis, Vol. 154, SAGE Publications, Incorporated, 2019.
149. W. Xing, A. Ghorbani, Weighted PageRank algorithm, in: Proceedings. Second Annual Conference on Communication Networks and Services Research, 2004., IEEE, 2004, pp. 305–314.
150. X. Li, B. Liu, S. Y. Philip, Time sensitive ranking with application to publication search, in: Link Mining: Models, Algorithms, and Applications, Springer, 2010, pp. 187–209.
151. R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, Trawling the web for emerging cyber-communities, Computer networks 31 (11-16) (1999) 1481–1493.
152. G. W. Flake, S. Lawrence, C. L. Giles, Efficient identification of web communities, in: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, 2000, pp. 150–160.
153. G. W. Flake, S. Lawrence, C. L. Giles, F. M. Coetzee, Self-organization and identification of web communities, Computer 35 (3) (2002) 66–70.
154. A. Lancichinetti, S. Fortunato, Community detection algorithms: a comparative analysis, Physical review E 80 (5) (2009) 056117.
155. P. Bedi, C. Sharma, Community detection in social networks, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 6 (3) (2016) 115–135.
156. D. Zhang, J. Yin, X. Zhu, C. Zhang, Network representation learning: A survey, IEEE transactions on Big Data.
157. S. Bhagat, G. Cormode, S. Muthukrishnan, Node classification in social networks, in: Social network data analytics, Springer, 2011, pp. 115–148.
158. A. Grover, J. Leskovec, node2vec: Scalable feature learning for networks, in: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, 2016, pp. 855–864.
159. B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk: Online learning of social representations, in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, 2014, pp. 701–710.
160. F. D. Malliaros, M. Vazirgiannis, Clustering and community detection in directed networks: A survey, Physics Reports 533 (4) (2013) 95–142.
161. M. Xie, H. Yin, H. Wang, F. Xu, W. Chen, S. Wang, Learning graph-based poi embedding for location-based recommendation, in: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, 2016, pp. 15–24.
162. J. Tang, J. Liu, M. Zhang, Q. Mei, Visualizing large-scale and high-dimensional data, in: Proceedings of the 25th international conference on world wide web, 2016, pp. 287–297.
163. Z. Liu, V. W. Zheng, Z. Zhao, F. Zhu, K. C.-C. Chang, M. Wu, J. Ying, Distance-aware DAG embedding for proximity search on heterogeneous graphs, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.
164. J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, M. Sun, Graph neural networks: A review of methods and applications, arXiv preprint arXiv:1812.08434.
165. Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, P. S. Yu, A comprehensive survey on graph neural networks, arXiv preprint arXiv:1901.00596.
166. J. Wang, Encyclopedia of Data Warehousing and Mining, (4 Volumes), iGi Global, 2009.
167. D. Tanasa, B. Trousse, Advanced data preprocessing for intersites web usage mining, IEEE Intelligent Systems 19 (2) (2004) 59–65.

168. F. Ricci, L. Rokach, B. Shapira, Introduction to recommender systems handbook, in: Recommender systems handbook, Springer, 2011, pp. 1–35.

169. J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, Recommender systems survey, Knowledge-based systems 46 (2013) 109–132.

170. S. Zhang, L. Yao, A. Sun, Y. Tay, Deep learning based recommender system: A survey and new perspectives, ACM Computing Surveys (CSUR) 52 (1) (2019) 1–38.

171. F. Silvestri, et al., Mining query logs: Turning search usage data into knowledge, Foundations and Trends® in Information Retrieval 4 (1–2) (2009) 1–174.

172. A. Al-Hegami, H. Al-Omaisi, Data mining techniques for mining query logs in web search engines.

173. H. Cui, J.-R. Wen, J.-Y. Nie, W.-Y. Ma, Probabilistic query expansion using query logs, in: Proceedings of the 11th international conference on World Wide Web, 2002, pp. 325–332.

174. R. Baeza-Yates, C. Hurtado, M. Mendoza, Query recommendation using query logs in search engines, in: International conference on extending database technology, Springer, 2004, pp. 588–596.

175. M. Speretta, S. Gauch, Personalized search based on user search histories, in: The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05), IEEE, 2005, pp. 622–628.

176. B. J. Jansen, D. L. Booth, A. Spink, Determining the user intent of web search engine queries, in: Proceedings of the 16th international conference on World Wide Web, 2007, pp. 1149–1150.

177. K. Dave, V. Varma, et al., Computational advertising: Techniques for targeting relevant ads, Foundations and Trends® in Information Retrieval 8 (4–5) (2014) 263–418.

178. D. Hillard, S. Schroedl, E. Manavoglu, H. Raghavan, C. Leggetter, Improving ad relevance in sponsored search, in: Proceedings of the third ACM international conference on Web search and data mining, 2010, pp. 361–370.

179. M. Vlachos, C. Meek, Z. Vagena, D. Gunopulos, Identifying similarities, periodicities and bursts for online search queries, in: Proceedings of the 2004 ACM SIGMOD international conference on Management of data, 2004, pp. 131–142.

180. I. Weber, V. R. K. Garimella, E. Borra, Mining web query logs to analyze political issues, in: Proceedings of the 4th annual ACM web science conference, 2012, pp. 330–334.

181. P. M. Polgreen, Y. Chen, D. M. Pennock, F. D. Nelson, R. A. Weinstein, Using internet searches for influenza surveillance, Clinical infectious diseases 47 (11) (2008) 1443–1448.

182. C. Bizer, T. Heath, T. Berners-Lee, Linked Data—The Story So Far., Int. J. Semantic Web Inf. Syst. 5 (3) (2009) 1–22.

183. M. Schmachtenberg, C. Bizer, H. Paulheim, Adoption of the linked data best practices in different topical domains, in: International Semantic Web Conference, Springer, 2014, pp. 245–260.

184. S. Staab, R. Studer, Handbook on ontologies, Springer Science & Business Media, 2010.

185. H. Paulheim, Knowledge graph refinement: A survey of approaches and evaluation methods, Semantic web 8 (3) (2017) 489–508.

186. Q. Wang, Z. Mao, B. Wang, L. Guo, Knowledge graph embedding: A survey of approaches and applications, IEEE Transactions on Knowledge and Data Engineering 29 (12) (2017) 2724–2743.

187. H. Cai, V. W. Zheng, K. C.-C. Chang, A comprehensive survey of graph embedding: Problems, techniques, and applications, IEEE Transactions on Knowledge and Data Engineering 30 (9) (2018) 1616–1637.

188. P. Goyal, E. Ferrara, Graph embedding techniques, applications, and performance: A survey, Knowledge-Based Systems 151 (2018) 78–94.

189. P. Ristoski, J. Rosati, T. Di Noia, R. De Leone, H. Paulheim, RDF2Vec: RDF graph embeddings and their applications, Semantic Web 10 (4) (2019) 721–752.

190. R. Meusel, Web-scale profiling of semantic annotations in html pages, Ph.D. thesis (2017).

191. P. Petrovski, A. Primpeli, R. Meusel, C. Bizer, The WDC gold standards for product feature extraction and product matching, in: International Conference on Electronic Commerce and Web Technologies, Springer, 2016, pp. 73–86.
192. P. Ristoski, P. Petrovski, P. Mika, H. Paulheim, A machine learning approach for product matching and categorization, Semantic web (Preprint) (2018) 1–22.
193. Z. Zhang, M. Paramita, Product classification using microdata annotations, in: International Semantic Web Conference, Springer, 2019, pp. 716–732.