Fuzzy Online Reputation Analysis Framework

Edy Portmann
Information Systems Research Group, University of Fribourg, Switzerland

Tam Nguyen
Mixed Reality Lab, National University of Singapore, Singapore

Jose Sepulveda
Mixed Reality Lab, National University of Singapore, Singapore

Adrian David Cheok
Mixed Reality Lab, National University of Singapore, Singapore

ABSTRACT

The fuzzy online reputation analysis framework, or "foRa" (plural of *forum*, the Latin word for *marketplace*) framework, is a method for searching the Social Web to find meaningful information about reputation. Based on an automatic, fuzzy-built ontology, this framework queries the social marketplaces of the Web for reputation, combines the retrieved results, and generates navigable Topic Maps. Using these interactive maps, communications operatives can zero in on precisely what they are looking for and discover unforeseen relationships between topics and tags. Thus, using this framework, it is possible to scan the Social Web for a name, product, brand, or combination thereof and determine query-related topic classes with related terms and thus identify hidden sources. This chapter also briefly describes the youReputation prototype (www.youreputation.org), a free web-based application for reputation analysis. In the course of this, a small example will explain the benefits of the prototype.

INTRODUCTION

The Social Web consists of software that provides online *prosumers* (combination of *pro*ducer and con*sumer*) with a free and easy means of interacting or collaborating with each other. Consequently, it is not surprising that the number of people who read Weblogs (or short blogs) at least once a month has grown rapidly in the past few years and is likely to increase further in the foreseeable future. Blogging gives people the ability to express their opinions and to start conversations about matters that affect their daily lives. These conversations strongly influence what people think about companies and what products they purchase. The influence of these conversations on potential purchases is leading many companies to strategically conduct blogosphere scanning. Through this scanning, it is possible to identify conversations that mention a company, a brand, the name of high-profile executives, or particular products. Through participation in the conversations, the affected parties can improve the company's image, mitigate damage to their reputation posed by unsatisfied consumers and critics, and promote their products.

To proactively shield their reputation from damaging content, companies increasingly rely on online reputation analysis. Because consumer-created Web sites (such as blogs) have enhanced the public's voice and made it very simple to make articulated standpoints and, given the advances and attractiveness of search engines, these analyses have recently become more important. They can map opinions and influences on the Social Web, simultaneously determining the mechanisms of idea formation, idea-spreading, and trendsetting. In light of these factors, the intention of the foRa framework is to let communications operatives search the Social Web to find meaningful information in a straightforward manner. The term *foRa* originates from the plural of *forum*, the Latin word for marketplace. Thus, the foRa framework allows an analysis of reputation in online marketplaces and provides communications operatives—i.e., the companies concerned with reputation management—with an easy-to-use dashboard. This dashboard, which is an interactive user interface, allows the browsing of related topics.

This chapter is organized into six subchapters:

- The first subchapter—*Background*— provides the reader in four sections with definitions and discussions of the topic: the first section states the paradigms of the Social Web with respect to electronic business; the second section introduces Web search engines and their Web agents; the third section introduces the overall approach to overcome the gap between inexplicit humans and explicit machines; the last section illustrates a visual approach as a link between humans and machines. All of the sections of this subchapter, likewise, incorporate literature reviews.
- The second subchapter—*The Use of Search Engines for Online Reputation Management*—comprises two sections: the first explains reputation management and the second discusses online reputation analysis.
- The third subchapter—*The Fuzzy Online Reputation Analysis Framework*—demonstrates the whole chapter's underlying foRa framework. In doing so, the first section illustrates the framework briefly; the second section explains the building of the fuzzy grassroots ontology; the third section reveals the selection of its ontology storage system; and the fourth section presents the reputation analysis engine.
- The fourth subchapter—*YouReputation: A Reputation Analysis Tool*—presents the youReputation prototype. To provide the readers not only with an abstract framework but also an easy-to-use tool, the building of the *youReputation* (combination of *your* and *reputation*) prototype is also described.
- The fifth subchapter—*Future Research Directions*—discusses further emerging trends and promising fields of study.
- The last subchapter—*Conclusion*—summarizes the key aspects developed and suggests possible further improvements of the presented framework.

BACKGROUND

To explain the advantages of the framework, this subchapter illustrates the underlying fields of study. The first section introduces electronic business and explains the paradigms of the Social Web. Because the framework collects tags from folksonomies by dint of Web agents, the second section provides an overview of the functionalities of Web agents. The third section reveals the metamorphoses from folksonomies to ontologies. The fourth and last section aims to introduce interactive knowledge visualization as Topic Maps.

Electronic Business in the Social Web

In 1972, McLuhan and Nevitt predicted that, with technology, the consumer would increasingly turn into a producer and that the role of producers and consumers would begin to blur and merge. From this concept, the term *prosumer* was coined to express the mutual roles of producer and consumer in online relationships (McLuhan & Nevitt, 1972). As individuals become involved in online processes, their role shifts from passive to active. Thus, *social software* has created a new generation of consumers who are far more interested in companies and products than were the consumers of the past. Social software commonly refers to media that facilitate interactive information-sharing, interoperability, user-centered design, and collaboration. In contrast to erstwhile Web sites where users were limited to passive content browsing, a *social Web site* gives users the freedom to enter a conversation and, thus, to interact or collaborate with others. Examples range from folksonomies, mashups, social networking, and video sharing sites to Web applications, blogs, and wikis (O'Reilly, 2005).

Most online prosumers know the structure of the *Social Web* well, so they share their experiences with companies, brands, products, and services online. Social software is shifting the way in which people communicate by giving them the opportunity to contribute to discussions about anything. Social Web sites are amplifying voices in marketplaces and exerting far-reaching effects on the ways in which people buy. As a result, these Web sites have implications for companies and should be taken seriously while doing (electronic) business.

According to Meier and Stormer (2009), electronic business is defined as the exchange of services with the help of media to achieve added value. In electronic business companies, both public institutions and individuals can be prosumers, and the relationship therein generates added value for all involved. This relationship may take the form of either a monetary or an intangible contribution. A central need of an electronic business is to appropriately manage its relationships with consumers (Bruhn, 2002). As the Social Web is not moderated or censored, individuals can say anything they want, whether it is good or bad. This freedom indicates the need to manage relationships with consumers by carefully watching and, if necessary, interacting with them in an appropriate way (Scott, 2010). Because there are plenty of examples of how not to interact, this communication with consumers should be carefully considered and relinquished to communications operatives to optimize business relationships (Portmann, 2008). Increasingly, companies are looking to gain access to conversations and to take part in the dialogue. This strategy can be integrated into a customer relationship management (CRM) strategy for managing and nurturing the company's interactions with its stakeholders. In electronic business, cautious monitoring of the company's reputation in online marketplaces should be considered.

Now that we have introduced the prosumer paradigm, one can easily imagine the flood of information produced by all of them. Therefore, the next section briefly introduces the concept of Web agents and search engines.

Finding Appropriate Information in the Social Web

A *Web agent* is a program that accumulates information from the Web in an automated and methodical fashion. Primarily, Web agents are used to create a copy of visited sites for later processing by a search engine that allows fast and sophisticated searches. Hence, the Web agent initially starts with a list of sites to visit. While the agent visits this list, it identifies sources in the site and subjoins them into a crawl frontier list. Sources from the crawl frontier list are visited recursively in accordance with a set of conventions.

Using a Web agent, a *Web search engine (WSE)* stores information on the visited Web sites. The information of each site is analyzed, and the results of the analysis are stored and indexed for rapid searching later. Based on an index, a WSE later provides a listing of best-matching Web sites according to a search query. Since Manning et al. (2008), this process has become an accepted standard for information searches and an often-visited source of information-finding. WSEs typically present their results in a single list, called a hit list. The hits can consist of images, text, Web sites, and auxiliary types of documents such as multimedia files (Baeza-Yates & Ribeiro-Neto, 2011).

The foRa framework rests upon the principles of WSEs and Web agents. Based on a search query, our query engine connects to queries with underlying Web content. An important point is that, even though millions of hits may be found by WSEs, some sites may be more relevant, popular, or authoritative than others (Baeza-Yates & Ribeiro-Neto, 2011). A possible means of sorting the found Web sites is by exploring the associations between objects that provide different types of relationships and that are not apparent from isolated data. These analyses have increasingly been applied by search engines to provide a relevance rating.

Today, most search engines apply common operators to specify a search query, but some engines provide an advanced feature that allows for a definition of the distance between topics. In a similar manner, foRa finds and provides better results. Based on the built-in query engine, the framework employs methods to rank the results according to several different factors.

After we have demonstrated the functionality of Web agents and search engines, we now introduce a concept to minimize the gap between humans and machines: in the Social Web, human-made taxonomies can be collected and retooled into machine-understandable ontologies by Web agents. This is the topic of the following section.

From Folksonomies to Fuzzy Ontologies

About three millennia ago, the ancient Assyrians annotated clay tablets with small labels to make them easier to tell apart when they were filed in baskets or on shelves. The idea survived into the twentieth century in the form of the catalog cards that librarians used to record a book's title, author, subject, etc. before library records were moved to computers (Gavrilis et al., 2008). The actual books constituted the data; the catalogue cards comprised the metadata. Metadata in the Social Web are called *tags*: non-hierarchical keywords assigned to a piece of information, such as a uniform resource locator (URL), a picture, or a movie (Smith, 2008; Troncy et al., 2011). According to Peters (2009), the outcomes of collaboratively creating and manipulating tags to annotate and categorize content are *folksonomies* (a blend of *folk* and *taxonomy*). Tags are generally chosen informally and personally by a creator or by viewers (depending on the system used to describe the item) to aid searching. For this reason, tags are simple to create but generally lack a formal grounding, as intended by the *Semantic Web* (Voss, 2007). Through tags, value is added by structuring the information and ranking it in order of relevance to ease query searches, as outlined by Agosti (2007).

In the framework, folksonomies are used as a starting point to harvest collective knowledge, which is then normalized and converted into a machine-understandable ontology. This conversion marks a transition from a human-oriented Social Web to a machine-oriented Semantic Web; because both concepts are joined, it is labeled the *Social Semantic Web* (Breslin et al., 2009; Blumauer & Pellegrini, 2009). An *ontology* is a design model for specifying the world that consists of a set of types, relationships, and properties. According to Gruber (1993), an ontology is a "*formal, explicit specification of a shared conceptualization*." Ontologies offer a common terminology, which can be used to model a domain. A domain comprises the types of objects and concepts that exist and their properties and relations.

Through harvesting tags from folksonomies, a tagspace (a set of associated tags with related weights) is created in which semantic closeness is represented by distance. To achieve an allied tagspace (where all harvested tags are related to each other), it is essential to establish tags and their relationships to each other (Kaser & Lemire, 2007). As Hasan-Montero and Herrero-Solana (2006) suggested, the easiest way to find the similarity between two tags is to count the number of co-occurrences, i.e., the number of times the two tags are allocated to the same source. However, there are other measurements to establish similarity, such as locality-sensitive hashing (where the tags are hashed in such a way that similar tags are mapped to the same set with a high probability) and collaborative filtering (where several users define tags and their relations jointly). Each of these methods produces relationships among tags, and each offers a semantically-consistent picture in which nearly all of the tags are related to each other to some degree (Setsuo & Suzuki, 2008).

At present, our intention for the Semantic Web is to amend the bottom-up attempt of the Social Web in a top-down manner (Cardoso, 2007). The fundamental aim is a stronger knowledge representation, as can be achieved with folksonomies, for example. *Fuzziness* can overcome the gap between folksonomies and ontologies because fuzziness corresponds to the way in which humans think (Werro, 2008) and it is, thus, suitable for characterizing vague information and helps to more efficiently handle real-world complexities. One possible way to use these advantages is through *fuzzy clustering algorithms*, which allow modeling of the uncertainty associated with vagueness and imprecision through mathematical models (Oliveira & Pedrycz, 2007; Bezdek et al., 2008).

To build the ontology, the tagspace will first be clustered with random initialization by a fuzzy clustering algorithm into pre-computed classes (Portmann & Meier, 2010). Thus, the number of classes can be determined by various methods. In the section about the fuzzy grassroots ontology, a simple method is explained. Because this is fuzzy clustering, it is possible for each tag to belong to one or more classes with different degrees of membership. Thus, it is possible that linguistic issues such as homographs, homophones, and synonyms, as well as their overlaps, can be identified. For example, because every tag can belong to different classes, it is possible that the tag *"bow"* can belong to either the class *"ship"* or the class *"weapon."* Because the harvested tags will be normalized, it is, furthermore, possible to spot homophones such as *"bow"* and *"bough"* (Fig. 1).

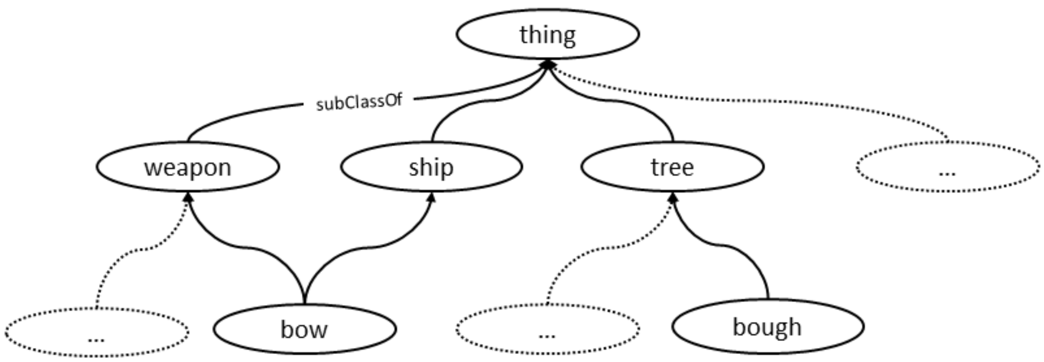

Figure 1. A Fragment of an Ontology.

The creation of an ontology is an iterative process where the first node in the hierarchy is stored in a *knowledge database* and the process is repeated. To complete an ontology of the created first-nodes, they are clustered again into pre-computed classes, making it possible to get a second-step, followed by a third-step hierarchy, and so forth. During this iterative process, all of the built hierarchies—now called ontologies—are collected and stored in a knowledge database by an ontology tool. Using this established ontology, it is possible for both humans and machines to recognize dependencies. For example, by trailing up a "*watercraft*" ontology, it is feasible to deduce that "*boats*" are related to "*ships*." Furthermore, it is possible to recognize that, besides "*watercrafts*," there are also "*aircrafts*," for example.

Now that we have demonstrated the automatic processing of folksonomies to ontologies, we want to show how the machine-oriented ontologies can be easily made available to humans again. Therefore, the next section introduces interactive visualization techniques to let humans experience these ontologies.

Ontology-based Knowledge Representation as Interactive Topic Maps

Visualization techniques should empower people to spot patterns in Web data, identify areas that need additional analysis, and make sophisticated decisions based on these patterns (Zudilova-Seinstra et al., 2008). The human capability to converse, communicate, reason, and make rational decisions in an environment of imprecision, uncertainty, incomplete information, and partial truth will be supported by this visualization. The manner in which people experience and interact with visualizations affects their understanding of the data; people benefit from the ability to visually manipulate and explore. Visual interaction can support gut instincts and provide an instrument to both substantiate theses and support viewpoints.

Besides mere visualization, an interesting feature of this method is the ability to discover hotspots through an interactive possibility. To increase the ability to explore the data (and thus, to better understand the results), an effective integration of the *visualization* and *interaction applications* is important. According to Ward et al. (2010), interactive visualization can be used at each step of knowledge discovery, such as the process of automated data mining for characterizing patterns in the data. Nevertheless, the field of analyzing data to identify relevant concepts, relations, and assumptions, combined with the conversion of data into machine language, is known as *knowledge representation* (Van Harmelen et al., 2007; Weller, 2010). Because knowledge is used to achieve intelligent behavior, the fundamental goal of knowledge representation is to present data in a manner that will facilitate reasoning; here, knowledge representation and reasoning are seen as two sides of the same coin. In the field of artificial intelligence, problem-solving can be simplified by an appropriate choice of knowledge representation (Agosti et al., 2009; Sirmakessis, 2005). Presenting data in the right way makes certain problems easier to solve. On the one hand, our ontology provides machines with a general knowledge of vague human concepts and, on the other hand, the ontology-based, interactive visualization of this knowledge through Topic Maps helps people to find related patterns. Importantly, for a straightforward search of a company's online reputation, this interactive visualization can be used as a starting point; the similar topics and tags are visualized closer and the more dissimilar topics and tags are placed farther apart.

Within the framework, a dashboard is used to visualize topics and tags using *interactive Topic Maps*. These maps rely on a formal model that subsumes those of traditional identification guides (such as indexes, glossaries, and thesauri) and extends them to cater to the additional complexities of digital information. Interactive Topic Maps are also an international standard technology for qualifying knowledge structures (Pepper, 2010). They provide a way to visualize how a topic is connected to other topics. Based on these maps, the findability of information is improved. Related tags are displayed using interactive Topic Maps, enabling a communications operative to find related tags by browsing. The topic contains a set of related tags presented on the screen and allows the clicking of any tag that appears around the topic. Comparable to Zadeh's (2010) z-mouse, the dashboard allows the user to zoom in and out (akin to the zooming function in Google Maps) to find related topics and associated tags for a stated query. Hence, this interactive visualization helps to identify the previously unknown but related topics and tags and to thereby gain new knowledge.

The next subchapter introduces the use of Web search engines for online reputation management. It is shown why reputation management is such an important point in doing business in the Social Semantic Web.

THE USE OF SEARCH ENGINES FOR ONLINE REPUTATION MANAGEMENT

The practices of monitoring, addressing, or mitigating search engine results pages or mentions in social media are summarized as online reputation management (ORM). The first section of this subchapter gives a short introduction into reputation management, both online and general. The second section illustrates the necessity of online reputation analysis, a reputation management task that is conducted by communications operatives.

Online Reputation Management

Shakespeare defined reputation as the *"purest treasure mortal times afford,"* Abraham Lincoln labeled it a *"tree's shadow,"* and Benjamin Franklin pointed out that *"it takes many good deeds to build a good one, and only one bad to lose it."* Because of reputation, companies and other institutions have failed or succeeded. *Reputation* can be defined as a social evaluation of a group of entities toward a person, a group of people, or an organization regarding a certain criterion. More simply, reputation is the result of what you do, what you say, and what other people say about you (Gaines-Ross, 2008). Although reputation is built upon trust, in turn, trust is an outcome of a sound reputation; these two concepts form a symbiotic relationship to each other (Picot et al., 2003; Ebert, 2009; Klewes & Wreschniok, 2009). Chun (2005) considered a company's reputation to be a synoptic standpoint of the perceptions held by all of the germane stakeholders of a company, that is, what communities, creditors, consumers, employees, managers, the media, and suppliers believe that the organization stands for and the associations they make with it. A sound reputation sustainably strengthens a company's position in the struggle for profitable clients in the hunt for talent and in its affiliations with stakeholders.

Reputation management, if it is to evolve as a prevailing business task, rests on the basis of public relations (PR). In the Social Web, its form and character encompasses social media and such communication platforms as personal computers, laptops, and mobile phones (Phillips & Young, 2009). Companies with stronger positive reputations are able to attract more and better customers; their customers are more loyal and buy broader ranges of products and services. Because the market believes that these companies will deliver sustained earnings and future growth, they have higher price-to-earnings ratios, higher market values, and lower costs of capital. Moreover, in an economy where 70% to 80% of equity is derived from intangible assets that are difficult to assess, companies are particularly vulnerable to anything that damages their reputation (Eccles et al., 2007). Reputation management is becoming a paradigm in its own right as a consistent way of looking at a company and at its business performance. As an aid for communications operatives, a number of ORM applications already exist, but only some of them are free and only a handful deal with reputation analysis (Gunelius, 2010). In the literature, several approaches have described how to identify reputation; most of them rely on the management task of reputation analysis (Fombrun & Wiedmann, 2001; Eisenegger & Imhof, 2007; Ingenhoff & Sommer, 2008). Nevertheless, the significance of these analyses is critical considering that a negative search engine result will often be clicked first when listed with a company's Web site. An instrument to measure the information on a company's reputation is *reputation analysis*. Reputation analysis involves scanning and monitoring reputation data. Therefore, the next section inducts the reader in reputation analysis in more depth.

Online Reputation Analysis

With its emphasis on influencing search engine results to protect a company, *online reputation analysis* can be viewed as a field that relates to other areas of online marketing, such as word-of-mouth marketing (WOMM), search engine optimization (SEO), and PR. An organization must present the same message to all of its stakeholders to convey coherence, credibility, and ethics. Communications operatives can help to build this message by combining the vision, mission, and values of the company. Corporate communication can be both internal (e.g. employees and stakeholders) and external (e.g. agencies, channel partners, media, government, industry bodies and institutes, educational institutes, and the general public) (Röttger, 2005). According to Van Riel and Fombrun (2007), corporate communication is the set of activities required to manage and orchestrate all of the internal and external communications, which are aimed at creating favorable starting points with the stakeholders on whom the company depends. It consists of the accumulation and dissemination of information with the common goal of enhancing the organization's ability to retain its license to operate.

Hence, as reported by Ingenhoff (2004), the goal of *scanning* for a company's reputation, on the one hand, is the early detection of changes in the environment of the company that may affect or restrict the company's scope. On the other hand, new sectors can be detected through scanning. To position itself as an expert and opinion leader and to realize new opportunities, the company can occupy these new sectors. Another goal of this approach is to evaluate the reputations of competitors; occasionally, a competitor will launch an unknown product or a new production method that can be detected through scanning. Nevertheless, a challenge of reputation scanning is the prevention of flooding caused by vast amounts of data. Issues must be summarized into manageable topics, and their changes must be surveyed in the ensuing permanent *monitoring* to avoid surprises. Monitoring is a method of reputation analysis that is equivalent to scanning but watches a selected range of topics.

Though most executives know the value of their own reputations, it is also not uncommon for companies to hire professionals to manage their reputation risks. According to Eccles et al. (2007), effectively managing reputational risk involves assessing the company's reputation among stakeholders, evaluating the company's real character, closing reputation-reality gaps, monitoring changing beliefs and expectations, and placing a particular executive in charge of these tasks. The assignment of this executive typically consists of tracking the actions of an entity and the opinions of other entities about those actions, reporting on the actions and opinions, and reacting to the report, creating a feedback loop.

Now that we have demonstrated all important background information for our framework, we present in the ensuing subchapter the foRa framework.

THE FUZZY ONLINE REPUTATION ANALYSIS FRAMEWORK

The foRa framework permits searching of the Social Web to find reliable information on reputation. Using this framework, it is possible to scan the Web according to a query to determine topic classes with related tags and, thus, to identify hidden information. The first section provides an overview of the framework, followed by an explanation of the three main parts of the framework. The second section explains the creation of the fuzzy grassroots ontology. The third section describes the storage of the established ontology with ontology tools. The fourth section presents the reputation analysis engine.

Component Overview

As discussed, ORM deals with monitoring, addressing, or mitigating mentions in social media. It grew out of the perception of the significant influence that a Web search could have on business and the desire to change unpleasant results. Herein, we intend to illustrate our approach for reputation analysis in the Social Web. The foRa framework consists of three main parts: a fuzzy grassroots ontology where collected data will be converted into an ontology, an ontology storage system for the established ontology, and a reputation analysis engine where a communications operative can identify reputation information using a dashboard (Fig. 2).

Figure 2 indicates the alternating roles of prosumers with the arrow around the framework. The topmost layer in our framework includes the producer role and illustrates, thereby, a part of the Social Web. However, the Social Web also contains the consumer role of consuming, for instance, annotation and Web sources. Therefore, this layer should be pictured as steadily moving from the producer to the consumer role. The bottommost layer illustrates the components of the Semantic Web stack used by the framework; this layer does not involve prosumer roles because it is not comprised of the whole stack. The middle layer illustrates the structure of the framework.

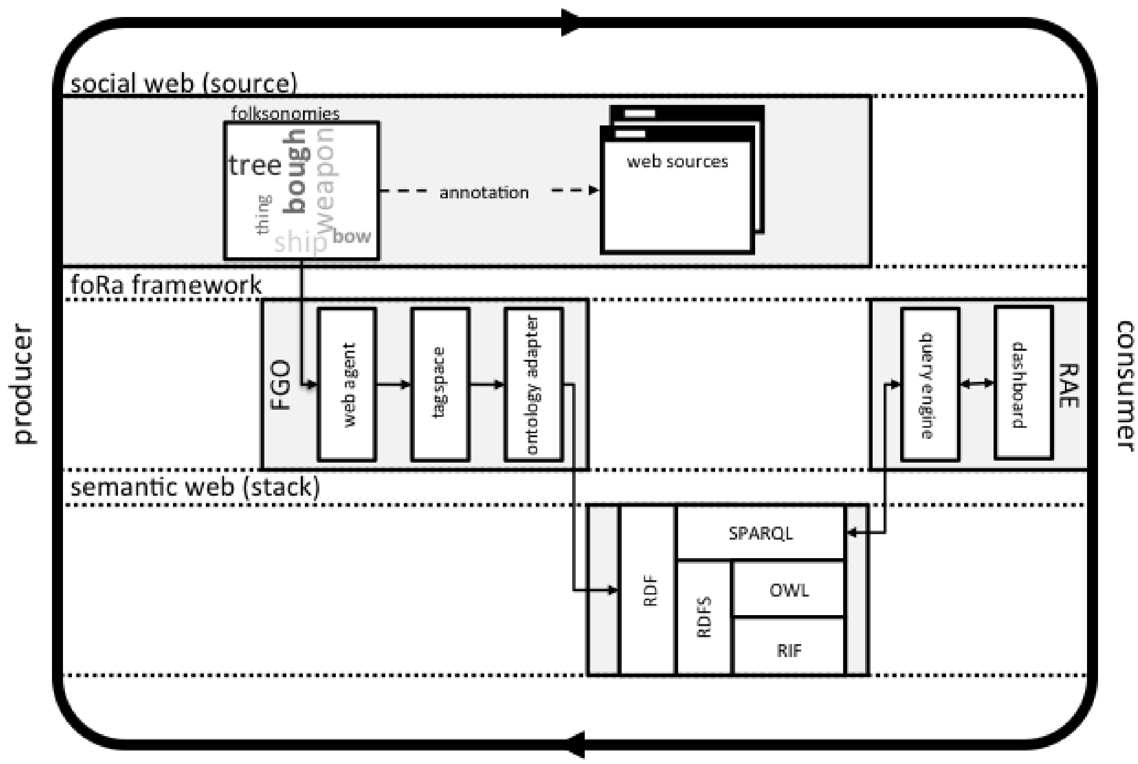

Figure 2. The foRa Framework Architecture.

The next section presents the core of the foRa framework, the fuzzy grassroots ontology (in Fig. 2, abbreviated with "FGO"). As already envisaged, this fuzzy grassroots ontology marks the handover of vague, human-created knowledge to machines.

The Fuzzy Grassroots Ontology

The fuzzy grassroots ontology comprises three elements. The first element is a *Web agent* that constantly crawls the Social Web, looking for tags and the underlying Web sites. In fact, the fuzzy grassroots ontology relies on not one but several Web agents. Agents identify all tagged sources and subjoin them into a crawl frontier list. During this process, the tags are normalized and the underlying sources are ranked.

The second element is the creation and plotting of the *tagspace*. The previously collected and normalized tags are linked to each other using a metric function, and distance metrics identify the distance between each two individual tags. After this step, all of the tags are linked to each other and plotted onto a tagspace, which is the input for the ontology adaptor.

The *ontology adaptor* separates the plotted tagspace into hierarchies of classes with the help of Bezdek's (1981) FCM (fuzzy c-means) algorithm. To build our ontology, we clustered the tagspace with random initialization using FCM. The tag nearest to the center names the class, and the other tags—including the eponym itself—are stored in this class by name and the membership degree for belonging to the class.

Web agents first used collected tags from folksonomies to establish a fuzzy grassroots ontology. In this sense, the ability to find high-quality sources is important for overcoming information overload. Collaborative filtering, or recommender systems, can identify high-quality sources that utilize individual knowledge. One known algorithm that has proven to be successful in automatically identifying high-quality sources within a hyperlinked environment is Kleinberg's (1998) *Hyperlink-Induced Topic Search*

(HITS) algorithm. HITS starts with a small root set of documents and moves to a larger set T by adding up documents that link to and from the documents in the root set. The goal of the algorithm is to identify hubs (i.e., documents that link to numerous high-quality documents) and authorities (i.e., documents that are linked from numerous high-quality documents). The hyperlink structure of the documents in T is given by the adjacency matrix A, where A_{ij} denotes whether there is a link from document d_i to document d_j. Using this matrix A, a weighting algorithm constantly updates the hub weight and authority weight for each document until the weights converge. Essentially, the hubs and authorities are the documents with the biggest values in the main eigenvectors of $A^T A$ and AA^T, respectively. HITS is used by the framework to rank all of the Web sites in combination with their identified tags according their relevance. Later, during a search, these ranked sources are then displayed according to various context dimensions.

A well-known problem with folksonomies is that typing errors can occur because there is no editorial supervision and people choose their own tags to annotate Web sources. This problem leads to overlapping but only slightly related terms in the underlying ontology. Certainly, it can be assumed that a search system can find relevant information despite misspelling in tags because queries could contain the same mistakes, but the necessity of a fault-tolerant treatment of queries soon becomes clear. According to Lewandowski (2005), one has to distinguish between different types of typing error strategies, such as dictionary-based and statistical approaches. Dictionary-based approaches compare entered query terms with a dictionary and if the dictionary does not cover the query term, they search for similar terms. Statistical methods refer misspellings with no or only a few hits to the most commonly used similar syntax. To determine phonetic similarity, tags will be reduced to a code that is able to conform to similar tags. A well-known basic example for the English language is the Soundex algorithm for indexing names by sound (Russell, 1918; Russell, 1922). Algorithm 1 illustrates the method of Russell's Soundex algorithm. The goal of this method is to encode homophones to the same representation so that they can be matched, despite their minor differences in spelling. The algorithm mainly encodes consonants; a vowel will not be encoded unless it is the first letter.

1. *Capitalize all of the letters in the word and drop all of the punctuation marks.*
2. *Retain the first letter of the word.*
3. *Change all occurrence of the letter:*
   $"A," "E," "I," "O," "U," "H," W","Y" \rightarrow 0$
4. *Replace consonants with digits as follows:*
   $"B," "F," "P," "V" \rightarrow 1$
   $"C," "G," "J," "K," "Q," "S," "X," "Z" \rightarrow 2$
   $"D," "T" \rightarrow 3$
   $"L" \rightarrow 4$
   $"M,""N" \rightarrow 5$
   $"R" \rightarrow 6$
5. *Collapse adjacent identical digits into a single digit of that value.*
6. *Remove all non-digits after the first letter.*
7. *Return the starting letter and the first three remaining digits. If needed, append zeroes to make it a letter and three digits.*

Algorithm 1. Soundex Algorithm.

A major advantage of the utilization of a Soundex algorithm is that the correctly spelled ontology terms can be used as auto-completion and auto-suggestion while the user is typing search terms into the dashboard.

However, after all of the tags have been collected and normalized, they need to be sorted. Because our Web agents are constantly crawling through the Web, this sorting process must be periodically repeated.

The tagspace is a representation of a consistent picture and serves as the input for the ontology adaptor. Several steps are required to plot the tagspace from the found tags. The first step is to define the relationship of the various found tags. To define these relationships, variations of the Minkowski metric are normally used:

$$d_M(j,k) = \left(\sum_{i=1}^{n} \left|x_{ji} - x_{ki}\right|^p\right)^{1/p} \quad (1)$$

Here, $d_M(j,k)$ denotes the distance of the objects j and k, x_{ji} and x_{kj} the value of the variable i for the object j and k ($j = 1,2,\dots,n$), and p (≥ 1) the Minkowski constant. The critical factor in this equation is to obtain the constant p, which defines the Minowski metric. A simple Minkowski metric-based coefficient that can be used to measure the semantic correlation between tags is the Jaccard similarity coefficient $d_J(A,B)$. Let A and B be the sets of resources characterized by two tags. Relative co-occurrence is ascertained with the following formula:

$$d_J(A,B) = \frac{|A \cap B|}{|A \cup B|} \quad (2)$$

That is, relative co-occurrence is identical to the partition among the amount of resources in which tags co-occur and the amount of resources in which either of the two tags appear. This collection method causes tags to become united and offers a semantically consistent picture in which nearly all of the tags are related to each other. This semantically consistent picture is referred to as the tagspace.

To begin the point representation, it is necessary to set a limitation for the tagspace. The plotting algorithm starts with a number of seed points (Algo. 2). Some seed points will be referred from the seeds, but they are limited to a certain depth. Child point locations are computed based on Bourke's (1997) algorithm, which calculates the intersection of two or three circles.

1. *Create the point list from a number of seeds with a predefined depth and select one source point.*
2. *Select each point in the list except the selection point.*
3. *Calculate the plotted points that are within a given distance to the selected point.*
4. *Check the number of plotted points that have a relationship with the current point.*
   a. *If no plotted points are detected, then draw the current point with a random position.*
   b. *If there is one plotted point detected, then draw the current point with the same y but with an x value that is calculated to fit the distance.*
   c. *If there are two plotted points detected, then draw the current point as one of the two intersections point of two circles whose centroids and radii are the two plotted points and their distances to the current point, respectively.*
   d. *If there are three plotted points detected, then draw the current point as the intersection of the three circles whose centroids and radii are the three plotted points and their distances to the current point, respectively.*
5. *Return to Step 2 for the next point.*

Algorithm 2. Plotting the Points.

After the found and normalized tags have been united, assorted, and plotted into a tagspace, a machine-understandable ontology can be established. The algorithm allocates the position of each point in the tagspace. Based on this algorithm, we can easily show the necessary points in the selected region, which is very effective for supporting a zoom function. Another parameter to take into account is the constant variability of the underlying data. We are familiar with the idea that data are at fixed values to be analyzed, but here, they are constantly moving around. In fact, they change every second, hour, or week. This consideration is legitimate because most data come from the real world, where no absolutes exist. The trends or demands of the Web can change dramatically. To interact with *live data*, we need to continually update the data and distance among the tags in the tagspace. As a result, the plotting algorithm described above is able to provide a good perspective on moving data.

The ontology adaptor can be described as follows. All n tags plotted in the tagspace will be sorted by the fuzzy c-means (FCM) algorithm (Algo. 3). This algorithm attempts to split a limited collection of elements $X = \{x_1, \dots, x_n\} \subseteq \mathbb{R}^n$ into an assortment of c fuzzy classes according to a specified condition. Assigning cluster numbers c ex ante is a common problem in clustering. In this case, to roughly define the number of clusters, we use the following rule:

$$c \approx \sqrt{n/2} \tag{3}$$

In fuzzy clustering, each point has a degree of belonging to a class using fuzzy logic rather than belonging to one particular class. Thus, points on the edge of a class may participate to a less significant degree than points in the center of a class. The degree of membership is u_{ik} in the interval $[0..1]$. The greater u_{ik} is, the stronger the membership of an element x_k to the class i will be. Hence, for each point x, there is a coefficient denoting participation at the k^{th} level $u_k(x)$. Thus, the FCM algorithm is based on the minimization of an objective function:

$$J_{(U,V,X)} = \sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik})^m d^2(v_i, x_k) \tag{4}$$

where m is the weighting exponent (or fuzzifier), u_{ik} is the membership degree of element x_k to class i, and $d(v_i, x_k)$ is the distance of x_k to v_i, represented by the prototype v_i. Characteristically, the sum of all of the coefficients $u_k(x)$ is defined as 1:

$$\forall_x \left(\sum_{k=1}^{c} u_{ik}(x) = 1 \right) \tag{5}$$

By FCM, the focal point of a class is the average of all of the points, each weighted by its amount of belonging to the class:

$$v_i = \frac{\sum_x u_{ik}(x)^m x}{\sum_x u_{ik}(x)^m} \tag{6}$$

The amount of belonging is associated with the inverse of the distance to the heart of the class:

$$u_{ik}(x) = \frac{1}{d(v_i, x_k)} \tag{7}$$

After the coefficients are normalized and fuzzified with a true parameter $m(> 1)$, their sum is 1. In other words, the weighting exponent is adjusted with parameter m. This leads to:

$$u_{ik}(x) = \frac{1}{\sum_j \left(\frac{d(v_i, x_k)}{d(v_j, x_k)} \right)^{2/(m-1)}} \tag{8}$$

For m equal to 2, this method is the same as normalizing the coefficients linearly so that their sum is equal to 1. When m is close to 1, the class center closest to the point is given a considerably larger weight relative to the others.

1. *Select an amount of classes with Formula (3) above.*
2. *Assign coefficients randomly to each point in the classes.*
3. *Reiterate until the algorithm has converged (that is, the adjustment of the coefficients between two iterations is no more than ε, a given sensitivity boundary value):*
   a. *Calculate the centroid for each class, using Formula (6) above.*
   b. *For each point, compute its coefficients within the class, using Formula (8) above.*
4. *Reiterate Steps 1 to 3 for every class until there is only one term left in the class.*
5. *Concatenate all of the same terms together.*

Algorithm 3. Fuzzy C-Means Algorithm.

Step 5 of Algorithm 3 is necessary because the terms can belong to more than one class (by drawing on the FCM algorithm). Nevertheless, using the proposed method, a model can be derived with several classes that the term belongs in to a certain degree, dependent on the degree of membership. By Step 4, the procedure is repeated until we have a class with a single tag in it; this tag forms the tip of the hierarchy. Figure 3 graphically indicates how the conversion of tags to ontologies is executed. Starting from the left, the algorithm splits the tagspace using FCM, denoted according to the mathematical perspective. The ontological perspective shows the classification of tag A (eponym of the class A). The relationship (along with the distance) to the other classes (B, C, D, etc.) and also to the tags of each class is stored in the ontology storage system.

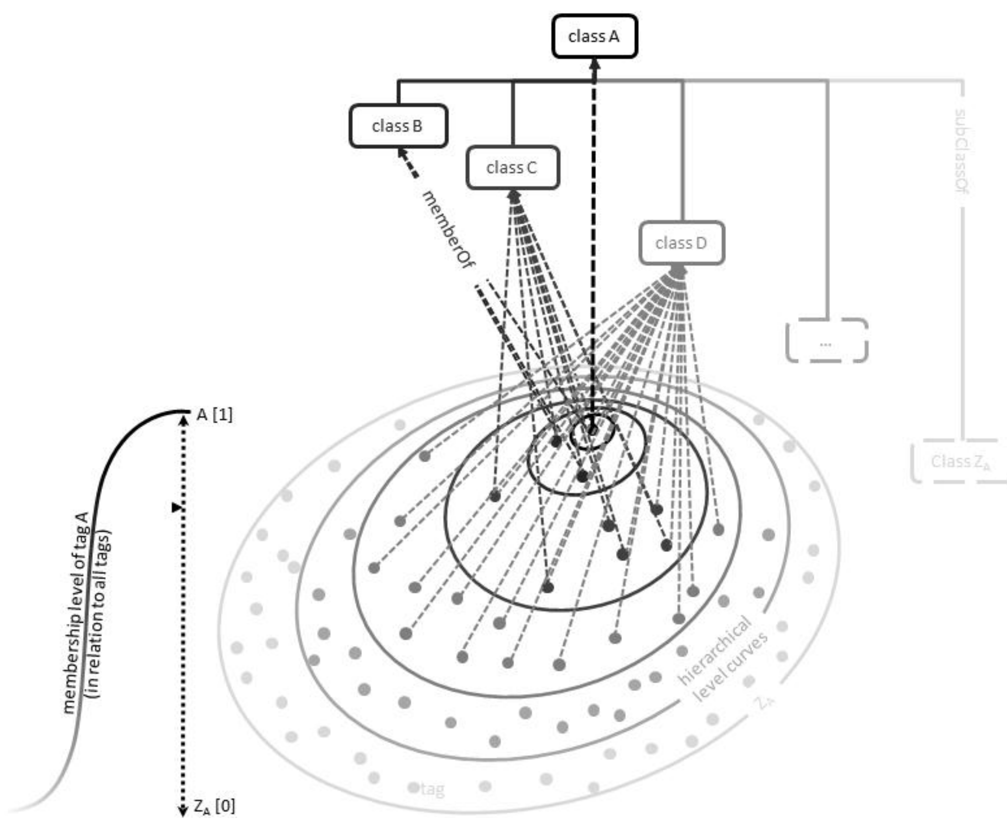

Figure 3. Schematic Representation of the Ontology-building Process.

However, the hierarchy of all of the classes is stored using the ontology tool, so we obtain several hierarchies that are jointly called ontology. The ranked Web sites that belong to the single tags are stored separately but linked to the ontology.

The created fuzzy grassroots ontology now needs to be stored with an adequate ontology tool. Therefore, the next section presents several recent ontology tools. Since we do not implement such an ontology tool by ourselves but rely on the World Wide Web Consortium (W3C) recommendations, this ensuing section reveals also our selection of an ontology tool to store and process the fuzzy grassroots ontology in an effective manner.

The Ontology Storage System

Several Semantic Web or ontology tools have recently been developed. In this section, we analyze and classify the most important of these tools. Many available applications are academic prototypes, meaning that most of the implementation in the query language aims to support but not to provide the necessary programming and administrative abilities to make them operational within a real working environment. Besides the emerging commercial software that supports ontology, an increasing number of ontology applications boost the advancement of ontology storage and query support. In this section, we compare the most common ontology tools. Accordingly, we select the most attractive ontology tool and corresponding query language for our framework. Table 1 lists different ontology tools according to various categories.

Tool	Cat.	Note
AllegroGraph RDFStore (4.2)	*OS*	*AllegroGraph RDFStore is a modern, high-performance, persistent RDF graph database. AllegroGraph uses disk-based storage, enabling it to scale to billions of triples while maintaining superior performance. AllegroGraph supports SPARQL, RDFS++, and Prolog reasoning from numerous client applications; (http://www.franz.com/agraph/allegrograph).*
COE (5.0)	*OE*	*COE is a project whose goal is to develop an integrated suite of software tools for constructing, sharing, and viewing OWL-encoded ontologies based on CmapTools, concept-mapping software used in educational settings, training, and knowledge capturing. Concept maps provide a human-centered interface to display the structure, content, and scope of an ontology; (http://www.ihmc.us/groups/coe).*
HOZO (5.2.30)	*OE*	*Hozo is an ontology editor with multi-window functions, globalization support, zooming (Concept Map), and other functions to make the editor more user-friendly; (http://www.hozo.jp).*
Jena (2.6.4)	*OA OE OS*	*Jena is a Java framework for building Semantic Web applications. It provides a programmatic environment for RDF, RDFS and OWL, and SPARQL and includes a rule-based inference engine; (http://jena.sourceforge.net).*
KAON (1.2.9)	*OS*	*KAON is an ontology management infrastructure targeted for business applications. It includes a comprehensive tool suite allowing easy ontology creation and management. Persistence mechanisms of KAON are based on relational databases; (http://sourceforge.net/projects/kaon).*
Major-ToM (2.0)	*TM*	*The Merging Topic Maps Engine (MaJorToM) project was founded to develop a light-weight, merging, and flexible Topic Maps engine satisfying different business use cases. The engine provides a couple of new features listed above to other engines based on Topic Maps API version 2.0; (http://code.google.com/p/majortom).*
Networked Planet (1.3)	*TM*	*The Networked Planet Web3 Platform is a complete solution for creating, organizing, and publishing structured semantic data. The Web3 platform stores and manages data in a schema-less data store, allowing complete flexibility in the shape of the data stored; (http://www.networkedplanet.com/Products/Web3).*
OBO-Edit (2.1.11)	*OE*	*OBO-Edit, an open-source ontology editor written in Java, is optimized for the OBO biological ontology file format. It features an easy-to-use editing interface, a simple but fast reasoner, and powerful search capabilities. OBO-Edit was developed by the Berkeley Bioinformatics and Ontologies Project and is funded by the Gene Ontology Consortium; (http://oboedit.org).*
Ontopia (5.1.3)	*TM*	*Open-source tools for building, maintaining, and deploying Topic Maps-based applications; (http://www.ontopia.net).*

OWL-API (3.2.2)	*OA*	*The OWL API is a Java API and reference implementation for creating, manipulating, and serializing OWL Ontologies; (http://owlapi.sourceforge.net).*
Protégé (4.1)	*OA OE*	*Protégé is a free, open-source ontology editor and knowledge base framework. The Protégé platform supports two main ways of modeling ontologies via the Protégé-Frames and Protégé-OWL editors. Protégé ontologies can be exported into a variety of formats, including RDF(S), OWL, and XML Schema. Protégé is based on Java, is extensible, and provides a plug-and-play environment that makes it a flexible base for rapid prototyping and application development; (http://protege.stanford.edu).*
REDLAND (1.0.13)	*OS*	*Redland, a set of free software C libraries supporting RDF, providing storage for graphs in memory and persistently with Sleepycat/Berkeley DB, MySQL 3-5, PostgreSQL, AKT Triplestore, SQLite, files, or URIs. It supports multiple syntaxes for reading and writing RDF as RDF/XML, N-Triples and Turtle Terse RDF Triple Language, RSS, and Atom syntaxes via the Raptor RDF Syntax Library and querying with SPARQL and RDQL using the Rasqal RDF Query Library; (http://librdf.org).*
Ruby Topic Maps (2.0)	*TM*	*Ruby Topic Maps (RTM) is a Topic Maps engine for the Ruby programming language. It can be used alone or together with other frameworks such as Ruby on Rails; (http://rtm.topicmapslab.de).*
Semantic Studio (1.7)	*OE*	*Semantic Studio is an ontology development tool with presentations in various formats, including visual presentation and persistence into semantic repositories, file systems, or databases. It allows the development of ontologies by using different presentations, among which there is an inner kernel presentation on which all other presentations are based. In our terms, all presentations are re-presentations of the kernel presentation. The kernel presentation is close to the "in-memory" model of Jena, but it differs in many respects, which we regard as further abstraction from presentation details; (http://www.w3.org/2001/sw/wiki/Semanticstudio).*
Sesame (2.0)	*OS*	*Sesame is a Java framework for storing, querying, and inferencing for RDF. It can be deployed as a Web server or used as a Java library. Features include several query languages (SeRQL and SPARQL), inferencing support, and RAM, disk, or RDBMS storage; (http://sourceforge.net/projects/sesame).*
SOFA-API (0.3)	*OA*	*SOFA (Simple Ontology Framework API) is an open-source project aimed at the development of an integral software infrastructure and a common development platform for various ontology-oriented and ontology-based software applications; (http://sofa.projects.semwebcentral.org).*
tinyTiM (2.0)	*TM*	*A very small and easy-to-use Topic Map engine implementing the TMAPI interfaces; (http://tinytim.sourceforge.net)*
TRIPLE	*OS*	*TRIPLE is an RDF query, inference, and transformation language for the Semantic Web; (http://triple.semanticweb.org)*

Table 1. List of Ontology Tools.

The latest versions of the different tools introduced in Table 1 is given in brackets after each tool's name and a brief sketch of each presented category (cat.) is given in alphabetical order below:

- *Ontology API (Application Programming Interface: OA)* is a set of classes for manipulating ontology information, for example, adding or removing classes, properties, relationships, or individuals. The alignment API is an implementation for expressing and sharing ontology alignments.
- *Ontology Editors (OE)* are applications designed to assist knowledge engineers or domain experts in the creation or manipulation of ontologies. They often express ontologies in one of several ontology languages and propose graphical design environments and interfaces for implementing reasoners. Some of these applications are able to export their output to other languages.

- *Ontology Storage (OS)* can handle a large number of connections. Some prominent systems include RDFStore, Jena, and Sesame (Rohloff et al., 2007). The storage is also an ontology server that is used at design, commit, and runtime. Ontologists use this server to develop ontologies. There are various ways to store ontologies; there are database management systems such as PostgreSQL or MySQL and in-memory or distributed data systems such as Common Object Request Broker Architecture (CORBA). Each system has its own advantages and disadvantages.
- *Topic Map Engine (TM)* provides a comprehensive API to allow programmers to create, modify, and query Topic Map structures.

The necessity for ontology-building, annotating, and integrating storage and learning tools is indisputable. Additionally, human information consumers and Web agents must use and query ontologies and the resources committed to them, creating a further need for ontologies and querying tools. However, the context of storing and querying knowledge has changed due to the wide acceptance and use of the Web as a platform for communicating knowledge. In the past few years, the number of ontology query languages has increased rapidly. Depending on the input data, different query languages are needed. Furthermore, not all ontology tools support all kinds of input data and query languages. Table 2 shows a classification of the most prominent query languages. Currently, efforts are being made to define languages for the Semantic Web. The goal of these languages is to represent Web information so it is understandable and accessible to a machine. In addition, it should also be guaranteed that these languages have enough expressive power to represent the rich semantics of real-world information. They should also be efficient enough to be processed by a machine. All of the languages are based on eXtensible Markup Language (XML). According to Anoniou and Van Harmelen (2008), some of these languages are very successful, such as Resources Description Framework (Schema) (RDF(S)) and Web Ontology Language (OWL); these languages are recommended by the W3C. Table 2 reveals the established query languages according to the underlying data.

Data	Query Lang.	Note	Supported by
XML	*XQuery, XPath, XPointer*	*Query languages for XML data sources.*	*AltovaXML, SAXON*
RDF	*SPARQL*	*SPARQL is a recursive acronym for SPARQL Protocol and RDF Query Language. As implied by its name, SPARQL is a general term for both a protocol and a query language.*	*AllegroGraph Prova, RDFStore, SparqlOwl, etc.*
	RDQL	*Query language for RDF in Jena models.*	*Jena; Sesame*
OWL	*OWL-QL*	*The Joint US/EU ad hoc Agent Markup Language Committee is proposing an OWL query language called OWL-QL.*	*Bossam, FaCT++, Hoolet, KAON2, SHER*
	SQWRL	*Semantic Query-Enhanced Web Rule Language (SQWRL, pronounced "squirrel") is a SWRL-based language for querying OWL ontologies. It provides SQL-like operations to retrieve knowledge from OWL.*	*Protégé-OWL*
SWRL	*SWRL*	*SWRL is a rules-language that combines OWL with Rule Markup Language (RuleML).*	*Bossam, KAON2, Pellet, Protégé-OWL, RacerPro*
	DLP	*Description logic programs (DLPs) are another proposal for integrating rules and OWL. Compared with description logic programs, SWRL takes a diametrically opposed integration approach. DLP is the intersection of Horn logic and OWL, whereas SWRL is (roughly) the union of them.*	*KAON*

Table 2. The Classification of Selected Query Languages.

At this point our focus was mainly to demonstrate the application of RDF triples and accordingly we abstained from trying a Topic Map engine. Instead, for our implementation of the youReputation prototype, we used AllegroGraph's RDFStore, which is a modern, high-performance, persistent RDF graph database. It uses disk-based storage, enabling it to scale to billions of triples while maintaining superior performance. Additionally, it supports SPARQL, RDFS++, and Prolog reasoning from numerous client applications. In a future revision of this prototype, the Topic Map engines should necessarily be evaluated again

After we have presented the creation and management of our fuzzy grassroots ontology, we show in the following section the dashboard for the human-computer interface and its query engine, together called the reputation analysis engine (in Fig. 2, abbreviated as "RAE").

The Reputation Analysis Engine

This system consists of two parts: the dashboard and the query engine. The *dashboard* is a user interface designed so that its text can easily be read; it is the part of the framework communication that operatives interact with. The second and equally important part of the system is the *query engine,* with which automatically presented queries are created after first use. Every interaction initiated by the communication operatives on the dashboard-visualized Topic Map prompts the query engine to provide a new SPARQL query to find the related topics and tags within the ontology storage system. Once the related topics and tags have been located, the query engine also provides the dashboard with the stored and ranked underlying Web sites.

The dashboard is the main visualization of the system (Fig. 4). It provides a Topic Map that conveys information such as topics and the relationships between topics. Topic Maps are standardized and similar to the concepts of mind maps or concept maps in many respects (Pepper, 2010). Topic Maps can also be expressed using XML. However, one difference between the Resource Description Framework (RDF) and a Topic Map is that the latter is centered on topics while the former is centered on resources. The RDF data model is based upon the idea of making statements about Web resources in the form of subject-predicate-objective expressions; known as triples. Topic Maps are not limited to triples, and they represent information using a topic (representing any concept), association (representing hyper-graph relationships between topics), and occurrences (representing information resources). Furthermore, while RDF directly annotates resources, Topic Maps create a semantic network layer—a *virtual map*—above the information resources, leaving the information resources unchanged. Topic Maps explicitly support the concept of identity merging between multiple topics or Topic Maps. Furthermore, because ontologies are Topic Maps themselves, they can also be merged, allowing the automated integration of information from diverse sources into a coherent new Topic Map. The visualization not only shows Topic Maps that were inducted from search results but also more valuable information, such as the different layers (multi-level) that can be viewed by zooming in.

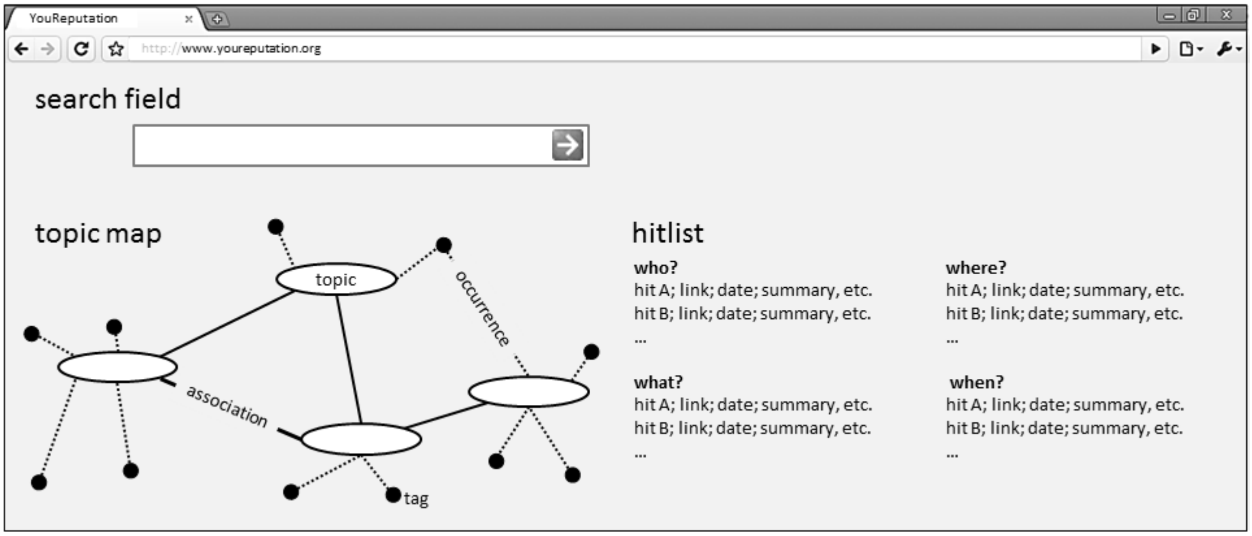

Figure 4. The Appearance of the Interactive Dashboard: the Topic Map (left) and the Hit Lists (right).

The first time the dashboard is used, users do not need to modulate any settings (such as weight K) but only feed the search field box with a name, product, brand, or combination thereof. Instead, the interactive visualization should intuitively lead users to their desired Web sources. Based on the query engine, the dashboard provides a suggested indicator to the user. In other words, a weight K is not set manually but automatically by the framework; however, an operative may change the predicted weight ($K \geq 0.8$) by using the mouse scroll wheel on the Topic Map. Using clicking, zooming in and out, and dragging and dropping, the user can evaluate an entered search term on the Topic Map of the dashboard; i.e., the user can adjust K implicitly. Furthermore, the visual displays hits in different context dimensions, allowing the gaining of further knowledge not only about the entered search term but also about who said what and how influencing this person is.

The Topic Map helps to identify search results by topics that communications operatives can focus on to find exactly what they are looking for or to discover unexpected relationships between items. Tags visualized farther away from a topic belong to it at a less significant level than do the tags that are closer; the same applies to the relationship of the topic itself. Nevertheless, each time the user clicks on the interactive Topic Map, the missing parts of the maps are downloaded from servers and inserted into the dashboard.

A smart representation of the topic-corresponding hits can support further insights. A good way to present hits are Dey and Abowd's (2000) *four w's* (*who, what, where*, and *when*) as the minimal context dimensions to display. Using this method, the different characteristics of social media can help to distinguish these dimensions; some are of greater value in achieving such a distinction, others less. A microblogging service, for instance, is a great tool for finding very recent information; it answers the when question. Social networks could become a tool for information on who, and wikis may be used to answer who, what, when, and where. As a tool for discussion, blogs might not just answer the questions of who, what, when, and where; they could also be used as an information tool to learn about the background of an issue and to analyze why (Hächler, 2010). Splitting hits, with respect to their origin, into context dimensions allows an intuitive interaction with a different kind of social media.

The query engine is an introduction into how a user-provided query can be enriched using ontology. However, to query ontology data, we need to use a query language. As more data is being stored in RDF formats— friend of a friend (FOAF) and really simple syndication (RSS) are two examples that can rest on it—a need has arisen for a simple way to locate specific information. SPARQL, as a powerful query language, fills that niche, making it easy to find data in RDF (Prud'hommeaux & Seaborne, 2008). Since SPARQL is an OWL query language, it appears to perfectly fit our needs for querying the RDF ontology data that is already stored in the knowledge database. Whenever the system receives search input from users, the query engine performs semantic queries to find the terms that are near to the input term regarding the semantic. For example, the SPARQL query below demonstrates the selection of relevant terms in relation to the search for "*bow*" (Qry. 1).

```
PREFIX dc:<http://purl.org/dc/elements/1.1/>
PREFIX ns:<http://example.org/ns#/>
SELECT ?relevant
WHERE { ?x ns:relevant ?relevant.
FILTER ?distance < 8 .
?x dc:term ?term .
?term = "bow"}
```

Query 1. Selecting Relevant Terms.

SPARQL will return the terms that are at a distance closer than 8 (the preset weight for K) to the search term *"bow."* As indicated by this simple example, even though it looks like an SQL query, the SPARQL query contains the semantic meaning. Depending on the proximities of the Web-agent-collected terms in the fuzzy grassroots ontology, this search will envisage, for example, the topics "*ship*" and "*weapon*" on the Topic Map.

To provide the readers not only with an abstract framework but also with an easy-to-use tool, the next subchapter showcases the youReputation prototype, a free Web-based reputation analysis tool.

YOUREPUTATION: A REPUTATION ANALYSIS TOOL

For a short demonstration of the foRa framework features, our youReputation prototype scanned data from the social bookmarking service Delicious (www.delicious.com) and the microblogging platform Twitter (www.twitter.com). The prototype provided the results regarding the input, computed a reputation with the relevant information, and converted the valuable information into an ontology. The goal of youReputation is to provide a reputation result based on the search input. The results include relevant terms and links to Web sites that correspond to a term that the user wants to evaluate. The data crawled from Delicious and Twitter was processed, clustered, and converted to an ontology. Figure 5 indicates the kernel of the youReputation prototype.

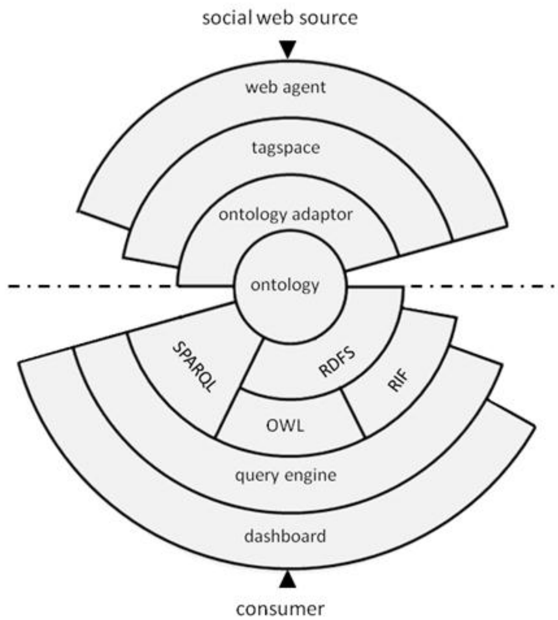

Figure 5. The youReputation Prototype Kernel.

The building blocks of the prototype kernels are briefly explained below. Viewed from above, the first part of the prototype consists of the following components: a Web agent, a tagspace, an ontology adaptor, and the ontology itself stored in RDF format. The system operates as follows. A Web agent travels across the Social Web to collect information. Heterogeneous data must be adjusted before storing. Noise data (i.e., advertisements, banners, footers, etc.) are removed. The remaining data (i.e., tags from Delicious or Twitter) are normalized and stored. Then, the ontology adaptor converts the raw data from the Web agents into an ontology. After conversion, the ontologies can be queried using SPARQL, which runs the queries on the RDF/OWL data.

Seen from below, the second part of the prototype consists of the following components: a dashboard; a query engine; the Semantic-Web-stack-based components SPARQL, OWL, RDFS, Rule Interchange Format (RIF); and an RDF-based semantic ontology. Although originally envisioned as a rules layer for the Semantic Web, in reality, the design of RIF is based on the observation that there are many rules languages in existence and what is needed is to exchange rules between them. RIF is not used by the youReputation prototype, but it could be integrated to personalize the ranking hits; in a possible later extension of youReputation to a system with login functionality, it would be feasible to analyze user behavior and accordingly favor the ranking of found sources. RDFS is an extensible knowledge representation language intended to structure RDF resources and provide the basic elements for the description of ontologies. Many RDFS components are included in OWL, a family of knowledge representation languages for authoring ontologies. The language is characterized by formal semantics and by RDF/XML-based serializations for the Semantic Web and has especially attracted academic, medical, and commercial interest. Through SPARQL, the semantic ontology can be grasped. It was standardized by the RDF data access working group of the W3C and later became an official recommendation. The query engine supports the query process using SPARQL.

Whenever the user inputs search text, the query engine attempts to find relevant terms by querying the ontology base. The dashboard as a visualization tool is responsible for showing these results as a Topic Map that indicates the relationship between tags and their information. Then, the system retrieves the stored URL related to the tag to provide the relevant information. Retrieved hits are also presented on the dashboard according to the context dimensions.

To explain the benefits of the youReputation prototype, let's introduce a small example. The problem with new and previously unobserved information on the Web is that the relationship between terms and topics is not precisely known. Thus, we assume that a communication operative at Apple Inc., a multinational corporation that designs and markets computer electronics, is asked to analyze the corporation's online reputation, so Apple's communication operative decides to give our prototype a chance and, hence, makes use of the youReputation prototype. For that purpose, the communication operative enters the search term "*Apple*" in the search field box on the start page. On the left side of the dashboard, immediately, the visualized ontology (meaning all Apple-related topics and terms) appears. Moreover, on the right side of the dashboard, a conventional hit list also appears. In Figure 6, a snapshot of the dashboard's Apple search of the youReputation prototype is presented.

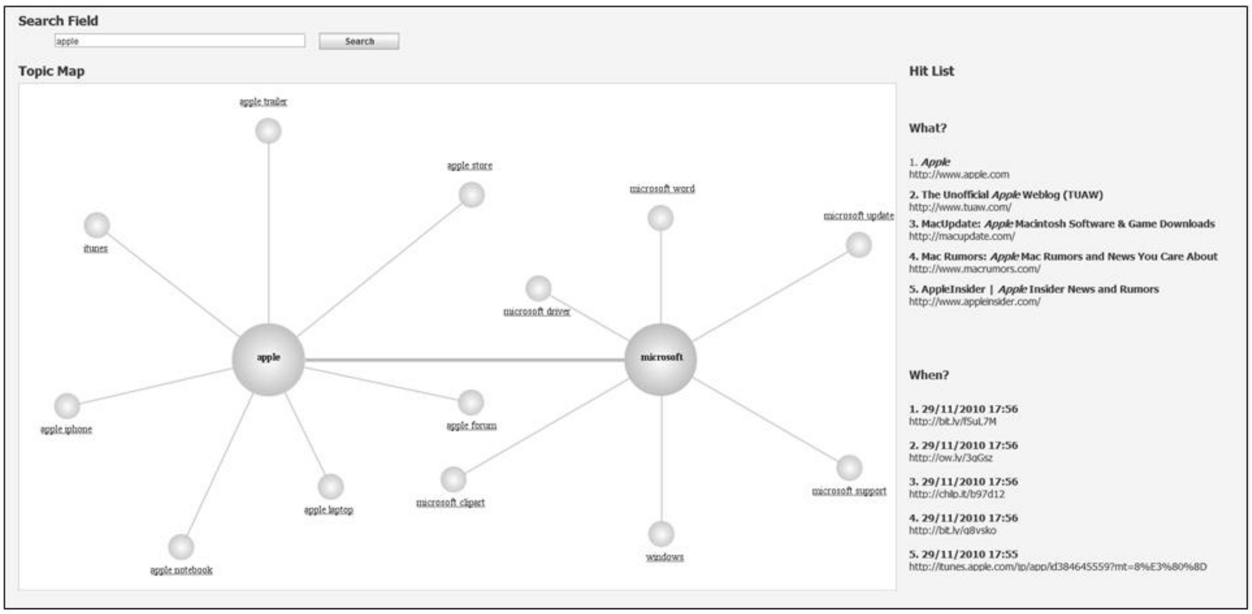

Figure 6. A Snapshot of the youReputation Prototype Dashboard.

As demonstrated by Figure 6, the hit list, in reality, is not very conventional; it is partitioned into the presented four context dimensions. To illustrate, the hits for the context dimension "what" are coming from Delicious. However, for the context dimension "when," the hits are coming from Twitter because Twitter provides additional data such as the date and time of a tweet. Thus, it is possible to find past as well as real-time information depending on the Web agents' frequency of updating the crawl frontier list and, in doing so, also the knowledge database.

The consequence for Apple's communication operative is that he not only finds more information concerning his entered search term "*Apple*" but also receives more structured information (based on the context dimension partitions). With a conventional Boolean search, he would find only information containing the term "*Apple*". In contrast, youReputation enables him to find not only the search term but also, depending on the fuzzy membership degree, more or less-related topics and terms, presented as an appealing interactive Topic Map. Topic Maps are generated based on the fuzzy grassroots ontology that is the tagspace clustered with FCM and stored in AllegroGraph's RDFStore. The different distances from the topic to the terms arise from the different membership degrees of each term to the related topic. Since this Topic Map is interactive, Apple's communication operatives can zoom in and out and browse the ontology in a straightforward way. Based on the Topic Map, in Figure 6, it is also noted that, after a search for

"*Apple*," the business competitors "*Microsoft*" and its related terms (e.g. "*Microsoft Word*," "*Microsoft update*," etc.) also can be found. By clicking on the topic "*Apple*," all terms (e.g. "*Apple Store*," "*Apple Forum*," etc.) are included in the search and presented in the hit list, whereas, by clicking on a single term (e.g. "*iPhone*"), only this search will be presented in the hit list. The fragmentation of the hit list according to the context dimensions helps to better structure his located information.

Having presented our foRa framework, including the associated youReputation prototype, the next subchapter discusses universal future research directions for the Social Semantic Web.

FUTURE RESEARCH DIRECTIONS

There have been several attempts to help not only online reputation analysis but also all kinds of WSEs to overcome keyword search. Through recent evolutions in Social Web technologies resulting in easy-to-use tools, users continually bustle in today's Web and create a quickly growing volume of data (O'Reilly & Battelle, 2009). This situation implies that the productivity of WSEs should increase. With tagging, an early limitation of current WSEs is redressed because people can label and find content themselves. Current reputation analysis relies heavily on customer surveys, although online reputation analysis tools, such as Actionly (www.actionly.com), BackType (www.backtype.com), Engagor (www.engagor.com), Radian6 (www.radian6.com), and ReputationDefender (www.reputationdefender.com), try to provide insights into a firm's Social Web prestige in an interactive style. Because most communications operatives are not familiar with the optimal wording of search queries (which requires the use of SPARQL), new search forms should be developed.

Solutions to overcome these limitations include the so-called QA (question-answering) systems (Zadeh, 2006). For instance, Wolfram Alpha's system (www.wolframalpha.com) is capable of handling user requests concerning measurements, but it is not able to understand the semantics behind user queries. The engine achieves nothing beyond the tasks at which machines are known to excel, manipulating numbers and symbols. A promising way to overcome these limitations is the *computing with words (CWW)* paradigm (Zadeh, 1996). Formulated in 1996 as a methodology in which the objects of computation are words and propositions drawn from natural language, such as, for example, *"high tree"* or *"a tree has many boughs."* Unlike a classical logical approach, CWW provides a much more expressive language for knowledge representation and, consequently, for reasoning. Because fuzziness is ubiquitous and essential for humans, CWW offers a new perspective for improving human-machine interactions (Zadeh, 2004). According to Hagras (2010), CWW relates to developing intelligent systems that are able to receive perceptions and propositions drawn from natural language as input words and then produce a decision or output based on these words. According to Guadarrama (2010), CWW can be used in knowledge representation, learning, and programming. Our interactive visualization of the ontology as Topic Maps is a first effort to represent world knowledge (Craven et al., 2000).

It is a long way toward an *intelligent Web*, as Spivack (2009) outlined. A first step in this direction will be to integrate natural language into the Web to achieve a real Semantic Web (Zadeh, 2009; Zadeh, 2010). The CWW paradigm can be used for this task. Thus, future communications operatives will be able to use natural language to search for online reputation information. In an intelligent Web, it will be possible to ask questions and receive context-dependent answers. There are several emerging technologies meant to overcome further challenges toward a true intelligent Web, such as the *Web of things* model (interconnection of all types of devices through Web standards), *machine translation* (the translation of text or speech from one natural language to another), *machine vision* (the recognition of objects in an image and the ability to assign properties to those objects to make them machine-readable), *structured storage* (which does not require fixed table schemas), and *quantum computing* (which will be able to solve certain types of problems much faster than any current computers).

In conclusion, the next subchapter discusses once more the overall coverage of this chapter and closes with concluding remarks.

CONCLUSION

The proposed approach attempts to establish a knowledge representation for reputation analysis through Topic Maps (which are a standard for the representation and exchange of knowledge) with an emphasis on the findability of information. So far, a few WSEs have relied on clustering content into different classes, but none of them have been able to understand or process intuitive and human-oriented Web queries based on linguistic terms or expressions. Visualized interactive Topic Maps can help find similar linguistic terms clustered around a topic. The interactive visualization of these maps is a first step that was established solely to augment user understanding, but, in the future, maps underlying semantic ontologies ought to be enhanced to knowledge bases to allow the machines to reason using the user's vague natural language expressions. Based on this knowledge base, a future WSE that is able to understand natural language queries could be established.

This chapter provided a foundation for further analysis of a reputation system, the foRa framework, and the youReputation prototype. We studied the development and analysis of reputation systems and developed methodologies to address some of the fundamental definitions, such as the Semantic Web, ontologies, and fuzzy clustering algorithms. Our prototype was designed to be simple and easily extensible. The introduced framework is an approach for communications operatives to gain deeper insights into the reputation of a company in online media. Because the boundaries in fuzzy classification systems are not rigorous, it is possible to find more and higher-quality results. As revealed here, the found hits can be presented in an understandable way using an appropriate form of splitting into context dimensions according to their origin. This system allows communications operatives to successfully interact with different kinds of social media. Among other things, the prototype is intended to illustrate the possibilities provided by the vast amount of recent Social Web data. Therefore, it simulates only a few aspects of the foRa framework. Developing the youReputation prototype according to the framework with a prototyping method has the benefit of allowing us to obtain feedback from users during the development process.

Strictly following the methods of prototyping and to increase comprehension, we always used the simplest formulas and algorithms to highlight our ideas in this chapter. Furthermore, we believe that this system represents a good starting point to develop other kinds of social semantic software. However, we will continue to experiment with variations of more advanced formulas and algorithms. For example, we will evaluate whether there are potentially superior measures to the Jaccard similarity measurement. Further experimental tests include comparisons with commonly used non-metric measurements: the Dice, the Kulczynski, the Russel and Rao, Simple Matching, and the Tanimoto coefficient. Additionally, Kleinberg's HITS algorithm will be tested against other comparable algorithms: Google's PageRank, Marchiori's Hyper Search, and Yahoo's TrustRank. Using a variety of Soundex algorithm derivatives, such as Daitch–Mokotoff, fuzzy Soundex, Metaphone and Double Metaphone, New York State Identification and Intelligence System (NYSIIS), and the Reverse Soundex algorithm, we expect further improvements of our prototype. Although we focused here on the English language, semantic ontologies for other languages could be established through the same methods with adapted language-relevant algorithms (e.g., the "Kölner phonetic" for the German language). Another essential evaluation will be to weigh the FCM algorithm against other comparable fuzzy clustering algorithms, such as fuzzy self-organizing maps (FSOMs), fuzzy clustering by local approximation of memberships (FLAME), Gath-Geva, and Gustafson-Kessel.

Another important point for further reputation analysis should be the analysis of word-inherent positive or negative connotations to automatically categorize found Web sources; this objective could, potentially, be based on fuzzy techniques. With further involvement of the previously discussed context dimensions, the influence of different Social Web prosumers can be elicited. In addition, the patterns of interaction with our dashboard displayed by communications operatives can be used as a springboard to affect the future ranking of documents, producing a more personalized outcome.

REFERENCES

Agosti, M. (2007). *Information Access through Search Engines and Digital Libraries*. Berlin: Springer.
Anoniou, G. and Van Harmelen, F. (2008). *A Semantic Web Primer*. Cambridge: MIT Press.
Baeza-Yates, R, and Ribeiro-Neto, B. (2011). *Modern Information Retrieval: The Concepts and Technology Behind Search*. New York: Addison-Wesley Educational Publishers.
Bezdek, J.C. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum Press.
Bezdek, J.C., Keller, J., Krisnapuram, R., Pal, N. R. (2008). Fuzzy Models and Algorithms for Pattern Recognition and Image Processing. New York: Springer
Bourke, P. (1997). Intersection of two circles. Retrieved November 30, 2010, from http://local.wasp.uwa.edu.au/~pbourke/geometry/2circle/.
Blumauer, A. and Pellegrini, T. (2009). Social Semantic Web: Web 2.0 - Was nun? Berlin: Springer.
Breslin, J.G., Passant, A., Decker, St. (2009). The Social Semantic Web. Berlin: Springer.
Bruhn, M. (2002). *Relationship Marketing: Management of Customer Relations*. Essex: Financial Times.
Cardoso, J. (2007). The Semantic Web Vision, Where Are We? *IEEE Computer Society*, 22(5) 22-26.
Chun, R. (2005). Corporate reputation: Meaning and measurement. *International Journal of Management Review*, 7(2) 91-109.
Craven, M., DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, S. (2000). Learning to construct knowledge bases from the World Wide Web. *Artificial Intelligence*, 118 (1–2), 69–113.
Dey, A.K. and Abowd, G.D. (2000, April). *Towards a Better Understanding of Context and Context-Awareness*. Paper presented at the CHI 2000 Workshop on the What, Who, Where, When, Why and How of Context-Awareness, Hague, Netherlands.
Ebert, T. (2009). *Trust as the Key to Loyalty in Business-to-Consumer Exchanges: Trust Building Measures in the Banking Industry*. Wiesbaden: Gabler.
Eccles, R.G., Newquist, S.C., Schatz, R. (2007). Reputation and Its Risks. *Harvard Business Review*, 85(2), 104-14.
Eisenegger, M. and Imhof, K. (2007). Das Wahre, das Gute und das Schöne: Reputations-Management in der Mediengesellschaft. *Fög discussion paper 2007-0001*. Retrieved November 30, 2010, from http://www.foeg.uzh.ch/staging/userfiles/file/Deutsch/f%C3%B6g%20discussion%20papers/2007-0001_Wahr_Gut_Schoen_2007_d.pdf
Fombrun, Ch. J. and Wiedmann, K. P. (2001). Reputation Quotient. Analyse und Gestaltung der Unternehmensreputation auf der Basis fundierter Erkenntnisse. *Schriftenreihe Marketing Management*, 1-52.
Gaines-Ross, L. (2008). *Corporate Reputation: 12 Steps to Safeguarding and Recovering Reputation*. Hoboken: John Wiley & Sons.
Gavrilis, C., Kakali, C., Papatheodoro, C. (2008). Enhancing Library Services with Web 2.0 Functionalities. In: B. Christensen-Dalsgaard et al. (Eds.), *ECDL 2008*, LNCS 5173, 148–159.
Gruber, T. R. (1993): A Translation Approach to Portable Ontology Specifications. Knowledge Systems Laboratory, *Technical Report KSL*, 92(71), 199-220.
Guadarrama, S. (2010). Guadarrama on CWW. In J. Mendel (Ed.), *What Computing with Words Means to Me* (pp. 24-25). IEEE Intelligence Magazine .
Gunelius, S. (2010). *Blogging All-in-One For Dummies*. Indianapolis: John Wiley & Sons.
Hächler, L. (2010). *Web 2.0 and 3.0: How Online Journalists Find Relevant and Credible Information*. Unpublished master thesis, University of Fribourg, Fribourg, Switzerland.
Hagras, H. (2010). Hagras on CWW. In J. Mendel (Ed.), *What Computing with Words Means to Me* (pp. 24-25). IEEE Intelligence Magazine .
Hasan-Montero, Y. and Herrero-Solana, V. (2006): Improving Tag-Clouds as a Visual Information Retrieval Interfaces. *Proceedings of International Conference on Multidisciplinary Information Sciences and Technologies*.

Ingenhoff, D. (2004). *Corporate Issues Management in multinationalen Unternehmen: Eine empirische Studie zu organisationalen Strukturen und Prozessen*. Wiesbaden: VS Verlag für Sozialwissenschaften.
Ingenhoff, D. and Sommer, K. (2008). The interrelationships between corporate reputation, trust and behavioral intentions. A multi-stakeholder approach, *58th Annual Conference of the International Communication Association (ICA),* 21.-27, Montreal, Canada
Kaser, O. and Lemire, D. (2007). Tag-Cloud Drawing: Algorithms for Cloud Visualization. *Electronic Edition*, Banff, 2007.
Kleinberg, J. (1998). Authoritative sources in a hyperlinked environment. In: *9th ACM-SIAM Symposium on Discrete Algorithms,* 1-33, Odense, Denmark.
Klewes, J., Wreschniok, R. (2009). *Reputation Capital: Building and Maintaining Trust in the 21st Century*. Berlin: Springer.
Lewandowski, D. (2005). *Web Information Retrieval: Techniken zur Informationssuche im Internet*. Düsseldorf: Deutsche Gesellschaft f. Informationswissenschaft u. Informationspraxis.
Manning, Ch. D., Raghavan, P., Schütze, H. (2008). *Introduction to Information Retrieval*. New York: Cambridge University Press.
McLuhan, M. and Nevitt B. (1972). *Take Today: the Executive as Dropout*. New York.
Meier A. and Stormer H. (2009). *eBusiness & eCommerce: Managing the Digital Value Chain*. Berlin: Springer.
Oliveira, J.V., Pedrycz, W. (2007). Advances in Fuzzy Clustering and its Applications. West Sussex: John Wiley & Sons.
O'Reilly, T. (2005). What Is Web 2.0. Design Patterns and Business Models for the Next Generation of Software. Retrieved November 30, 2010, from http://oreilly.com/web2/archive/what-is-web-20.html
O'Reilly, T. and Battelle, J. (2009). Web Squared: Web 2.0 Five Years On, *Special Report*. Retrieved November 30, 2010, from http://www.web2summit.com/web2009/public/schedule/detail/10194
Pepper, S. (2010). Topic Maps. *Encyclopedia of Library and Information Sciences*. Retrieved November 30, 2010, from http://www.google.ch/url?sa=t&source=web&cd=1&sqi=2&ved=0CCMQFjAA&url=http%3A%2F%2Fwww.ontopedia.net%2Fpepper%2Fpapers%2FELIS-TopicMaps.pdf&rct=j&q=pepper%20topic%20maps&ei=aF7uTKjoHYiSswavzpT-Cg&usg=AFQjCNFolzoDB1u5NNgkRRbRi9itkEaJnA&sig2=cmLGL3R5x6CubvBQfKmnmA&cad=rja
Peters, I. (2009). *Folksonomies: Indexing and Retrieval in the Web 2.0*. Berlin: Saur.
Phillips, D. and Young, P. (2009). *Online Public Relations: A practical guide to developing an online strategy in the world of social media*. London: Kogan Page Limited.
Picot, A., Reichwald, R., Wigand, R. T. (2003). *Die grenzenlose Unternehmung: Information, Organisation und Management*. Wiesbaden: Gabler Verlag.
Portmann, E. (2008). *Informationsextraktion aus Weblogs*. Saarbrücken: VDM.
Portmann, E. and Meier, A. (2010). A Fuzzy Grassroots Ontology for improving Weblog Extraction. *Journal of Digital Information Management*, 8(4), 276-284.
Prud'hommeaux, E. and Seaborne, A. (2008). SPARQL Query Language for RDF. *W3C Recommendation 15 January 2008*. Retrieved November 30, 2010, from http://www.w3.org/TR/rdf-sparql-query/
Röttger, U. (2005). Kommunikationsmanagement in der Dualität von Struktur. *Medienwissenschaft Schweiz/Science des mass média suisse*, 1(2), 12-19.
Rohloff, K., Dean, M., Emmons, I., Ryder, D., Summer, J. (2007). An evaluation of triple-store technologies for large data stores. *Proceedings of the 2007 OTM Confederated international conference on the move to meaningful internet systems*.
Russel, R. C. (1918). US Patent 1261167, 1918.
Russel, R. C. (1922). US Patent 1435663, 1922.
Scott, D. M. (2010). The *New Rules of Marketing and PR: How to Use Social Media, Blogs, News Releases, Online Video, and Viral Marketing to Reach Buyers Directly*. New Jersey: John Wiley & Sons.
Setsuo, A., Suzuki, E. (2008). Discovery Science: 7th International Conference, DS 2004, Padova, Italy, October 2-5, 2004. Berlin: Springer.

Sirmakessis, Sp. (2005). Knowledge Mining: Proceedings of the NEMIS 2004 Final Conference. Berlin: Springer.
Smith, G. (2008). Tagging. People-powered Metadata for the Social Web. Berkeley: New Riders.
Spivack, N. (2009). The Evolution of the Web: Past, Present, Future. Retrieved November 30, 2010, from http://www.novaspivack.com/uncategorized/the-evolution-of-the-web-past-present-future
Van Harmelen, F., Lifschitz, V., Porter, B. (2007). *Handbook of Knowledge Representation*. New York: Elsevier.
Troncy, R., Huet, B., Schenk, S. (2011). Multimedia Semantics: Metadata, Analysis and Interaction. New Jersey: John Wiley & Sons.
Van Riel, C.B.M. and Fombrun, C. (2007). *Essentials of corporate communication: implementing practices for effective reputation management*. Abingdon: Routledge.
Voss, J. (2007). Tagging, Folksonomy & Co - Renaissance of Manual Indexing? *Proceedings of the International Symposium of Information Science*, 234–254.
Ward, M., Grinstein, G. G., Keim D. (2010). *Interactive Data Visualization: Foundations, Techniques, and Applications*. Natick: Transatlantic Publishers.
Weller, K. (2010). *Knowledge Representation in the Social Semantic Web*. Berlin/New York: de Gruyter Saur.
Werro, N. (2008). Fuzzy Classification of Online Customers. Retrieved November 30, 2010, from http://ethesis.unifr.ch/theses/downloads.php?file=WerroN.pdf
Zadeh, L.A. (1996). Fuzzy Logic = Computing with Words. *IEEE Transactions on Fuzzy Systems*, 4(2), 103-111.
Zadeh, L.A. (1999). From Computing with Numbers to Computing with Words – from Manipulation of Measurements to Manipulation of Perceptions. *IEEE Transactions on Circuits and Systems: Fundamental Theory and Applications,* 45(1), 105-119.
Zadeh, L.A. (2004). A note on web intelligence, world knowledge and fuzzy logic. *Data Knowledge Engineering*, 50, 291–304.
Zadeh, L.A. (2006). From Search Engines to Question Answering Systems – The Problems of World Knowledge, Relevance, Deduction and Precisiation. In: Elie Sanchez (Ed.), *Fuzzy Logic and the Semantic Web*, Elsevier, 163–210.
Zadeh. L.A. (2009). Toward extended fuzzy logic—A first step. *Fuzzy Sets and Systems*, 160(21), 3175-3181.
Zadeh, L.A. (2010, July). *Precisiation of Meaning: Toward Computation with Natural Language*. Presented at the IEEE 2010 Summer School on Semantic Computing, Berkeley, CA.
Zudilova-Seinstra, E., Adriaansen, T., van Liere, R. (2008). *Trends in Interactive Visualization: State-of-the-Art Survey*. Berlin: Springer.

KEY TERMS & DEFINITIONS

Folksonomy: is a system of classification derived from the practice and method of collaboratively creating and managing tags to annotate and categorize content; this practice is also known as collaborative tagging, social classification, social indexing, and social tagging.

Ontology: provide criteria for distinguishing various types of objects (e.g. concrete and abstract, existent and non-existent, real and ideal, independent and dependent) and their ties (relations, dependences and predication). Within computer science the term stands for a design model for specifying the world that consists of a set of types, relationships and properties.

Reputation Analysis: is a reputation management task conducted by communications operatives and consist of the process of tracking an entity's actions and other entities' opinions about those actions; reporting on those actions and opinions; and reacting to that report creating a feedback loop.

Semantic Web: describes methods and technologies to allow machines to understand the meaning—or semantics—of information on the Web. According to the original vision, the availability of machine-readable metadata would enable automated agents and other software to access the Web more intelligently.

Search Engine: is an instrument intended to search for information on the Web. The search results are typically presented in a single list and are generally called hits. The information can consist of images, text, Web pages, and auxiliary types of documents.

Social Semantic Web: subsumes developments in which social interactions on the Web lead to the creation of explicit and semantically rich knowledge representations. The Social Semantic Web can be seen as a Web of collective knowledge systems, which are able to provide useful information based on human contributions and which get better as more people participate. The Social Semantic Web combines technologies, strategies and methodologies from the Social and the Semantic Web.

Social Web: describes how people socialize or interact with each other throughout the Web as a medium with easy-to-use software.

Topic Map: is a standardized format of representation and interchange of knowledge, with an emphasis on the findability of information.

Interactive Visualization: is a branch of graphic visualization in computer science that involves studying how humans interact with computers to create graphic representations of information.