A Fuzzy Grassroots Ontology for Improving Social Semantic Web Search

Edy Portmann

Information Systems Research Group
Department of Informatics
Boulevard de Pérolles 90
CH-1700 Fribourg
edy.portmann@unifr.ch

Summary

The web is continuously evolving into a collection of many data, which results in the interest to collect and merge these data in a meaningful way. Based on that web data, this paper describes the building of an ontology resting on fuzzy clustering techniques. Through continual harvesting folksonomies by web agents, an entire automatic fuzzy grassroots ontology is built. This self-updating ontology can then be used for several practical applications in fields such as web structuring, web searching and web knowledge visualization. A potential application for online reputation analysis, added value and possible future studies are discussed in the conclusion.

Keywords: Folksonomy, Fuzzy Data Clustering, Knowledge Visualization, Ontology, Search Engine, Semantic Web, Social Semantic Web, Social Web.

1 Motivation and Related Work

In web search, it is vital to know how to retrieve relevant content. Due to noisy data descriptions in web systems, it can be much more difficult to find relevant content as with conventional search systems. However, defining relevance is a challenge. For example, does retrieved data resulting from a keyword search exactly match the semantic content of the given search? The great problem with the resulting data is that it often yields findings which are only partially relevant to their semantics. As a consequence, the matching of a document to the query terms is often vague.

This paper discusses an approach to find more reasonable web content using fuzzy logic. The core power of fuzzy logic is the fuzzy set theory, first proposed by Zadeh as an extension of the traditional set theory [22]. Meier et al. describe fuzzy logic as a suitable instrument for roughly modeling the kind of uncertainty related to vagueness [12]. Fuzzy logic is an addition to conservative logic and handles the concept of partial truth along with true and false, which is used for qualitative rather than quantitative judgment. Hence fuzzy logic follows the way humans think and helps to handle real world complexities more efficiently.

In this way, folksonomies will be converted to machine-understandable ontologies. To this end, this paper shows that fuzzy sets can overcome the gap between the bottom-up approach of folksonomies and the top-down approach of ontologies. Nevertheless, the obtained ontology can afterwards help to recognize search results by topic, not only in the same domain. In addition, through an interactive visualization of parts of the ontology in a graphical user interface (GUI) the user can interact with search results in a straightforward manner. So it is possible during a search process to zoom in on precisely what was searched for or expose unforeseen relationships among items.

2 Concept and Components

2.1 From Social to Social Semantic Web

The term social web is used to describe how user socialize or interact with each other through the web. Its aim is to ease communication, promote safe information sharing, and enable interoperability and collaboration. This has led to the expansion and development of web-based communities, hosted services, and applications such as social software. A major reason for the overnight success of social web is its simplicity of use. These sites not only afford data but also generate a plethora of weakly arranged metadata.

At present, the social semantic web amends the bottom-up wisdom-of-the-crowds attempt of social web in a top-down manner [4]. Its fundamental aim is a stronger knowledge representation as it is possible in the social web. It is made of interlinked documents, data, and even applications created by the users themselves as a consequence of all kinds of social interactions, and it is depending on computer-readable formats from semantic web, so that it can be used for purposes that the actual state of the social web cannot accomplish without difficulty.
2.2 Folksonomy

Typical examples of the aforementioned social software are folksonomies. Folksonomy labels the practice and technique of collaboratively creating and manipulating tags to annotate and categorize content through users. By this means a document, a uniform resource locator (URL), a picture or a movie can be marked as content.

In folksonomies, freely chosen tags are used instead of a controlled vocabulary [18]. Such metadata are straightforward to create, but generally lack any kind of formal base, as used in the semantic web. In this sense folksonomies will be used as a starting point to harvest social knowledge from these users and to compile a fuzzy grassroots ontology for the social semantic web.

2.3 Fuzzy Data Clustering

The method of separating data elements into clusters or classes, so that objects in the same cluster are as identical as possible, and objects in different clusters are as diverse as possible, is called data clustering. Depending on the intention for which clustering is being used and the nature of the data, special measures of relationship may be used to place items into clusters, where the relationship measure controls how the clusters are shaped. Connectivity, distance, and intensity are examples of measurements that can be used for clustering [1, 6, 19]

In hard clustering, data is separated into distinctive clusters, where all data elements belong to precisely one single cluster. In fuzzy clustering, data elements can belong to more than one cluster, and each element is associated with a set of membership levels [19]. These indicate the potency of the relationship between that data element and a particular cluster. Fuzzy clustering is a method of assigning these diverse levels of membership and allocating data elements to one or more clusters according to the membership values.

2.4 Ontology

Within computer science, the term ontology stands for a design model for specifying the world that consists of a set of types, relationships and properties. What precisely is provided can vary, but these characteristics are fundamental to every ontology [7]. There is an expectation that the model bear a resemblance to the real world as well. However, it will clearly 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. It is customary for an ontology to refer only to the ontology model. An ontology, hence, is merely a specification of conceptualization without naming instances [7]. If instances are annotated by ontology tags and modeled as ontology, then we speak of a knowledge base. Thus, a knowledge base is a repository of instances of the concepts defined in the ontology, and the ontology provides the knowledge pattern stored in the knowledge base.

2.5 Knowledge Representation and Visualization

Visualization techniques empower people to spot patterns in (web) data, identify areas that need additional analysis and make sophisticated decisions based on these patterns [23]. 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.

An interesting feature of visualization 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. 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 [21].

3 A Fuzzy Grassroots Ontology

3.1 Building Blocks

This section focuses primarily on the creation of the fuzzy grassroots ontology, but secondly also on their utilization. The creation compromises four steps:

1. In the first step, web agents are constantly crawling the social web, looking for folksonomies (tags) and the underlying websites (source).
2. During the second step, the found tags are normalized and the underlying sources are ranked.
3. In the third step, a tagspace is created. The previously collected and normalized tags are linked to each other. After this step, all of the tags are linked to each other and plotted onto a tagspace.
4. The fourth step is the ontology adaption, which separates the plotted tagspace into hierarchies of classes. To build an ontology, the tagspace is clustered with random initialization using a fuzzy data clustering algorithm.

Note that for the creation of the fuzzy grassroots ontology, the “lex parsimoniae” is followed. This law of parsimony suggests tending towards simpler solutions until some simplicity for increased explanatory power can be traded.

3.2 Crawling the social web

At the beginning web agents identify, by constantly crawling the web, all tags and the underlying sources and subjoin them into lists. These collected tags from folksonomies are needed to establish the fuzzy grassroots ontology. However, for the underlying knowledge base, 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 the Hyperlink-Induced Topic Search (HITS) algorithm [11].

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 and authority weights for each document until they converge. Essentially, the hubs and authorities are the documents with the biggest values in the main eigenvectors of $A^T A$ and $A A^T$, respectively. HITS is used to rank all of the sources of the knowledge base in combination with their identified tags in the fuzzy grassroots ontology according to their relevance.

### 3.3 Normalizing the collected tags

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.

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 [16, 17]. Algorithm 1 illustrates the method of the Soundex algorithm that is in this used for the normalization of the tags. 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.

A major advantage of the utilization of this 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 a dashboard for example.

### 3.4 Creating the tagspace

However, after all of the tags have been collected and normalized, they need to be sorted. Because the web agents are constantly crawling through the web, this sorting process must be periodically repeated. The tagspace is a two-dimensional representation of a consistent picture and serves as the input for the ontology adaption. 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 [1]:

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

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 $k$ ($j = 1, 2, ..., n$), and $p (\geq 1)$ the Minkowski constant. The critical factor in this equation is to obtain the constant $p$, which defines the Minkowski metric. A simple Minkowski metric-based coefficient that can be used to measure the semantic correlation between tags is the Jaccard similarity coefficient $d_I(A, B)$. Let $A$ and $B$ be the sets of resources characterized by two tags. Relative co-occurrence is ascertained with the following formula:

$$d_I(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{2}$$

In other words, 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 certain depth. Child point locations are computed based on an algorithm, which calculates the intersection of two or three circles [3].
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, the necessary points in the selected region can easily be shown, which is very effective for supporting a zoom function. Another parameter to take into account is the constant variability of the underlying data. Normally 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 acute. As a result, the plotting algorithm (Algo. 2) is able to provide a good perspective on moving data.

3.5 Adapting the ontology

The ontology adaption can be described as follows: all n tags plotted in the tagspace will be sorted by the fuzzy c-means (FCM) algorithm [2]. This algorithm attempts to split a limited collection of elements X = \{x_1, ..., 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, the following “rule of thumb” is used:

\[ c \approx \sqrt{n/2} \]

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 modified FCM algorithm (Algo. 3) is, as the original, based on the minimization of an objective function:

\[ J(u,v) = \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik}^{m} d^2(v_i, x_k) \]

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.

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 \( \varepsilon \), a given sensitivity boundary value):
   a. Calculate the centroid for each class, using Formula (5) below.
   b. For each point, compute its coefficients within the class, using Formula (7) below.
4. Reiterate until the algorithm has converged (that is, the adjustment of the coefficients between two iterations is no more than \( \varepsilon \), a given sensitivity boundary value):
5. Concatenate all of the same terms together.

Algorithm 3: Modified Fuzzy C-Means Algorithm.

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:

\[ v_i = \frac{\sum_{x} u_{ik}(x)^{m} x}{\sum_{x} u_{ik}(x)^{m}} \]  

(5)

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

\[ u_{ik}(x) = \frac{1}{d(v_i, x_k)} \]  

(6)

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:

\[ u_{ik}(x) = \frac{1}{\sum_{i}(d(v_i, x_k)^{2/(m-1)})} \]  

(7)
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.

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 1 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 a semantic web ontology storage system (e.g. AllegroGraph’s RDFStore).

![Figure 1: Schematic representation of the ontology building process.](image)

However, the hierarchy of all of the classes is stored using an ontology tool, so we obtain several hierarchies that are jointly called fuzzy grassroots ontology. The ranked websites (source) that belong to the single tags are ranked and stored separately but linked to the ontology to establish a knowledge base.

### 3.6 Applying the fuzzy grassroots ontology

The generated ontology can be used in many ways. Because the approach converts human concepts into mathematical models, it is valuable as a basis for (social) semantic web applications that need a structured foundation. The integration of an automatically built grassroots ontology into the microblogging platform Twitter, for example, could lead to a better search for related topics. Using the fuzzy grassroots ontology could unveil unforeseen relationships.

Another viable way to improve existing searches is to teach the system the meaning of real world parameter values. So an expert system, for instance, can sample the most promising results for the user. Additional machine learning, a scientific discipline that is concerned with the design and development of algorithms, can be used to learn based on the proposed ontology. Consequently the users would be better off, if they not only received exact results, but also related outcomes. These related outcomes should ideally also somehow be searchable, allowing the users to interact and hence find more suitable results. In doing so the aim should be to organize the search results according to the ontology into meaningful categories (clusters). The benefits to the user include an impression of the available themes or topics (by the ontology itself), as well as an overview of related results (by the knowledge base).

A focal point is to construct an adaptive man-machine interaction interface as previous search interfaces do not capitalize sufficiently on the need of the users to interact with the web search engine in a straightforward manner. In order not to confuse the user, the interaction should be kept as simple as possible. The key for it, is an easily manageable GUI. Portmann et al. presents interactive cartographic visualization or interactive topic maps respectively as a crucial element of such a GUI [13, 14].

### 4 Conclusion and Outlook

Due to the fact that the boundaries in the fuzzy set theory are not rigorous, it is possible to find more and higher quality results with this kind of ontology. Based on the fuzzy grassroots ontology, YouReputation (www.youreputation.org), a prototype for online reputation analysis is at the early stage of development. Its GUI is designed as a dashboard, which allows users to browse related topics with the use of topic maps. Hence the appearance of the dashboard includes two parts, first the user-navigable visualization of the grassroots ontology as topic map and second the hit list of the knowledge base found by the search engine (Fig. 2).

The topic map helps to see search results by topic that a user can zoom in on exactly what he looked for or discover unexpected relationships between items for example by scrolling with his mouse. So terms visualized further away from a topic belong to it to a less significant level, than terms closer to the topic. The same applies to the relationship of the topic itself; topics nearer to other topics are more akin. So rather than scrolling through page after page, the topic map help find results which would have been missed or that were buried deep in the ranked list. The topics itself will be defined by the fuzzy grassroots ontology.
Figure 2: The appearance of the interactive dashboard.

Strictly following the methods of prototyping and to increase comprehension, here always the simplest formulas and algorithms are used to highlight the ideas in this paper (cf. lex parsimoniae). However, further tests can include variations of more advanced formulas and algorithms. For example, it can be evaluated whether there are potentially ‘superior’ measures to the Jaccard similarity measurement [8]. According the law of parsimony, the exact meaning of superior can be nuanced in the first place; ergo, it must be set what superior means during the process of evaluation. The same applies to the following prospects for improvements.

Further experimental tests include, for example, comparisons with commonly used non-metric measurements [1], variations of Soundex algorithm [9] or FCM competitors as fuzzy self-organizing maps (FSOM) and fuzzy clustering by local approximation of memberships (FLAME) [5, 20]. Although the implementation of the prototype focuses 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 [15]).

Acknowledgements

A special thank goes to the Fuzzy Marketing Methods Research Center (www.fmsquare.org) of the University of Fribourg for contributing valuable thoughts.

References