A Fuzzy Grassroots Ontology for improving Weblog Extraction

Edy Portmann, Andreas Meier

Information Systems Research Group
Department of Informatics
University of Fribourg
Boulevard de Pérolles 90
CH-1700 Fribourg
Switzerland
{edy.portmann, andreas.meier}@unifr.ch

Abstract: *This paper presents fuzzy clustering algorithms to establish a grassroots ontology – a machine-generated weak ontology – based on folksonomies. Furthermore, it describes a search engine for vaguely associated terms and aggregates them into several meaningful cluster categories, based on the introduced weak grassroots ontology. A potential application of this ontology, weblog extraction, is illustrated using a simple example. Added value and possible future studies are discussed in the conclusion.*

Keywords: Fuzzy Data Clustering, Information Retrieval, Ontology, Search Engine, Semantic Web, Weblog

1. Motivation and Related Work

In Information Retrieval (IR), it is pivotal to know how to find relevant information despite to information overload. However, defining relevance is a challenge. Do the retrieved documents and/or queries resulting from a set of keywords exactly match the semantic content of the given search terms?

The problem with the resulting documents and queries is that they often yield results which are only partially relevant to their actual semantic contents. As a consequence, the matching of a document to the query terms is often vague. The users would be frequently better off if they not only received exact results, but also related outcomes (presented as *suggestions*). These *suggestions* ideally should also be searchable, allowing the users to interact during the search process and hence find more suitable results, narrow the query or expand it as desired.

This paper discusses an approach to find more appropriate information by searching weblogs using fuzzy logic, here referred to as fuzzy weblog extraction. In [1], Meier, Schindler and Werro describe fuzzy logic as an appropriate instrument for rough modelling the kind of uncertainty related with vagueness. The core power of fuzzy logic is the fuzzy set theory, first proposed by Zadeh in [2] as an extension of the traditional set theory.

This paper shows that fuzzy sets can overcome the gap between the bottom-up-approach of folksonomies and the top-down-approach of ontologies, because fuzzy sets are more suitable for characterizing vague information. Murthy and Biswas state in [3] that 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 judgement. Hence fuzzy logic follows the way humans think and helps to handle real world complexities more efficiently. Therefore, it converts imprecise human information to precise mathematical models.

Consider, for example, the following paradox of a marriage: I cannot live with her, and I cannot live without her. Both statements are – to a certain degree – true. The dynamic between those statements is – along with other things – what keeps marriage interesting. That is what fuzzy logic deals with. Down with both statements (with, without) in fuzzy set theory certain membership degrees comes along.

In this proposal folksonomies – human-made taxonomies – will be converted to machine-understandable ontologies adapted from fuzzy set theory. To search for information from weblogs, a user types one or more search terms in a graphical user interface (GUI) and defines an associated relevance for each of them (as a membership function). This *user need* is then processed by a query engine, which generates term- and relevance-based clusters. The relationship grade depends on the relevance selected by the user. To discover the related terms, the query engine primarily draws upon a previously built ontology.

The ontology, which is a set of associated tags with related weights, is compiled with a folksonomy. The weights are represented using the semantic closeness of the terms. To achieve this, it is essential to establish terms and their relationships to each other, as Hasan-Montero and Herrero-Solana discuss in [4]. They propose an algorithm for semantic clustering and provide an example of its use. Furthermore, Kaser and Lemire suggest in [5] that associated tags ought to be placed near one another. The easiest possible way similarity between two tags can be measured is to count the number of co-occurrences, that is, the number of times two tags are allocated to the same source. Nevertheless, there exist other, different measurements to establish similarity, such as locality sensitive hashing (LSH) – where the tags are hashed, so that similar tags are mapped to the same set with a high probability – and collaborative filtering (CF) – where several users define tags and their relations jointly – as well.

After the similarities among all necessary tags are calculated, it is possible to derive clusters with the help of fuzzy clustering algorithms – for instance, the fuzzy c-means (FCM) algorithm described in [6] by Bezdek. The resulting clusters are the initial point to obtain an ontology and the basis for the suggestions related to the search. By this means the clusters can be anticipated in relation with the *user need* as folders and contain the individual term-based *search results*; in addition they are searchable – as a result the user can *interact*. The established *search results* of all the documents represented will be compiled by a meta search engine. A meta search engine permits the user to enter the query once and thereby it accesses several search engines at once, as Schwartz in [7] explains.

The benefit to the user is that instead of delivering millions of search results in one long list, this search engine groups similar results together into clusters and presents them in folders like Ferragina and Gulli in [8] propose. These folders help to recognize search results by topic so it is possible to zero in on precisely what was searched for or expose unforeseen relationships among items. Rather than scrolling through multiple pages, the folders help to uncover missed results or results that were hidden in a ranked list.

The paper is structured as follows: Chapter 2 clarifies the main elements of the proposed concept. Chapter 3 describes the research topic in more depth; furthermore a simple example is given. Chapter 4 summarizes the results and shows potential issues for further research.

2. Concept and Associated Components

For better understanding, this chapter first clarifies the main fundamentals of this paper. In chapter 2.1, necessary Semantic Web issues are outlined. The next chapter 2.2 describes social software (and its applications), followed by Information Retrieval (including web search engines) in chapter 2.3. Then common metrics (chapter 2.4) will be explained, since the fuzzy data clustering (chapter 2.5) is based on a metric-based tag space – a two dimensional space model. At the end, the resulting ontology is described in chapter 2.6. Each of these fundamentals will be briefly demonstrated and related academic work will be provided as well.

2.1 From Web 2.0 to Web 3.0

The expression Web 2.0 refers to the alleged next generation of web improvement, which aims to ease communication, promote safe information sharing, and enable interoperability and collaboration on the Internet. Thus Web 2.0 concepts have led to the expansion and development of web-based communities, hosted services, and applications such as *social software*, to cite an example. A major reason for their overnight success is the breathtaking simplicity of use. These sites do not only afford data but also generate a plethora of weakly arranged meta data.

Although the term Web 2.0 seemed to announce a new version of the Internet, according to O'Reilly, it is just a shift, not on technical, but on social interaction, as for example software developer and end-user use the Internet. In [9], O'Reilly declares that Web 2.0 technically does not differ from the earlier Internet, retrospectively marked as Web 1.0. In contrast to this static expert generated content in Web 1.0, interactive elements are crucial in Web 2.0.

At present, the Web 3.0 can amend the bottom-up wisdom-of-the-crowds attempt of the Web 2.0 in a top-down manner, as Cardoso states in [10]. Its fundamental aim is a stronger knowledge representation as is possible with folksonomies, for example. While users provide their data, they generally have a structure in mind. Indeed, this structure is buried in the data and it needs to be extracted for advanced use. Different kinds of self-acting mechanisms are indispensable to extract hidden information and to reveal the underlying structure in a profitable way for the end user. Using established methods to represent knowledge gained from unstructured data will be beneficial for the Web 2.0 too, in that it provides users with enriched Semantic Web features to organize and understand their data. Hence it is possible for the user to generate new *knowledge* as knowledge is the understanding of a subject with the ability to use it for a specific purpose if appropriate.

2.2 Folksonomy and Weblogs

Social software covers a collection of tools that empowers users to interact and share data, as *communication* and *interactive* applications. *Communication* tools such as weblogs typically handle the capturing, storing and presentation of statements. *Interactive* tools manage interceding interactions between user groups. They focus on establishing and obtaining a relation amongst users, facilitating the mechanics of conversation. A typical example of interaction tools are folksonomies.

The portmanteau "folksonomy" from 'folk' and 'taxonomy' means the practice and technique of collaboratively creating and manipulating tags to annotate and categorize content. By this means a document, a uniform resource locator (URL), a picture, a movie, etc. can be marked as content. As Voss explains in [11], in folksonomies, freely chosen keywords are used instead of a controlled vocabulary. Such meta data are straightforward to create, but generally lack any kind of formal grounding, as used in the Semantic Web. In this sense folksonomies will be used as a starting point to harvest social knowledge from these user-generated taxonomies and to build-up an ontology, as Wu, Zubair and Maly suggest in [12].

The second element important to mention are weblogs or short blogs. As Picot and Fischer illustrate in [13], a "weblog" is a made-up word of 'World Wide Web' (WWW) and 'log' and describes a type of website, usually administrated by an individual with periodical reverse-chronological entries of comments, description of events, or other objects such as movies, pictures or diagrams. Large quantities of blogs are composed of commentary, news or information on special topics. A typical weblog combines text, images, and essential links to other blogs, web pages, and additionally to other media. Therefore, an important advantage is that blogs based on hyperlinks typically inform faster than broadcast or print media. Thus, the latest information on specific topics is generally found in weblogs.

2.3 Information Retrieval and Search Engines

Information Retrieval is interdisciplinary, based on computer science, and establishes the retrieval of information from a set of documents as Baeza-Yates and Ribeiro-Neto outline in [14]. Accordingly, IR represents the science of searching for documents, for information within documents and for meta data about documents, as well as that of searching databases and the WWW. There is a partial coverage in the handling of the terms Data Retrieval, Document Retrieval, Information Retrieval, and Text Retrieval, but each has its own body of literature, practices, technologies and theory.

Normally, IR systems are used to diminish what is called *information overload*. Web search engines are the most obvious IR application. A web search engine is an instrument intended to search for information on the Internet. 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.

A number of search engines mine data by dint of a *web agent* available in newsbooks, databases, or open directories. Particular search engines as Technorati, Blogdigger, etc. are special search engines for searching blogs. To index the underlying sources the search engine draws on web agents.

A web agent is a program that accumulates pages from the Internet in an automated and methodical way. According to Kobayashi and Takeda in [15] there are other terms such as ants, automatic indexers, bots, crawlers, worms, or web spiders and web robots.

Primarily, these agents are used to create a copy of all the visited pages for later processing by the search engine that will list the pages to provide fast and sophisticated searches. Therefore, these agents gather meta data from web pages, such as tags from folksonomies.

Hence, the web agent initially starts with a list to visit, called the “seeds”. While the agent visits these seeds, it identifies all the tagged sources in the site and subjoins them in the so-called “crawl frontier list”. Sources from the crawl frontier are visited recursively in accordance to a set of conventions.

2.4 Distance Metrics

A metric is a function which defines a distance between elements. Therefore a distance metric attends to the analysis of differences. Adequate distance indices should consider the character of different scales and at the same time undertake the effort to figure out the common most significant information content of different elements.

Distance metrics attend to the identification of the distance between two individual elements. The basis for distance measurement is the *Minkowski* metric as discussed by Su and Chou in [16]:

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

The critical factor in this equation is to obtain the constant $r\ (\geq 1)$, which defines the Minowski metric. In so doing, one has to consider that Minowski metrics count as a matter of principle on dissimilarity measurements; that is, the bigger the measure the more dissimilar the single elements are. In many cases, not the distance but the similarity is desired. Commonly the similarity is obtained by subtracting the particular distance from 1. Therefore it is possible to calculate one from the other.

For metric data mainly four distance measurements are used: the Euclidean, the squared Euclidean, the Block- and Chebyshev's distance.

However, for non-metric data, coefficients such as for example the Dice, the Jaccard, the Kulczynski, the Russel and Rao, the Simple Matching and the Tanimoto coefficient, are used widely, as Backhaus, Erichson, Plinke and Weiber in [17] explain; multiple non-metric but set-based ordinal distance used in Information Retrieval are the Jaccard, the Simple Matching and lastly the Dice coefficient.

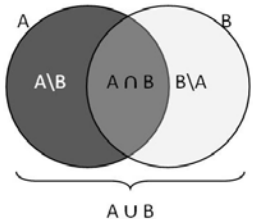

Figure 1. Venn diagram.

The *Jaccard* coefficient for example measures similarity between sample sets. Let A and B be the sets of resources so the Jaccard coefficient is defined as the size of the intersection divided by the size of the union of the sample sets (cf. Venn diagram in fig. 1):

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

Another commonly used coefficient is the *Simple Matching* coefficient. This coefficient for a set A and B is calculated as:

$$d_{SM}(A,B) = \frac{|A \cap B|}{|A| + |B|} \tag{3}$$

The *Dice* coefficient is a modification of the Simple Matching coefficient. The Dice coefficient for a set A and B is obtained with:

$$d_D(A,B) = \frac{2\,|A \cap B|}{|A| + |B|} \tag{4}$$

2.5 Fuzzy Logic, Fuzzy Sets and Fuzzy Data Clustering

Fuzzy logic allows the modeling of uncertainty associated with vagueness and imprecision and putting this into appropriate mathematical equations. Human reasoning is not dichotomous, unlike software programs, where everything is either true or false. It deals with imprecision and the conceptions are ambiguous in the sense that they cannot be sharply defined. For instance, the question whether the temperature is ‘hot’ or not cannot be unanimously answered. Despite the fact that the definition of the word ‘hot’ is clear, it is not possible to clearly state if a temperature is hot because the answer may depend on the individual perception. For a person it may even not be possible to give a precise and clear answer as belonging to a concept (e.g. hot temperature) is often not sharp but fuzzy, involving a partial matching expressed in the natural language by the expressions ‘fairly, ‘slightly’, ‘more or less’, etc.

In figure 2, the meaning of the expressions ‘cold’, ‘warm’, and ‘hot’ is represented by functions mapping a temperature scale. A point on that scale has three *truth values* — one for each of the three functions. The vertical line in the image represents a particular temperature that the three arrows (truth values) determine. Since the topmost arrow (one) points at 0.8, this temperature may be interpreted as ‘fairly cold’. The second arrow (two) pointing at 0.3 may be described as ‘slightly warm’ and the arrow at the bottom (three) points to zero, as ‘not hot’.

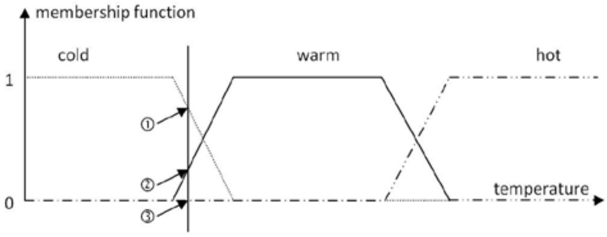

Figure 2. Fuzzy logic temperature sets (including truth values).

Based on fuzzy logic, the fuzzy set theory was developed in 1965 by Zadeh at the University of Berkeley, California. The fuzzy set theory is – built on intuitive deduction by considering human imprecision and subjectivity – not an inexact theory but a rigorous mathematical one which deals with uncertainty and subjectivity.

In [2] and [18], Zadeh, Dubois, Prade and Yager explain that in traditional set theory, the membership of elements in a set is assessed in binary terms corresponding to a two-valued condition. As a result, an element either belongs or does not belong to the set. Contrary to this, the fuzzy set theory allows for continuous assessment of the membership of elements in a set. According to Dubois and Prade in [19], fuzzy sets generalize the classical sets, since the indicator functions of classical sets are particular cases of the membership functions of fuzzy sets, where the second only take values 0 or 1.

The term “data clustering” describes the method of grouping data elements into clusters or classes, so that elements in the same cluster are as identical as possible, and elements in different clusters are as diverse as possible. Depending on the intention for which clustering is being used and the nature of the data, special metrics of relationship may be used to place items into clusters, where the relationship measure controls how the clusters are shaped. Hence one has to distinguish between hard and soft clustering.

In hard clustering, data is separated into distinctive clusters, where all data elements belong precisely to one single cluster. According to fuzzy set theory, Bezdek shows in [6] that in fuzzy clustering, data elements can belong to more than one cluster. Associated with each element is a set of membership levels, which 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.

2.6 Ontology

Ontologies – the term has its origin in philosophy – are in theory artifacts of objects and their ties. Hence ontologies 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. What is provided precisely can deviate, but these properties are fundamentals of every ontology.

According to Gruber in [20], an ontology is a "*formal, explicit specification of a shared conceptualization"*. There is an expectation that the model bear analogy to the real world as well; however, it definitely offers a common terminology which can be used to model a domain. A domain is the type of objects and concepts that exist, and their properties and relations

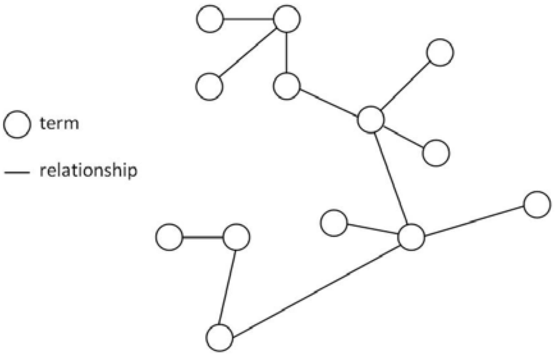

Figure 3. An example of a weak ontology.

In theory there is a distinction between strong and weak ontologies, whereas we use at this point only weak ontologies (cf. fig. 3). A weak ontology is one that is not sufficiently as rigorous as a strong one and therefore allows software to insert new details without an intervention by human beings. In addition, a weak ontology converges with Boolean logic and other subfields in which automatic reasoning is known to be possible.

3. The Fuzzy Grassroots Ontology

In the following, important aspects of this research are presented in more detail. To establish a common vocabulary, the scope of this research is specified in chapter 3.1. Next, each element of the proposed search engine is elaborated. This chapter closes with a simple example to emphasize the benefit of the fuzzy weblog extraction (chapter 3.2).

3.1 Building Blocks

Information overload leads to an important question: How can relevant information be extracted from Web 2.0 and Web 3.0 applications? One possible approach is the proposed extraction with the use of a fuzzy data clustering algorithm.

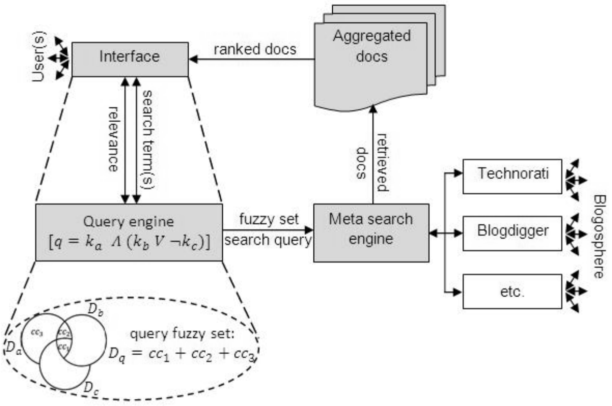

Figure 4. Overview over the weblog extraction process.

In figure 4, an overview over the fuzzy weblog extraction is given. The main parts are the graphical user interface (with integrated search query engine), the meta search engine, and the aggregated documents. Each element is discussed in more depth in the rest of this section.

3.1.1 Interface

To search for relevant information from weblogs, a user enters the search terms $\mathrm{k_a, k_b}$ and $\mathrm{k_c}$ in an interface, called *user need* (cf. chapter 1). Included in this user interface is a convenient tool (e.g. a slider) to determine the weight K of the key terms $\mathrm{k_a, k_b}$ and $\mathrm{k_c}$ and which generates the search query. A suitable example which clarifies this can be found in [14] by Baeza-Yates and Ribeiro-Neto. They define a search query [q = $\mathrm{k_a} \wedge (\mathrm{k_b} \vee \neg \mathrm{k_c})$]. In their example, each $\mathrm{cc_i}$, $\mathrm{i} \in \{1,2,3\}$ is a conjunctive module.

Let $\mathrm{D_a}$ be the fuzzy set of documents related to the term $\mathrm{k_a}$. This set may be composed, for example, of the documents $\mathrm{d_j}$ which have a degree of relationship $\mu_{\mathrm{a,j}}$ bigger than the already predefined weight K. Further, let $\overline{\mathrm{D}}_\mathrm{a}$ be the complement of the set $\mathrm{D_a}$. The fuzzy set $\overline{\mathrm{D}}_\mathrm{a}$ is related to $\overline{\mathrm{k}}_\mathrm{a}$, the negation of the term $\mathrm{k_a}$. Correspondingly, we can characterize fuzzy sets $\mathrm{D_b}$ and $\mathrm{D_c}$ related to the key terms $\mathrm{k_b}$ and $\mathrm{k_c}$, equally. Because every single one of these sets is fuzzy, a document $\mathrm{d_j}$ might belong to the set $\mathrm{D_a}$ for example, even if the document text $\mathrm{d_j}$ does not include the term $\mathrm{k_a}$.

After the entire *user need* is specified, it can be processed by the query engine.

3.1.2 The Query Engine

The query engine generates a fuzzy set search query from the provided data (*user need*). The query fuzzy set $\mathrm{D_q}$ is a fusion of the fuzzy set related with the three conjunctive components of $\mathrm{cc_1}$, $\mathrm{cc_2}$, $\mathrm{cc_3}$. The relationship $\mu_{\mathrm{q,j}}$ of a document $\mathrm{d_j}$ in the fuzzy answer set $\mathrm{D_q}$ is calculated as $\mu_{\mathrm{q,j}} = \mu_{\mathrm{cc_1+cc_2+cc_{3,j}}}$, where $\mu_{\mathrm{i,j}}, \mathrm{i} \in \{\mathrm{a,b,c}\}$ is the relationship of $\mathrm{d_i}$ in the fuzzy set associated with $\mathrm{k_i}$. In this case, the level of relationship in the disjunctive fuzzy set is calculated using an algebraic sum. Additionally, the level of relationship in a conjunctive fuzzy set is calculated using an algebraic product. The adoption of algebraic sums and products yields relationship levels which can vary seamlessly.

To generate an adequate fuzzy set search query, the query engine uses the data provided by web agents that crawled through the Internet and collected meta data (e.g. tags from folksonomies). In this sense the ability to find high-quality sources, such as documents or people, is important to overcome the information overload. Collaborative filtering systems, or recommender systems, identify high-quality sources utilizing individual knowledge. One known algorithm which is successful in identifying *high-quality sources* in a hyperlinked environment automatically is the Hyperlink-Induced Topic Search (HITS) algorithm proposed by Kleinberg in [21].

HITS starts from a small root set of documents to a larger set T by adding up documents that link to and from the documents in the root set. The ambition of the algorithm is to identify hubs, the documents that link to numerous high quality documents, and authorities, the documents that are linked from numerous high quality documents. The hyperlink structure amongst the documents in T, is exposed by the adjacency matrix A, where $\mathrm{A_{ij}}$ denotes whether there is a link from document $\mathrm{d_i}$ to document $\mathrm{d_j}$. By means of this matrix A, a weighting algorithm constantly updates the hub weight and authority weight for every document, until the weights converge. Essentially, the hubs and authorities are documents with biggest values in the main eigenvectors of $\mathrm{A^TA}$ and $\mathrm{AA^T}$, correspondingly.

A well-known problem in folksonomies, where people choose their own tags to annotate sources, is that *typing errors* can occur since there is no editorial supervision. This leads to overlapping but barely relating terms in the underlying ontology. Certainly it can be assumed that a search system finds relevant information despite misspelling in tags, because the queries contain the same mistakes. But the necessity of fault-tolerant treatment of queries becomes clear. According to Lewandowski in [22] one has to distinguish between different types of typing error improvements as for example *dictionary-based* and *statistical approaches*. Within the *statistical* approaches, moreover, we need to distinguish between the single-word and the phrase-based approaches as well.

Dictionary-based approaches compare entered query terms with a dictionary and if the query term is not covered by the dictionary they search for similar terms.

Statistical methods refer by misspellings with no or only few hits to the most commonly used similar syntax. To determine the phonetic similarity the words will be reduced to a code, which conforms to similar terms. A well-known example especially for the English language is the Soundex algorithm – patented by Russell – for indexing names by sound. The goal is for homophones to be encoded to the same representation so that they can be matched despite minor differences in spelling. The patents to this can be found in [23] and [24]. The algorithm mainly encodes consonants; a vowel will not be encoded unless it is the first letter. The Soundex algorithm as an outline is:

1. Capitalize all letters in the word and drop all punctuation marks.
2. Retain the first letter of the word.
3. Change all occurrence of the letters
   - $'A', E', 'I', 'O', 'U', 'H', 'W', 'Y' \rightarrow 0$
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.

Improvements to the Soundex algorithm, as for example a fuzzy Soundex algorithm as Holmes and McCabe present in [25], are the basis for many modern phonetic algorithms and can be used to correct misspellings in taxonomies. A major advantage of the utilization of a Soundex algorithm is that the correctly spelled ontology terms can as well be used as a kind of auto-completion and -suggestion while the user is typing search terms in the interface.

Nevertheless, the recovered tags will be arranged as Hasan-Montero and Herrero-Solana in [4] suggest. Tag similarity is measured as a kind of semantic correlation between tags, considered by means of relative co-occurrence among tags. This is the already presented Jaccard similarity coefficient, which is, according to these authors, superior to other coefficients. Let A and B be the sets of resources characterized by two tags, relative co-occurrence is ascertained with (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 any one of the two tags appear. This collection method causes these tags to become united and offers a semantically consistent picture where nearly all tags are related to each other. This semantically consistent picture is referred to as tag space.

Hence these tags will be sorted by a fuzzy clustering algorithm, for instance the aforementioned FCM algorithm by Bezdek in [6].

The FCM algorithm attempts to split a limited collection of elements X = $\{x_1, . . ., x_n\}$ into a assortment of c fuzzy clusters with regard to some specified condition. In fuzzy clustering, each point has a level of belonging to clusters, as in fuzzy logic, rather than belonging to just one particular cluster. Thus, points on the edge of a cluster may be *in the cluster* to a less significant level than points *in the center of cluster*. For each point x there is a coefficient giving the grade of being in the k^{th} cluster $u_k(x)$. Characteristically, the sum of those coefficients is defined as 1:

$$\forall_x \left(\sum_{k=1}^{\text{num. cluster}} u_k(x) = 1 \right) \tag{5}$$

The focal point of a cluster is by fuzzy c-means, the average of all points, weighted by their amount of belonging to the cluster:

$$\text{center}_k = \frac{\sum_x u_k(x)^m x}{\sum_x u_k(x)^m} \tag{6}$$

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

$$u_k(x) = \frac{1}{d(\text{center}_k, x)} \tag{7}$$

In that case the coefficients are normalized and fuzzyfied with a true parameter $m(> 1)$ so that their sum is 1. Hence,

$$u_k(x) = \frac{1}{\sum_j \left(\frac{d(\text{center}_k, x)}{d(\text{center}_j, x)} \right)^{2/(m-1)}} \tag{8}$$

For m equal to 2, this is the same as normalizing the coefficients linearly to make their sum 1. When m is close to 1, then the cluster center closest to the point is given a considerably larger amount extra weight than the others.

The FCM algorithm focuses on minimizing an objective function. To generate an ontology the proposed, extended standard function is:

1. Select an amount of clusters.
2. Assign coefficients erratically to each point for being in the clusters.
3. Reiterate until the algorithm has converged (that is, the coefficients' adjust among two iterations is no more than ε, a given sensitivity boundary value):
   a. Calculate the centroid for each cluster, using the formula (6) above.
   b. For each point, compute its coefficients of being in the clusters, using the formula (8) above.
4. Reiterate step 1 to 3 for every cluster until there is only one term left in the cluster.
5. Concatenate all same terms together.

The concatenation is necessary because the terms can – by drawing on the proposed fuzzy clustering method – belong to more than one cluster. Nevertheless, with the proposed method a model can be derived (cf. dendrogramm in fig. 5) with several clusters the term belongs to a certain degree in, dependent on the membership level. This model is referred to as weak ontology.

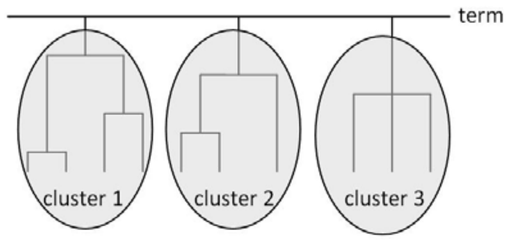

Figure 5. Dendrogramm with example clusters.

With the FCM algorithm – on the basis of the collected tags and in alliance with the chosen weight K (*user need*) of the key terms $\mathrm{k_a, k_b}$ and $\mathrm{k_c}$ (cf. fig. 4) – it is possible to identify all related terms to a certain degree. These terms are sub-summarized with *fuzzy set search query*, which after that will be sent to a meta search engine. The belonging degrees of the terms later are used as ranking factor.

3.1.3 The Meta Search Engine

A meta search engine sends the provided terms of the fuzzy set search query to numerous weblog search engines, such as Technorati, Blogdigger, etc. (cf. fig. 4).

The fuzzy set search query contains the term searched for, and the user-chosen related terms are drawn on the build ontology.

After the completed search, the meta search engine aggregates the results and displays them clustered in folders according to their source.

To rank the returned documents inside the clusters, machine learning (ML) – as Taylor et al. in [26] explains – could be used to discover appropriate web ranking functions. An input for ML can be the degrees of affiliation for the ontology-based terms found.

3.1.4 Aggregated Documents

The aim is to organize the search results into several meaningful categories (*clusters*). These clusters are a group of similar topics related to a term. The benefits to the user include an impression of the available themes or topics, as well as an overview of related results in folders rather than scattered throughout a list.

The basis for the definition of the clusters is, as previously, the preliminarily built grassroots ontology.

3.2 Example of Information Screening

To explain the benefit of this proposed fuzzy weblog extraction approach, let's introduce a small example. In the first part of this section the example will be explained. Afterwards the results of the *boolean search* will be compared with the result of the *fuzzy search*.

3.2.1 Problem Specifications

The problem with new and previously unobserved information is that the relationship to a term or other terms is not precisely known. For example, the screen-producing company, Samsung, is screening the business competitors in the Internet for new killer applications for organic light emitting diodes (OLEDs).

An OLED (very often also named Organic Electro Luminescence – OEL) is any light emitting diode (LED) whose emissive electroluminescent layer is made up of a film of organic compounds. The layer typically contains a polymer substance that allows appropriate organic compounds to be deposited. The completely different manufacturing process of OLEDs lends itself to various advantages over flat-panel displays prepared with Liquid Cristal Display (LCD) technology, as OLEDs enable a larger variety of colors, gamut, brightness, contrast and viewing angle than LCDs, due to the fact that OLED pixels directly emit light.

... OLED .....

... LED .....

... LCD .....

... OEL .....

Figure 6. Related terms in four different weblogs.

In the Blogosphere new technologies are discussed earlier and therefore information is spread faster than in most other media. Thus in this example there are four weblogs (A to D) with different entries related to OLEDs. As shown in figure 6 the terms mentioned in these weblogs are OLED, LED, LCD, OEL.

3.2.2 Pre-search

To create an ontology, an agent first crawls the Internet for tags. A web agent is – as already mentioned – a computer program that searches the Internet in a methodical, automated manner. Nevertheless this agent applies the previously discussed HITS algorithm with the specified Jaccard similarity coefficient to the set of tags it found to generate a tag space.

With the use of FCM, the specified tag space will be partitioned in several meaningful clusters, as for example an 'OLED' cluster (fig. 7). This cluster represents a part of the – with the FCM as well – built ontology with the focus 'OLED'. In this example, an arbitrary chosen range of relationship to the 'OLED' cluster is defined for OEL, LED and for LCD as well.

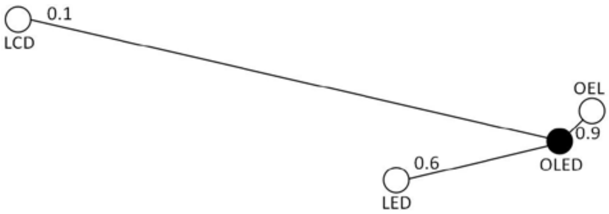

Figure 7. Weak ontology with reference to the example.

3.2.3 The search

To search for groundbreaking new OLED technology in these weblogs, a user enters search terms and related relevance in a simple-to-use interface such as for example, ‘OLED’ with a search range of [0.8..1] (see weight K in chapter 3.1). This range is defined with the use of a slide control.

In this example shown in figure 8a and 8b, the search terms and their relevance include the nucleus OLED. OEL is included as well, however only with a disc range of [0.9..1]. As depicted in figure 8b LED with the disc range of [0.6..1] is excluded.

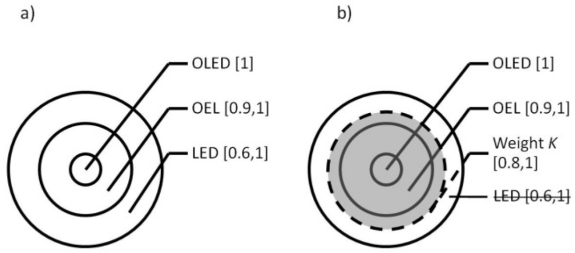

Figure 8a) Search with relation to other terms and 8b) with assigned weight K (≥ 0.8).

3.2.4 Results

With a *boolean search* it is only possible to find weblog A (fig. 9), because the term “OLED” is or is not mentioned in the weblog (two-valued logic). The other related terms cannot be found.

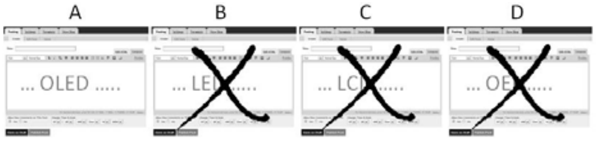

Figure 9. Boolean search result with reference to the example.

In contrast to the proposed *fuzzy search* approach, it is possible to find not only the searched term, but also to a certain degree related terms (as defined in the *user need*). As a result it is possible to find weblog A as well as weblog D (fig. 10). Thus, the benefit for the user is that he can discover new relationships. As illustrated above, the search for ‘OLED’ originates not only an ‘OLED’ result but also a ‘LED’ result. A transformation of the user predefined weight K leads to in- or exclusion of further related terms. This transformation can be performed easily with the use of the introduced slider.

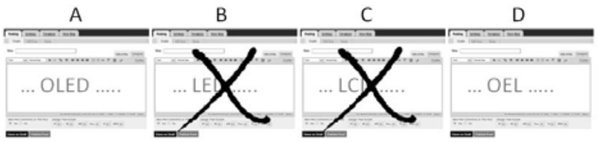

Figure 10. Search result based on weak ontology with reference to the example.

Instead of delivering millions of search results in one long list, the search engine groups similar results together into clusters. Clusters help to see *search results* by topic so it is possible to zero in on precisely what was searched for or discover unforeseen associations among elements. Rather than scrolling through page after page, the clusters help to find results missed without fuzzy weblog extraction or that were hidden deep in the ranked list.

Hence the two located weblogs will be presented to the user in two different folders. One folder is labeled ‘OLED’ and the other is labeled ‘OEL’.

4. Conclusion and Outlook

People questioned O’Reilly and Battelle ever since the term Web 2.0 was introduced, *"What’s next?"* as they state in [27]. *“Is it the Semantic Web? The Sentient Web? Is it the Social Web? The Mobile Web? Is it some form of Virtual Reality?”* and they answer with *“It is all of those, and more.”* They are probably right because *“the Web is no longer a collection of static pages [...]. Increasingly, the Web is the world – everything and everyone in the world casts an "information shadow", an aura of data which, when captured and processed intelligently, offers extraordinary opportunity and mind bending implications.”* as they answer the question themselves in [27].

A new approach to gain deeper insights in a part of this *“information shadow*” is the introduced *fuzzy weblog search engine*. Due to the fact that the boundaries in the fuzzy set theory are not well-defined, it is possible to find more numerous and higher quality results with this new Web 3.0 kind of a weblog search. The results should be presented in an understandable way by using folders as suggested in this paper. The folders will be defined by the fuzzy clusters through FCM.

A *prototype for fuzzy weblog extraction* is at the early stage of development (see architecture in fig. 4). It evaluates if there is a possibly superior coefficient to the Jaccard similarity coefficient proposed by Hasan-Montero and Herrero-Solana in [4]. Further tests include comparisons with the Dice, the Kulczynski, the Russel and Rao, the Simple Matching and the Tanimoto coefficient. Additionally, the HITS algorithm will be tested against other comparable algorithms as for example Google PageRank or Yahoo! TrustRank and with the variation of different associated Soundex algorithms it is also possible to vary. Another essential evaluation will be to weigh the FCM algorithm against other comparable algorithms, such as the “Fuzzy clustering by Local Approximation of Memberships” (FLAME) algorithm suggested by Fu and Medico in [28].

A focal point of this research is to construct an *adaptive man-machine interaction interface* as previous search engines capitalize only insufficiently on the need of the users to interact with the search engine in a straightforward manner. In order to not confuse the user, the interaction should be kept as simple as possible. Therefore the key for it is an easily manageable graphical user interface which has to be designed.

Other *2D or 3D visualization* possibilities as for instance Kuhn, Erni, Loretan and Nierstrasz with their approach for software visualization in [29] or Portmann and Kuhn for weblog search in [30] presents can be explored in further research. The overarching philosophy of the GUI should be to constantly simplify the search process for the user – for example by going into an interaction with them.

Another unexplored area for future research is the *dynamic storage* of the terms found by the agent. For this purpose, storage space is intended to use a multidimensional database, as a repository of the stored meta data. A conceivable approach to this is akin to the fuzzy data warehouse as proposed by Fasel and Zumstein in [31] for web analytics. For search speed reasons this data warehouse should possibly be split across several different machines in the long run. Consequently future balance-load considerations have to be taken into consideration.

Acknowledgement

Upon the recommendation of the referees this paper is a revised version of Portmann's paper – which can be found in [32] – for the 2nd International Conference on the Applications of Digital Information and Web Technologies (ICADIWT2009) in London. That is why we are grateful to the referees for this honor.

We also appreciate our colleagues of the Fuzzy Marketing Methods research center (www.FMsquare.org) who always help with word and deed. This center implements the ideas of fuzzy classification methods to numerous fields of applications and therefore, the research center welcomes cooperation with practitioners as well as researchers.

5. References

[1] Meier, A., Schindler, G., Werro, N. (2008). Fuzzy classification on relational databases. *In: Galindo M. (Ed.): Handbook of Research on Fuzzy Information Processing in Databases, Volume II, Information Science Reference*, pages 586-614, 2008.

[2] Zadeh, L. A. (1965). Fuzzy Sets. *Information and Control, 8*, pages 338-353, 1965.

[3] Murthy, S.G.K., Biswas, R.N. (2004). A Fuzzy Logic Based Search Technique for Digital Libraries. *DESIDOC Bulletin of Information Technology, Vol. 24, No. 6*, pages 3-9, November 2004.

[4] 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,* Mérida, October 2006.

[5] Kaser, O., Lemire, D. (2007). Tag-Cloud Drawing: Algorithms for Cloud Visualization. *Electronic Edition*, Banff, 2007.

[6] Bezdek, J.C. (1981). Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York, 1981.

[7] Schwartz, C. (1998). Web Search Engines. *Journal of the American Society for Information Science. 49(11),* pages 973–982, 1998.

[8] Ferragina, P. Gulli, A. (2005). A Personalized Search Engine Based on Web-Snippet Hierarchical Clustering. *International World Wide Web Conference Committee*, pages 801-810, Chiba, Japan, May 2005.

[9] Oreilly, T. (2007). What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software. *Communications & Strategies, No. 1*, page 17, First Quarter 2007.

[10] Cardoso, J. (2007). The Semantic Web Vision, Where Are We? *IEEE Computer Society*, pages 22-26, 2007.

[11] Voss, J. (2007). Tagging, Folksonomy & Co - Renaissance of Manual Indexing? *Proceedings of the International Symposium of Information Science*, pages 234–254, Cologne, 2007.

[12] Wu, H., Zubair M., Maly, K. (2006). Harvesting Social Knowledge from Folksonomies. *ACM*, pages 1-12, Odense, Denmark, August 2006.

[13] Picot, A., Fischer, T. (2006). Weblogs professional, fundamentals, concept and practice in a corporate environment (in German). Dpunkt, Heidelberg, 2006.

[14] Baeza-Yates, R., Ribeiro-Neto, B. (1999). Modern Information Retrieval. ACM press, Essex, 1999.

[15] Kobayashi, M., Takeda, K. (2000). Information Retrieval on the Web. *IBM Research, ACM Computing Surveys, Vol. 32, No. 2*, pages 144-173, June 2000.

[16] Su, M.C., Chou, C. H. (2001). A Modified Version of the K-Means Algorithm with a Distance Based on Cluster Symmetry. *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No. 6*, pages 674-680, June 2001.

[17] Backhaus, K., Erichson, B., Plinke, W., Weiber, R. (2006). Multivariate Analysis, an application-oriented adoption (in German). Springer, Berlin, 2006.

[18] Dubois, D., Prade, H., Yager, R.R. (1993). Readings in Fuzzy Sets for Intelligent Systems. Morgan Kaufmann Publishers, San Mateo, 1993.

[19] Dubois, D., Prade, H. (1988). Fuzzy Sets and Systems. Academic Press, New York, 1988.

[20] Gruber, T. R. (1993). A Translation Approach to Portable Ontology Specifications. *Knowledge Systems Laboratory, Technical Report KSL 92-71*, pages 199-220, April 1993.

[21] Kleinberg, J. (1998). Authoritative sources in a hyperlinked environment. *ACM-SIAM Symposium on*

Discrete Algorithms, pages 1-33, Odense, Denmark, 1998.

[22] Lewandowski, D. (2005). Web Information Retrieval, technologies for information search in the Internet (in German). Deutsche Gesellschaft f. Informationswissenschaft u. Informationspraxis, Düsseldorf, 2005.

[23] Russel, R. C. (1918). US Patent 1261167, 1918.

[24] Russel, R. C. (1922). US Patent 1435663, 1922.

[25] Holmes, D., McCabe, M. C. (2002). Improving Precision and Recall for Soundex Retrieval. *Proceedings of the International Conference on Information Technology: Coding and Computing*, page 22, 2002.

[26] Taylor, M., Zaragoza, H., Craswell, N., Robertson, S., Burges, C. (2006). Optimisation methods for ranking functions with multiple parameter. *CIKIM*, pages 585-593, 2006.

[27] O'Reilly, T., Battelle, J. (2009). Web Squared: Web 2.0 Five Years On. *Web 2.0 Summit, O'Reilly Media, Inc*, pages 1-13, 2009.

[28] Fu, L., Medico, E. (2007). FLAME, a novel fuzzy clustering method for the analysis of DNA microarray data. *BMC Bioinformatics*, pages 1-15, 2007.

[29] Kuhn, A., Erni, D., Loretan, P., Nierstrasz, O. (2009). Software Cartography: Thematic Software Visualization with Consistent Layout. *Proceedings of 15th Working Conference on Reverse Engineering, IEEE Computer Society Press*, pages 209-218, October 2008.

[30] Portmann, E., Kuhn, A. (2010). Extraction and Cartography of Weblog Information" (in German). *in HMD271 Web 3.0 & Semantic Web,* 2010 (forthcoming).

[31] Fasel, D., Zumstein, D. (2009). A Fuzzy Data Warehouse Approach for Web Analytics. *In: Miltiadis d. Lytras, E. Damiani, J.M. Carroll, R.D. Tennyson, D. Avison, A. Naeve, A. Dale, P. Lefrere, F. Tan, J. Sipior, and G. Vossen, editors, Visioning and Engineering the Knowledge Society – A Web Science Perspective, volume 5736 of Lecture Notes in Computer Science*, pages 276–285, Springer, 2009.

[32] Portmann, E. (2009). Weblog Extraction with Fuzzy Classification Methods. *2nd International Conference on the Applications of Digital Information and Web Technologies,* pages 422-427, London, August 2009.