How to Visualize a Crisp or Fuzzy Topic Set over a Taxonomy

Boris Mirkin[1,2], Susana Nascimento[3], Trevor Fenner[2], and Rui Felizardo[3]

[1] Division of Applied Mathematics and Informatics, National Research University - Higher School of Economics, Moscow, Russian Federation
[2] Department of Computer Science, Birkbeck University of London London WC1E 7HX, UK
[3] Department of Computer Science and Centre for Artificial Intelligence (CENTRIA) Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa 2829-516 Caparica, Portugal

Abstract. A novel method for visualization of a fuzzy or crisp topic set is developed. The method maps the set's topics to higher ranks of the taxonomy tree of the field. The method involves a penalty function summing penalties for the chosen "head subjects" together with penalties for emerging "gaps" and "offshoots". The method finds a mapping minimizing the penalty function in recursive steps involving two different scenarios, that of 'gaining a head subject' and that of 'not gaining a head subject'. We illustrate the method by applying it to illustrative and real-world data.

1 Background and Motivation

The concept of ontology as a computationally feasible environment for knowledge representation and maintenance has sprung out rather recently. The term refers, first of all, to a set of concepts and relations between them. These pertain to the knowledge of the domain under consideration. At the inception, the relations typically have been meant to be rule-based and fact-based. However, with the concept of "ontology" expanding into real-world applied domains such as in biomedicine, it would be fair to say that the core knowledge in ontology currently is represented by a taxonomic relation that usually can be interpreted as "is part of". Such are the taxonomy of living organisms in biology, ACM Classification of Computing Subjects (ACM-CCS) [1], and more recently a set of taxonomies comprising the SNOMED CT, the 'Systematized Nomenclature of Medicine Clinical Terms' [15]. Most research efforts on computationally handling ontologies may be considered as falling in one of the three areas: (a) developing platforms and languages for ontology representation such as OWL language (e.g. [14]), (b) integrating ontologies (e.g. [17,7,4,8]) and (c) using them for various purposes. Most efforts in (c) are devoted to building rules for ontological reasoning and querying utilizing the inheritance relation supplied by the ontologys

S.O. Kuznetsov et al. (Eds.): PReMI 2011, LNCS 6744, pp. 3–12, 2011.
© Springer-Verlag Berlin Heidelberg 2011

taxonomy in the presence of different data models (e.g. [5,3,16]). These do not attempt at approximate representations but just utilize additional possibilities supplied by the ontology relations. Another type of ontology usage is in using its taxonomy nodes for interpretation of data mining results such as association rules [10,9] and clusters [6]. Our approach naturally falls within this category. We assume a domain taxonomy has been built. What we want to do is to use the taxonomy for representation and visualization of a query set comprised of a set of topics corresponding to leaves of the taxonomy by related nodes of the taxonomy's higher ranks. The representation should approximate a query topic set in a "natuaral" way, at a cost of some "small" discrepancies between the query set and the taxonomy structure. This sets our work apart from other work on queries to ontologies that rely on purely logical approaches [5,3,16].

Computational treatises such as [11] mainly rely on the definition of visualization presented in the Merriam-Webster dictionary regarding the transitive verb "visualize" as follows: "to make visible, to see or form a mental image of" (see http://www.merriam-webster.com/dictionary/visualize). Here we assume a somewhat more restrictive view that computational visualization necessarily involves the presence of a ground image the structure of which should be well known to the viewer. This can be a Cartesian plane, a geography map, or a genealogy tree, or a scheme of London's Tube . Then visualization of a data set is such a mapping of the data on the ground image that translates important features of the data into visible relations over the ground image. Say, objects can be presented by points on a Cartesian plane so that the more similar are the objects the nearer to each other the corresponding points. Or geographic objects can be highlighted by a bright colour on a map.

Such is the visualization for a company delivering electricity to homes in a town zone. Figure 1, taken from [2], represents the energy network over a map of the corresponding district on which the topography and the network data are integrated in such a way that gives the company "an unprecedented ability to control the flow of energy by following all the maintenance and repair issues on-line in a real time framework.

There are three major ingredients that allow for a successful representation of the energy network:

(1) map of the district (the ground image),
(2) the energy network units (entities to be visualized), and
(3) mapping (2) at (1).

The mapping here needs not be overly complicated because the units are located at the very same ground image in real. Moreover, one could imagine an extension of this mapping to other infrastructure items, such as the water supply, sewage type, and transports, so that the map could be used for more long-term city planning tasks such as development of leisure or residential areas and the like.

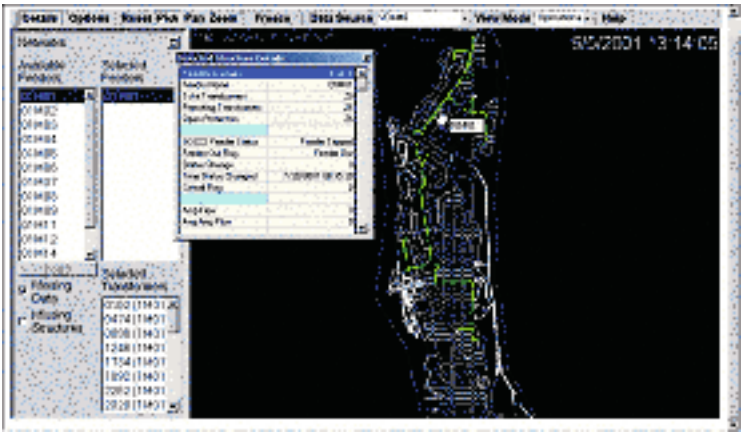

Fig. 1. Energy network of Con Edison Company on Manhattan New-York USA visualized by Advanced Visual Systems [2]

Is a similar mapping possible for a long-term analysis of an organization whose activity is much less tangible? For a research department, the following analogues to the elements of the mapping in Fig. 1 can be considered:

(1') a tree of the ACM-CCS taxonomy of Computer Science, the ground image,
(2') the set of CS research subjects being developed by members of the department, and
(3') representation of the research on the taxonomy tree.

Potentially, this can be used for:

- Positioning of the organization within the ACM-CCS taxonomy;
- Analyzing and planning the structure of research being done in the organization,
- Finding nodes of excellence, nodes of failure and nodes needing improvement for the organization;
- Discovering research elements that poorly match the structure of AMS-CCS taxonomy;
- Planning of research and investment
- Integrating data of different organizations in a region, or on the national level, for the purposes of regional planning and management.

2 Lifting Model and Method

2.1 Statement of the Problem

We assume that there are a number of concepts in an area of research or practice that are structured according to the relation "a is part of b" into a taxonomy,

that is a rooted hierarchy T. We denote the set of its leaves by I. Each interior node $t \in T$ corresponds to a concept that generalizes the concepts corresponding to the subset of leaves $I(t)$ descending from t, viz. the leaves of the subtree $T(t)$ rooted at t, which will be referred to as the leaf-cluster of t.

A fuzzy set on I is a mapping u of I to the non-negative real numbers assigning a membership value $u(i) \geq 0$ to each $i \in I$. We refer to the set $S_u \subset I$, where $S_u = \{i : u(i) > 0\}$, as the support of u.

Given a taxonomy T and a fuzzy set u on I, one can think that u is a, possibly noisy, projection of a high rank concept to the leaves I. Under this assumption, there should exist a "head subject" h among the interior nodes of the tree T that more or less comprehensively (up to small errors) covers S_u. Two types of possible errors are gaps and offshoots as illustrated in Figure 2.

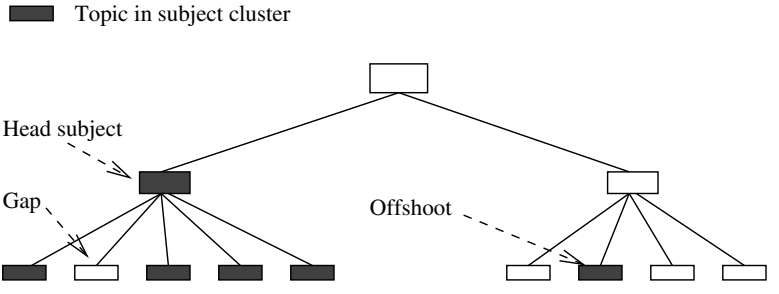

Fig. 2. Three types of features in lifting a topic set within taxonomy

A gap is a maximal node g in the subtree $T(h)$ rooted at h such that$I(g)$ is disjoint from S_u. The maximality of g means that $I(parent(g))$, the leaf-cluster of g's parent, does overlap S_u. A gap under the head subject h can be interpreted as a loss of the concept h by the topic set u. In contrast, establishing a node h as a head concept can be technically referred to as a gain.

An offshoot is a leaf $i \in S_u$ which is not covered by h, i.e., $i \notin I(h)$,

Since no taxonomy perfectly reflects all of the real-world phenomena, some topic sets u may refer to general concepts that are not captured in T. In this case, two or more, rather than just one, head subjects are needed to cover them. This motivates the following definition.

The pair (T, u) will be referred to as an interpretation query. Consider a set H of nodes of T that covers the support S_u; that is, each $i \in S_u$ either belongs to H or is a descendant of a node in H, viz. $S_u \subseteq \cup_{h \in H} I(h)$. This set H is a possible result of the query (T, u). Nodes in H will be referred to as head subjects if they are interior nodes of T or offshoots if they are leaves. A node $g \in T$ is a gap for H if it is a gap for some $h \in H$. Of all the possible results H, those bearing the minimum penalty are of interest only. A minimum penalty result sometimes is referred to as a parsimonious one.

Any penalty value $p(H)$ associated with a set of head subjects H should penalize the head subjects, offshoots and gaps commensurate with the weighting of nodes in H determined from the membership values in the topic set u. We assign the head penalty to be *head*, offshoot penalty, *off*, and the gap penalty, *gap*.

To take into account the u membership values, we need to aggregate them to nodes of higher rank in T. In order to define appropriate membership values for interior nodes of tree T, we assume one of the following normalization conditions:

(P) Probabilistic condition

$$\sum_{i \in I} u(i) = 1$$

(Q) Quadratic condition

$$\sum_{i \in I} u^2(i) = 1$$

(N) No condition

$$0 \leq u(i) \leq 1$$

We observe that a crisp set $S \subseteq I$ can be considered as a fuzzy set with the non-zero membership values defined according to the normalization principle.

The three normalization conditions correspond to three possible ways of aggregating a set of individual membership values. For each interior node $t \in T$, its membership weight is defined as follows:

$$\begin{array}{l} \text{(P)}\ \ u(t) = \sum_{i \in I(t)} u(i) \\ \text{(Q)}\ \ u(t) = \sqrt{\sum_{i \in I(t)} u(i)^2} \\ \text{(N)}\ \ u(t) = \max_{i \in I(t)} u(i) \end{array} \tag{1}$$

Under each of the definitions, the weight of a gap is zero. The membership weight of the root is 1 with each of the three normalizations. In the case of a crisp set S with no condition (N), the weight of node $t \in T$ is equal to zero if $I(t)$ is disjoint from S, and it is unity, otherwise.

We now define the notion of pruned tree. Pruning the tree T at t results in the tree remaining after deleting all descendants of t. The definitions in (1) are consistent in that the weights of the remaining nodes are unchanged by any sequence of successive prunings. Note, however, that the sum of the weights assigned to the leaves in a pruned tree with normalizations (Q) and (N) is typically less than that in the original tree. With the normalization (P), it unchanges. One can notice, as well, that the decrease of the summary weight at the repeated pruning of the tree is steeper with no normalization (N).

We consider that weight $u(t)$ of node t influences not only its own contribution, but also contributions of those gaps that are children of t. Therefore, the contribution to the penalty value of each of the gaps g of a head subject $h \in T$ is weighted according to the membership weight of its parent, as defined

by $\gamma(g) = u(parent(g))$. Let us denote by $\Gamma(h)$ the set of all gaps below h. The gap contribution of h is defined as $\gamma(h) = \sum_{g \in \Gamma(h)} \gamma(g)$. For a crisp query set S with no condition, (N), this is just the number of gaps in $\Gamma(h)$.

To distinguish between proper head subjects and offshoots in H we denote the set of leaves and interior nodes in H as H^- and H^+, respectively.

Then our penalty function $p(H)$ for the tree T is defined by:

$$p(H) = head \times \sum_{h \in H^+} u(h) + gap \times \sum_{h \in H^+} \gamma(h) + off \times \sum_{h \in H^-} u(h).$$

The problem is to find such a set H that minimizes the penalty - this will be the result of the query (T, u).

2.2 Lifting Method

A preliminary step is to prune the tree T of irrelevant nodes. We then annotate all interior nodes $t \in T$ by extending the leaf membership values as in (1). Those nodes in the pruned tree that have a zero weight are gaps; they are assigned with a γ-value which is the u-weight of its parent. This can be accomplished as follows:

(a) Label with 0 all nodes t whose clusters $I(t)$ do not overlap S_u. Then remove from T all nodes that are children of 0-labeled nodes since they cannot be gaps. We note that all the elements of S_u are in the leaf set of the pruned tree, and all the other leaves of the pruned tree are labelled 0.
(b) The membership vector u is extended to all nodes of the pruned tree according to the rules in (1).
(c) Recall that $\Gamma(t)$ is the set of gaps, that is, the 0-labeled nodes of the pruned tree, and $\gamma(t) = \sum_{g \in \Gamma(t)} u(parent(g))$. We compute $\gamma(t)$ by recursively assigning $\Gamma(t)$ as the union of the Γ-sets of its children and $\gamma(t)$ as the sum of the γ-values of its children. For leaf nodes, $\Gamma(t) = \oslash$ and $\gamma(t) = 0$ if $t \in S_u$. Otherwise, i.e. if t is a gap node (or, equivalently, if t is labelled 0), $\Gamma(t) = t$ and $\gamma(t) = u(parent(t))$.

The algorithm proceeds recursively from the leaves to the root. For each node t, we compute two sets, $H(t)$ and $L(t)$, containing those nodes at which gains and losses of head subjects occur. The respective penalty is computed as $p(t)$.

I **Initialisation**
At each leaf $i \in I$: If $u(i) > 0$, define $H(i) = i$, $L(i) = \oslash$ and $p(i) = off \times u(i)$.
If $u(i) = 0$, define $H(i) = \oslash$, $L(i) = \oslash$ and $p(i) = 0$.

II **Recursion**
Consider a node $t \in T$ having a set of children W, with each child $w \in W$ assigned a pair $H(w)$, $L(w)$ and associated penalty $p(w)$. One of the following two cases must be chosen:

(a) The head subject has been gained at t, so the sets $H(w)$ and $L(w)$ at its children $w \in W$ are not relevant. Then $H(t)$, $L(t)$ and $p(t)$ are defined by: $H(t) = t$;
$L(t) = \Gamma(t)$;
$p(t) = head \times u(t) + gap \times \gamma(t)$
(b) The head subject has not been gained at t, so at t we combine the H- and L-sets as follows:

$$H(t) = \bigcup_{w \in W} H(w), L(t) = \bigcup_{w \in W} L(w) \quad \text{and} \quad p(t) = \sum_{w \in W} p(w).$$

Choose whichever of (a) and (b) has the smaller value of $p(t)$.

III **Output**: Accept the values at the root:
$H(root)$ - the heads and offshoots, $L(root)$ - the gaps, $p(root)$ - the penalty.

It is not difficult to prove that the algorithm does produce a parsimonious result.

3 An Example of Application

Table 1 presents a fuzzy cluster obtained in our project (on the data from a survey conducted in CENTRIA of Faculdade de Ciencias e Tecnologia, Universidade Nova de Lisboa (DI-FCT-UNL) in 2009) by applying our Fuzzy Additive Spectral clustering (FADDIS) algoritm [13]. This cluster is visualized with the lifting method applied at penalty parameter values displayed in Figure 3. The description of the visualization is presented in Table 2.

Table 1. A cluster of research activities undertaken in a research centre

Membership value	Code	ACM-CCS Topic
0.69911	I.5.3	Clustering
0.3512	I.5.4	Applications in I.5 PATTERN RECOGNITION
0.27438	J.2	PHYSICAL SCIENCES AND ENGINEERING (Applications in)
0.1992	I.4.9	Applications in I.4 IMAGE PROCESSING AND COMPUTER VISION
0.1992	I.4.6	Segmentation
0.19721	H.5.1	Multimedia Information Systems
0.17478	H.5.2	User Interfaces
0.17478	H.5.3	Group and Organization Interfaces
0.16689	H.1.1	Systems and Information
0.16689	I.5.1	Models in I.5 PATTERN RECOGNITION
0.14453	I.5.2	Design Methodology (Classifiers)
0.13646	H.5.0	General in H.5 INFORMATION INTERFACES AND PRESENTATION
0.13646	H.0	GENERAL in H. Information Systems
0.16513	H.1.2	User/Machine Systems

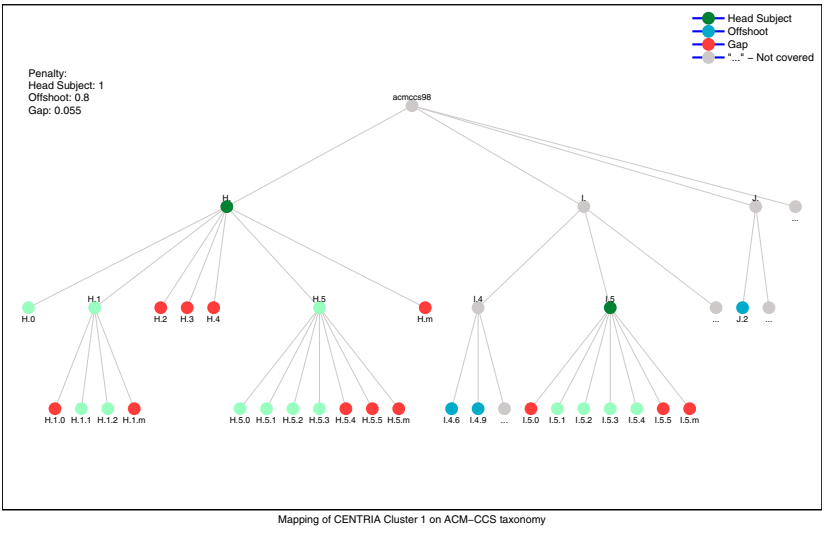

Fig. 3. Visualization of the optimal lift of the cluster in Table 1 in the ACM-CCS tree; most irrelevant leaves are not shown for the sake of simplicity

Table 2. Interpretation of the cluster with optimal lifting

	HEAD SUBJECTS
H.	Information Systems
I.5	PATTERN RECOGNITION
	OFFSHOTS
I.4.6	Segmentation
I.4.9	Applications
J.2	PHYSICAL SCIENCES AND ENGINEERING
	GAPS
H.2	DATABASE MANAGEMENT
H.3	INFORMATION STORAGE AND RETRIEVAL
H.4	INFORMATION SYSTEMS APPLICATIONS
H.5.4	Hypertext/Hypermedia
H.5.5	Sound and Music Computing
I.5.5	Implementation

4 Conclusion

The lifting method should be a useful addition to the methods for interpreting topic sets produced by various data analysis tools. Unlike the methods based on the analysis of frequencies within individual taxonomy nodes, the interpretation capabilities of this method come from an interplay between the topology of the taxonomy tree, the membership values and the penalty weights for the head subjects and associated gap and offshoot events.

On the other hand, the definition of the penalty weights remains of an issue in the method. One can think that potentially this issue can be overcome by using the maximum likelihood approach. This can happen if a taxonomy is used for visualization queries frequently – then probabilities of the gain and loss events can be assigned to each node of the tree. Using this annotation, under usual independence assumptions, the maximum likelihood criterion would inherit the additive structure of the minimum penalty criterion. Then the recursions of the lifting algorithm will remain valid, with respective changes in the criterion of course.

We can envisage, that such a development may put the issue of building the taxonomy tree onto a firm computational footing according to the structure of the flow of queries. An ideal taxonomy in an ideal world would be annotated with very contrast, one or zero probabilities, because most query topic sets would coincide with the leaf-clusters. On the contrary, the taxonomy at which the loss probabilities are similar to each other across the tree may be safely claimed unsuitable for the current query flow.

Acknowledgments

This work has been supported by the grant PTDC/EIA/69988/2006 from the Portuguese Foundation for Science & Technology. B.M. The partial financial support of the Laboratory of Choice and Analysis of Decisions at the State University – Higher School of Economics, Moscow RF, to BM is acknowledged.

References

1. ACM Computing Classification System (1998), `http://www.acm.org/about/class/1998` (Cited September 9, 2008)
2. Advanced Visual Systems (AVS), `http://www.avs.com/solutions/avs-powerviz/utility_distribution.html` (Cited November 27 2010)
3. Beneventano, D., Dahlem, N., El Haoum, S., Hahn, A., Montanari, D., Reinelt, M.: Ontology-driven semantic mapping. In: Enterprise Interoperability III, Part IV, pp. 329–341. Springer, Heidelberg (2008)
4. Buche, P., Dibie-Barthelemy, J., Ibanescu, L.: Ontology mapping using fuzzy conceptual graphs and rules. In: ICCS Supplement, pp. 17–24 (2008)
5. Cali, A., Gottlob, G., Pieris, A.: Advanced processing for ontological queries. Proceedings of the VLDB Endowment 3(1), 554–565 (2010)

6. Dotan-Cohen, D., Kasif, S., Melkman, A.: Seeing the forest for the trees: using the gene ontology to restructure hierarchical clustering. Bioinformatics 25(14), 1789–1795 (2009)
7. Gahegan, M., Agrawal, R., Jaiswal, A., Luo, J., Soon, K.-H.: A platform for visualizing and experimenting with measures of semantic similarity in ontologies and concept maps. Transactions in GIS 12(6), 713–732 (2008)
8. Ghazvinian, A., Noy, N., Musen, M.: Creating mappings for ontologies in Biomedicine: simple methods work. In: AMIA 2009 Symposium Proceedings, pp. 198–202 (2009)
9. Mansingh, G., Osei-Bryson, K.-M., Reichgelt, H.: Using ontologies to facilitate post-processing of association rules by domain experts. Information Sciences 181(3), 419–434 (2011)
10. Marinica, C., Guillet, F.: Improving post-mining of association rules with ontologies. In: The XIII International Conference Applied Stochastic Models and Data Analysis (ASMDA), pp. 76–80 (2009), ISBN 978-9955-28-463-5
11. Mazza, R.: Introduction to Information Visualization. Springer, London (2009), ISBN: 978-1-84800-218-0
12. Mirkin, B., Nascimento, S., Pereira, L.M.: Cluster-lift method for mapping research activities over a concept tree. In: Koronacki, J., Raś, Z.W., Wierzchoń, S.T., Kacprzyk, J. (eds.) Advances in Machine Learning II. SCI, vol. 263, pp. 245–257. Springer, Heidelberg (2010)
13. Mirkin, B., Nascimento, S.: Analysis of Community Structure, Affinity Data and Research Activities using Additive Fuzzy Spectral Clustering, TR-BBKCS-09-07, p. 24 (2009)
14. OWL 2 Web Ontology Language Overview (2009), `http://www.w3.org/TR/2009/RECowl2overview20091027/` (Cited November 27, 2010)
15. SNOMED CT (2011), `http://www.connectingforhealth.nhs.uk/systemsandservices/data/uktc/snomed` (Cited March 2011)
16. Sosnovsky, S., Mitrovic, A., Lee, D., Prusilovsky, P., Yudelson, M., Brusilovsky, V., Sharma, D.: Towards integration of adaptive educational systems: mapping domain models to ontologies. In: Dicheva, D., Harrer, A., Mizoguchi, R. (eds.) Procs. of 6th International Workshop on Ontologies and Semantic Web for ELearning (SWEL 2008) at ITS 2008 (2008), `http://compsci.wssu.edu/iis/swel/SWEL08/Papers/Sosnovsky.pdf`
17. Thomas, H., O'Sullivan, D., Brennan, R.: Evaluation of ontology mapping representation. In: Proceedings of the Workshop on Matching and Meaning, pp. 64–68 (2009)