Method for Generalization of Fuzzy Sets

Dmitry Frolov[1(✉)], Boris Mirkin[1,2], Susana Nascimento[3], and Trevor Fenner[2]

[1] Department of Data Analysis and Artificial Intelligence, National Research University "Higher School of Economics", Moscow, Russian Federation
dfrolov@hse.ru
[2] Department of Computer Science and Information Systems, Birkbeck University of London, London, UK
[3] Department of Computer Science and NOVA LINCS, Universidade Nova de Lisboa, Caparica, Portugal

Abstract. We define and find a most specific generalization of a fuzzy set of topics assigned to leaves of the rooted tree of a taxonomy. This generalization lifts the set to a "head subject" in the higher ranks of the taxonomy, that is supposed to "tightly" cover the query set, possibly bringing in some errors, both "gaps" and "offshoots". The method globally minimizes a penalty combining head subjects and gaps and offshoots. We apply this to extract research tendencies from a collection of about 18000 research papers published in Springer journals on data science. We consider a taxonomy of Data Science based on the Association for Computing Machinery Classification of Computing System 2012 (ACM-CCS). We find fuzzy clusters of leaf topics over the text collection and use thematic clusters' head subjects to make some comments on the tendencies of research.

Keywords: Recurrence · Generalization · Fuzzy cluster · Spectral clustering · Annotated Suffix Tree

1 Introduction

The issue of automation of structurization and interpretation of digital text collections is of ever-growing importance because of both practical needs and theoretical necessity. This paper concerns an aspect of this, the issue of generalization as a unique feature of human cognitive abilities. The existing approaches to computational analysis of structure of text collections usually involve no generalization as a specific aim. The most popular tools for structuring text collections are cluster analysis and topic modelling. Both involve features of the same level of granularity as individual words or short phrases in the texts, thus no generalization as an explicitly stated goal.

Nevertheless, the hierarchical nature of the universe of meanings is reflected in the flow of publications on text analysis. We can distinguish between at least three directions at which the matter of generalization is addressed. First of all, one should mention activities related to developing taxonomies, especially those

L. Rutkowski et al. (Eds.): ICAISC 2019, LNAI 11508, pp. 273–286, 2019.
https://doi.org/10.1007/978-3-030-20912-4_26

involving hyponymic/hypernymic relations (see, for example, [15,18], and references therein). A recent paper [16] should be mentioned here too, as that devoted to supplementing a taxonomy with newly emerging research topics.

Another direction is part of conventional activities in text summarization. Usually, summaries are created using a rather mechanistic approach of sentence extraction. There is, however, also an approach for building summaries as abstractions of texts by combining some templates such as subject-verb-object (SVO) triplets (see, for example, [8]).

Yet one more field of activities is what can be referred to as operational generalization. In this direction, the authors use generalized case descriptions involving taxonomic relations between generalized states and their parts to achieve a tangible goal such as improving characteristics of text retrieval (see, for example, [12] and [17].)

This paper falls in neither of these approaches, as we do not attempt to change any taxonomy. We rather try to use a taxonomy for straightforwardly implementing the idea of generalization. According to the Merriam-Webster dictionary, the term "generalization" refers to deriving a general conception from particulars. We assume that a most straightforward medium for such a derivation, a taxonomy of the field, is given to us. The situation of our concern is a case at which we are to generalize a fuzzy set of taxonomy leaves representing the essence of some empirically observed phenomenon. The most popular Computer Science taxonomy is manually developed by the world-wide Association for Computing Machinery, a most representative body in the domain; the latest release of the taxonomy has been published in 2012 as the ACM Computing Classification System (ACM-CCS) [1]. We take its part related to Data Science, as presented in a slightly modified form by adding a few leaves in [11]. We add a few more leaves to better reflect the research papers being analyzed [4].

The rest of the paper is organized accordingly. Section 2 presents a mathematical formalization of the generalization problem as of parsimoniously lifting of a given query fuzzy leaf set to higher ranks of the taxonomy and provides a recursive algorithm leading to a globally optimal solution to the problem. Section 3 describes an application of this approach to deriving tendencies in development of the data science, that can be discerned from a set of about 18000 research papers published by the Springer Publishers in 17 journals related to data science for the past 20 years. Its subsections describe our approach to finding and generalizing fuzzy clusters of research topics. The results are followed by our comments on the tendencies in the development of the corresponding parts of Data Science drawn from the lifting results. Section 3.6 concludes the paper.

2 Parsimoniously Lifting a Fuzzy Thematic Cluster in a Taxonomy: Model and Method

Mathematically, a taxonomy is a rooted tree whose nodes are annotated by taxonomy topics. We consider the following problem. Given a fuzzy set S of taxonomy leaves, find a node $t(S)$ of higher rank in the taxonomy, that covers

the set S as tight as possible. Such a "lifting" problem is a mathematical explication of the human facility for generalization, that is, "the process of forming a conceptual form" of a phenomenon represented, in this case, by a fuzzy leaf subset.

The problem is not as simple as it may seem to be. Consider, for the sake of simplicity, a hard set S shown with five black leaf boxes on a fragment of a tree in Fig. 1. Figure 2 illustrates the situation at which the set of black boxes is lifted to the root, which is shown by blackening the root box, and its offspring, too. If we accept that set S may be generalized by the root, this would lead to a number, four, white boxes to be covered by the root and, thus, in this way, falling in the same concept as S even as they do not belong in S. Such a situation will be referred to as a gap. Lifting with gaps should be penalized. Altogether, the number of conceptual elements introduced to generalize S here is 1 head subject, that is, the root to which we have assigned S, and the 4 gaps occurred just because of the topology of the tree, which imposes this penalty. Another lifting decision is illustrated in Fig. 3: here the set is lifted just to the root of the left branch of the tree. We can see that the number of gaps has drastically decreased, to just 1. However, another oddity emerged: a black box on the right, belonging to S but not covered by the root of the left branch at which the set S is mapped. This type of error will be referred to as an offshoot. At this lifting, three new items emerge: one head subject, one offshoot, and one gap. This is less than the number of items emerged at lifting the set to the root (one head subject and four gaps, that is, five), which makes it more preferable. Of course, this conclusion holds only if the relative weight of an offshoot is less than the total relative weight of three gaps.

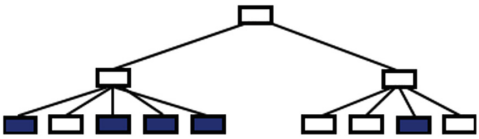

Fig. 1. A crisp query set, shown by black boxes, to be conceptualized in the taxonomy.

We are interested to see whether a fuzzy set S can be generalized by a node t from higher ranks of the taxonomy, so that S can be thought of as falling within the framework covered by the node t. The goal of finding an interpretable pigeon-hole for S within the taxonomy can be formalized as that of finding one or more "head subjects" t to cover S with the minimum number of all the elements introduced at the generalization: head subjects, gaps, and offshoots. This goal realizes the principle of Maximum Parsimony (MP) in describing the phenomenon in question.

Consider a rooted tree T representing a hierarchical taxonomy so that its nodes are annotated with key phrases signifying various concepts. We denote the set of its *leaves* by I. The relationship between nodes in the hierarchy is

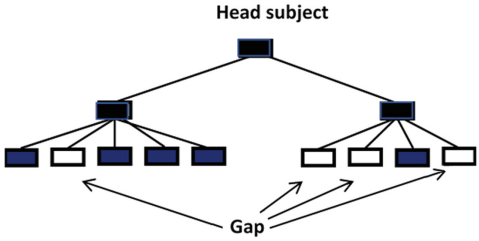

Fig. 2. Generalization of the query set from Fig. 1 by mapping it to the root, with the price of four gaps emerged at the lift.

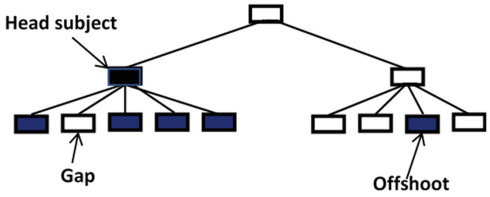

Fig. 3. Generalization of the query set from Fig. 1 by mapping it to the root of the left branch, with the price of one gap and one offshoot emerged at this lift.

conventionally expressed using genealogical terms: each node $t \in T$ is said to be the *parent* of the nodes immediately descending from t in T, its *children*. We use $\chi(t)$ to denote the set of children of t. Each *interior* node $t \in T - I$ is assumed to correspond to a concept that generalizes the topics corresponding to the leaves $I(t)$ descending from t, viz. the leaves of the subtree $T(t)$ rooted at t, which is conventionally 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 that assigns a membership value, or support, $u(i) \geq 0$ to each $i \in I$. We refer to the set $S_u \subset I$, where $S_u = \{i \in I : u(i) > 0\}$, as the *base* of u. In general, no other assumptions are made about the function u, other than, for convenience, commonly limiting it to not exceed unity. Conventional, or *crisp*, sets correspond to binary membership functions u such that $u(i) = 1$ if $i \in S_u$ and $u(i) = 0$ otherwise.

Given a fuzzy query set u defined on the leaves I of the tree T, one can consider u to be a (possibly noisy) projection of a higher rank concept, u's "head subject", onto the corresponding leaf cluster. Under this assumption, there should exist a head subject node h among the interior nodes of the tree T such that its leaf cluster $I(h)$ more or less coincides (up to small errors) with S_u. This head subject is the generalization of u to be found. The two types of possible errors associated with the head subject if it does not cover the base precisely, are false positives and false negatives, referred to in this paper, as *gaps* and *offshoots*, respectively, are illustrated in Figs. 2 and 3. Altogether, the total number of head subjects, gaps, and offshoots has to be as small as possible.

A node $t \in T$ is referred to as *u-irrelevant* if its leaf-cluster $I(t)$ is disjoint from the base S_u. Consider a candidate node h in T and its meaning relative to fuzzy set u. An *h-gap* is a node g of $T(h)$, other than h, at which a *loss* of the meaning has occurred, that is, g is a maximal u-irrelevant node in the sense that its parent is not u-irrelevant. Conversely, establishing a node h as a head subject can be considered as a *gain* of the meaning of u at the node. The set of all h-gaps will be denoted by $G(h)$. Obviously, if a node is u-irrelevant, all of its descendants are also u-irrelevant.

A gap is less significant if its parent's membership value is smaller. Therefore, a measure $v(g)$ of "gap importance" should also be defined, to be reflected in the penalty function. We suggest defining the *gap importance* as $v(g) = u(par(g))$, where $par(g)$ is the parent of g. An alternative definition would be to scale these values by dividing them by the number of children of $par(g)$. However, we note that the algorithm ParGenFS below works for any definition of gap importance. Also, we define a summary gap importance: $V(t) = \sum_{g \in G(t)} v(g)$.

An *h-offshoot* is a leaf $i \in S_u$ which is not covered by h, i.e., $i \notin I(h)$. The set of all h-offshoots is $S_u - I(h)$. Given a fuzzy topic set u over I, a set of nodes H will be referred to as a *u-cover* if: (a) H covers S_u, that is, $S_u \subseteq \bigcup_{h \in H} I(h)$, and (b) the nodes in H are unrelated, i.e. $I(h) \cap I(h') = \emptyset$ for all $h, h' \in H$ such that $h \neq h'$. The interior nodes of H will be referred to as *head subjects* and the leaf nodes as *offshoots*, so the set of offshoots in H is $H \cap I$. The set of *gaps* in H is the union of $G(h)$ over all head subjects $h \in H - I$.

We define the penalty function $p(H)$ for a u-cover H as:

$$p(H) = \sum_{h \in H - I} u(h) + \sum_{h \in H - I} \sum_{g \in G(h)} \lambda v(g) + \sum_{h \in H \cap I} \gamma u(h). \tag{1}$$

The problem we address is to find a u-cover H that globally minimizes the penalty $p(H)$. Such a u-cover will be the parsimonious generalization of the query set u.

Before applying an algorithm to minimize the total penalty, one needs to execute a preliminary transformation of the tree by pruning it from all the non-maximal u-irrelevant nodes, i.e. descendants of gaps. Simultaneously, the sets of gaps $G(t)$ and the internal summary gap importance $V(t) = \sum_{g \in G(t)} v(g)$ in Eq. (1) can be computed for each interior node t. We note that the elements of S_u are in the leaf set of the pruned tree, and the other leaves of the pruned tree are precisely the gaps. After this, our lifting algorithm ParGenFS applies. For each node t, the algorithm ParGenFS computes two sets, $H(t)$ and $L(t)$, containing those nodes in $T(t)$ at which respectively gains and losses of head subjects occur (including offshoots). The associated penalty is computed as $p(t)$ described below.

An assumption of the algorithm is that no gain can happen after a loss. Therefore, $H(t)$ and $L(t)$ are defined assuming that the head subject has not been gained (nor therefore lost) at any of t's ancestors. The algorithm ParGenFS recursively computes $H(t)$, $L(t)$ and $p(t)$ from the corresponding values for the child nodes in $\chi(t)$.

Specifically, for each leaf node that is not in S_u, we set both $L(\cdot)$ and $H(\cdot)$ to be empty and the penalty to be zero. For each leaf node that is in S_u, $L(\cdot)$ is set to be empty, whereas $H(\cdot)$, to contain just the leaf node, and the penalty is defined as its membership value multiplied by the offshoot penalty weight γ. To compute $L(t)$ and $H(t)$ for any interior node t, we analyze two possible cases: (a) when the head subject has been gained at t and (b) when the head subject has not been gained at t.

In case (a), the sets $H(\cdot)$ and $L(\cdot)$ at its children are not needed. In this case, $H(t)$, $L(t)$ and $p(t)$ are defined by:

$$H(t) = \{t\}; \quad L(t) = G(t); \quad p(t) = u(t) + \lambda V(t). \tag{2}$$

In case (b), the sets $H(t)$ and $L(t)$ are just the unions of those of its children, and $p(t)$ is the sum of their penalties:

$$H(t) = \bigcup_{w \in \chi(t)} H(w); \quad L(t) = \bigcup_{w \in \chi(t)} L(w); \quad p(t) = \sum_{w \in \chi(t)} p(w). \tag{3}$$

To obtain a parsimonious lift, whichever case gives the smaller value of $p(t)$ is chosen.

When both cases give the same values for $p(t)$, we may choose, say, (a). The output of the algorithm consists of the values at the root, namely, H – the set of head subjects and offshoots, L – the set of gaps, and p – the associated penalty.

We have proven that the algorithm ParGenFS leads to an optimal lifting indeed [4].

3 Structuring and Generalizing a Collection of Research Papers

Here are main steps of our approach:

- preparing a scholarly text collection;
- preparing a taxonomy of the domain under consideration;
- developing a matrix of relevance values between taxonomy leaf topics and research publications from the collection;
- finding fuzzy clusters according to the structure of relevance values;
- lifting the clusters over the taxonomy to conceptualize them via generalization;
- making conclusions from the generalizations.

Each of the items is covered in a separate subsection further on.

3.1 Scholarly Text Collection

Because of a generous offer from the Springer Publisher, we were able to download a collection of 17685 research papers together with their abstracts published in 17 journals related to Data Science, in our opinion, for 20 years from 1998–2017. We take the abstracts to these papers as a representative collection.

3.2 DST Taxonomy

Taxonomy is a form of knowledge engineering which is getting more and more popular. Most known are taxonomies within the bioinformatics Genome Ontology project (GO) [5], health and medicine SNOMED CT project [7] and the like. Mathematically, a taxonomy is a rooted tree, a hierarchy, whose all nodes are labeled by main concepts of a domain. The hierarchy corresponds to a relation of inclusion: the fact that node A is the parent of B means that B is part, or a special case, of A.

The subdomain of our choice is Data Science, comprising such areas as machine learning, data mining, data analysis, etc. We take that part of the ACM-CCS 2012 taxonomy, which is related to Data Science, and add a few leaves related to more recent Data Science developments. A major extract from the taxonomy of Data Science is published in [11]. The higher ranks of the taxonomy are presented in Table 1 and its full version in [4].

Table 1. ACM Computing Classification System (ACM-CCS) 2012 higher rank subjects related to Data Science.

Subject index	Subject name
1.	Theory of computation
1.1.	Theory and algorithms for application domains
2.	Mathematics of computing
2.1.	Probability and statistics
3.	Information systems
3.1.	Data management systems
3.2.	Information systems applications
3.3.	World Wide Web
3.4.	Information retrieval
4.	Human-centered computing
4.1.	Visualization
5.	Computing methodologies
5.1.	Artificial intelligence
5.2.	Machine learning

3.3 Evaluation of Relevance Between Texts and Key Phrases

Most popular and well established approaches to scoring keyphrase-to-document relevance include the so-called vector-space approach [14] and probabilistic text model approach [2]. These, however, rely on individual words and text preprocessing. We utilize a method [3,13], which requires no manual work.

An Annotated Suffix Tree (AST) is a weighted rooted tree used for storing text fragments and their frequencies. To build an AST for a text string, all suffixes from this string are extracted. A k-suffix of a string $x = x_1x_2 \dots x_N$ of length N is a continuous end fragment $x_k = x_{N-k+1}x_{N-k+2} \dots x_N$. For example, a 3-suffix of string $INFORMATION$ is substring ION, and a 5-suffix, $ATION$. Each AST node is assigned a symbol and the so-called annotation (frequency of the substring corresponding to the path from the root to the node including the symbol at the node). The root node of AST has no symbol or annotation. An algorithm for building an AST for any given string $x = x_1x_2 \dots x_N$ is described below.

1. Initialize an AST to consist of a single node, the root: T.
2. Find all the suffixes of the given string: $\{x^k = x_{N-k+1}x_{N-k+2} \dots x_N | k = 1, 2, \dots, N\}$.
3. For each suffix x^k find its maximal overlap, that is, a path from the root in T coinciding with its beginning fragment $x^{k_{max}}$. At each node of the path for $x^{k_{max}}$ add 1 to the annotation. If the length of the overlap $x^{k_{max}}$ is less than k, the path is extended by adding new nodes corresponding to symbols from the remaining part of this suffix. Annotations of all the new nodes are set to be 1.

To accelerate the working of the method, one should use efficient versions of algorithms utilising suffix trees and suffix arrays (see, for example, [6]).

Having an AST T built, we can score the string-to-document relevance over the AST. To do this, we follow [10] by computing the conditional probability of node u in T:

$$p(u) = \frac{f(u)}{f(parent(u))}. \tag{4}$$

For all the immediate offspring of the root (R), formula has the following form:

$$p(u) = \frac{f(u)}{\sum\limits_{v \in T: parent(v) = R} f(v)}, \tag{5}$$

where $f(u)$ is the frequency annotation of the node u. Using the formula above, one can calculate the probability of node u relative to all its siblings. For each suffix x_k of string x the relevance score $s(x_k, T)$ is defined as:

$$s(x_k, T) = \frac{1}{k_{max}} \sum_{i=1}^{k_{max}} p(x_i^k). \tag{6}$$

The AST relevance score of string x and text T is defined as the mean of all the suffix scores:

$$S(x, T) = \frac{1}{N} \sum_{k=1}^{N} s(x_k, T). \tag{7}$$

In practical computations, we split any document into a set of strings (usually consisting of 2–3 consecutive words), create an empty AST for the document and add these strings in the AST in sequence, by using the algorithm above.

To lessen the effects of frequently occurring general terms, the scoring function is modified by five-fold decreasing the weight of stop-words. The list of stop-words includes: "learning, analysis, data, method" and a few postfixes: "s/es, ing, tion". After an AST for a document has been built, the time complexity of calculating the string-to-document relevance score is $O(m^2)$ where m is the length of the query string. This does not depend on the document length, in contrast to the popular Levenshtein-distance based approaches.

3.4 Defining and Computing Fuzzy Clusters of Taxonomy Topics

Clusters of topics should reflect co-occurrence of topics: the greater the number of texts to which both topics t and t' are relevant, the greater the interrelation between t and t', the greater the chance for topics t and t' to fall in the same cluster. We have tried several popular clustering algorithms. Unfortunately, no satisfactory results have been found. Therefore, we present here results obtained with the FADDIS algorithm from [10] developed specifically for finding thematic clusters. This algorithm implements assumptions that are relevant to the task:

LN Laplacian Normalization: Similarity data transformation modeling – to an extent – heat distribution and, in this way, making the cluster structure sharper.

AA Additivity: Thematic clusters behind the texts are additive so that similarity values are sums of contributions by different hidden themes.

AN Non-Completeness: Clusters do not necessarily cover all the key phrases available as the text collection under consideration may be irrelevant to some of them.

Co-relevance Topic-to-Topic Similarity Score. Given a keyphrase-to-document matrix R of relevance scores, it is converted to a keyphrase-to-keyphrase similarity matrix A or scoring the "co-relevance" of keyphrases according to the text collection structure. The similarity score $a_{tt'}$ between topics t and t' can be computed as the inner product of vectors of scores $r_t = (r_{tv})$ and $r_{t'} = (r_{t'v})$ where $v = 1, 2, \ldots, V = 17685$. The inner product is moderated by a natural weighting factor assigned to texts in the collection. The weight of text v is defined as the ratio of the number of topics n_v relevant to it and n_{max}, the maximum n_v over all v = 1, 2, ..., V. A topic is considered relevant to v if its relevance score is greater than 0.2 (a threshold found experimentally, see [3]).

Additive Fuzzy Spectral Clustering. Let us denote the total set of leaf topics by T and assume that a fuzzy cluster over T is represented by a fuzzy membership vector $\boldsymbol{u} = (u_t)$, $t \in T$, such that $0 \leq u_t \leq 1$ for all $t \in T$, and an intensity $\mu > 0$, a scale coefficient to relate the membership scores to the

similarity scores. For T being a set of research topics and $\boldsymbol{u} = (u_t)$, $t \in T$, a membership values vector representing the a semantic substructure of a corpus of research papers under consideration, the product $(\mu u_t)(\mu u_{t'}) = \mu^2 u_t u_{t'}$ can be considered as the contribution by the research direction represented by the cluster under consideration to the total similarity score $a_{tt'}$ between topics t and t'. The additive fuzzy clustering model in [10] states that the entries in the topic-to-topic similarity matrix A can be considered as resulting from additive contributions of K fuzzy clusters, up to small errors to be minimized:

$$a_{tt'} = \sum_{k=1}^{K} \mu_k^2 u_{kt} u_{kt'} + e_{tt'}, \tag{8}$$

where $\boldsymbol{u}_k = (u_{kt})$ is the membership vector of cluster k, and μ_k its intensity. These assumptions require that clusters are extracted according to an additive model. A method developed in [10], FADDIS, finds clusters in (8) one-by-one, which accords with the assumptions above. Paper [10] provides some theoretical and experimental computation results to demonstrate that FADDIS is competitive over other fuzzy clustering approaches.

To make the hidden cluster structure in similarity data sharper, we apply the so-called Laplacian normalization [9].

FADDIS Thematic Clusters. After computing the 317×317 topic-to-topic co-relevance matrix, converting in to a topic-to-topic Lapin transformed similarity matrix, and applying FADDIS clustering, we sequentially obtained 6 clusters, of which three clusters seem especially homogeneous. We denote them using letters L, for 'Learning'; R, for 'Retrieval'; and C, for 'Clustering'. These clusters are presented in Table 2.

Table 2. Clusters L, R, C: topics with largest membership values.

Cluster L		Cluster R		Cluster C	
$u(t)$	Topic	$u(t)$	Topic	$u(t)$	Topic
0.300	Rule learning	0.211	Query representation	0.327	Biclustering
0.282	Batch learning	0.207	Image representations	0.286	Fuzzy clustering
0.276	Learning to rank	0.194	Shape representations	0.248	Consensus clustering
0.217	Query learning	0.194	Tensor representation	0.220	Conceptual clustering
0.216	Apprenticeship learning	0.191	Fuzzy representation	0.192	Spectral clustering
0.213	Models of learning	0.187	Data provenance	0.187	Massive data clustering
0.203	Adversarial learning	0.173	Equational models	0.159	Graph based conceptual clustering

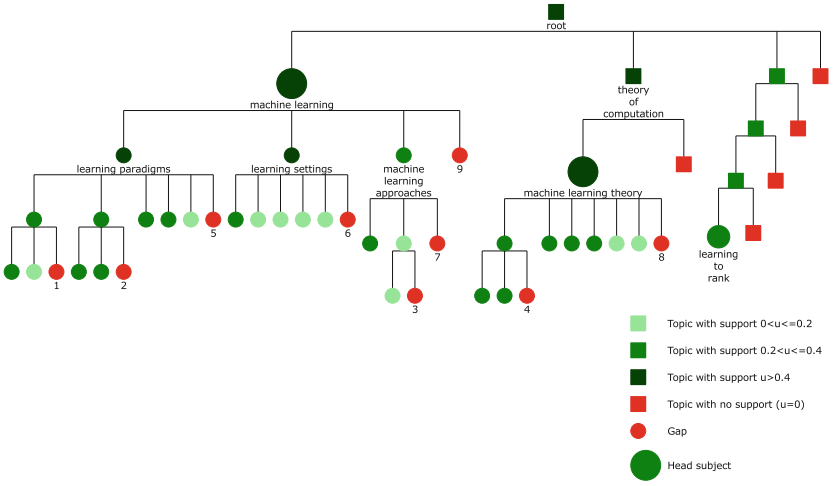

Fig. 4. Lifting results for Cluster L: Learning. Gaps are numbered, see Table 3.

3.5 Results of Lifting Clusters L, R, and C Within DST

All obtained clusters are lifted in the DST taxonomy using ParGenFS algorithm with the gap penalty $\lambda = 0.1$ and off-shoot penalty $\gamma = 0.9$.

The results of lifting Cluster L are shown in Fig. 4. The cluster has received three head subjects: machine learning, machine learning theory, and learning to rank. These represent the structure of the general concept "Learning" according

Table 3. Gaps at the lifting of Cluster L

Number	Topics
1	Ranking, supervised learning by classification, structured outputs
2	Sequential decision making in practice, inverse reinforcement learning in practice
3	Statistical relational learning
4	Sequential decision making, inverse reinforcement learning
5	Unsupervised learning
6	Learning from demonstrations, kernel approach
7	Classification and regression trees, kernel methods, neural networks, learning in probabilistic graphical models, learning linear models, factorization methods, markov decision processes, stochastic games, learning latent representations, multiresolution, support vector machines
8	Sample complexity and generalization bounds, Boolean function learning, kernel methods, boosting, bayesian analysis, inductive inference, structured prediction, markov decision processes, regret bounds
9	Machine learning algorithms

to our text collection. The list of gaps obtained is less instructive, reflecting probably a relatively modest coverage of the domain by the publications in the collection (see in Table 3).

Similar comments can be made with respect to results of lifting of Cluster R: Retrieval. The obtained head subjects: Information Systems and Computer Vision show the structure of "Retrieval" in the set of publications under considerations. Lifting of Cluster C leads to much fragmentary results. There are 16 (!) head subjects here: clustering, graph based conceptual clustering, trajectory clustering, clustering and classification, unsupervised learning and clustering, spectral methods, document filtering, language models, music retrieval, collaborative search, database views, stream management, database recovery, mapreduce languages, logic and databases, language resources. As one can see, the core clustering subjects are supplemented by methods and environments in the cluster – this shows that the ever increasing role of clustering activities perhaps should be better reflected in the taxonomy.

3.6 Making Conclusions

We can see that the topic clusters found with the text collection do highlight areas of soon-to-be developments. Three clusters under consideration closely relate, in respect, to the following processes:

- theoretical and methodical research in learning, as well as merging the subject of learning to rank within the mainstream;
- representation of various types of data for information retrieval, and merging that with visual data and their semantics; and
- various types of clustering in different branches of the taxonomy related to various applications and instruments.

In particular, one can see from the "Learning" head subjects (see Fig. 4 and comments to it) that main work here still concentrates on theory and method rather than applications. A good news is that the field of learning, formerly focused mostly on tasks of learning subsets and partitions, is expanding currently towards learning of ranks and rankings. Of course, there remain many sub-areas to be covered: these can be seen in and around the list of gaps in Table 3.

Moving to the lifting results for the information retrieval cluster R, we can clearly see the tendencies of the contemporary stage of the process. Rather than relating the term "information" to texts only, as it was in the previous stages of the process of digitalization, visuals are becoming parts of the concept of information. There is a catch, however. Unlike the multilevel granularity of meanings in texts, developed during millennia of the process of communication via languages in the humankind, there is no comparable hierarchy of meanings for images. One may only guess that the elements of the R cluster related to segmentation of images and videos, as well as those related to data management systems, are those that are going to be put in the base of a future multilevel system of meanings for images and videos.

Regarding the "clustering" cluster C with its 16 (!) head subjects, one may conclude that, perhaps, a time moment has come or is to come real soon, when the subject of clustering must be raised to a higher level in the taxonomy to embrace all these "heads". At the beginning of the Data Science era, a few decades ago, clustering was usually considered a more-or-less auxiliary part of machine learning, the unsupervised learning. Perhaps, soon we are going to see a new taxonomy of Data Science, in which clustering is not just an auxiliary instrument but rather a model of empirical classification, a big part of the knowledge engineering. When discussing the role of classification as a knowledge engineering phenomenon, one encounters three conventional aspects of classification:

– structuring the phenomena;
– relating different aspects of phenomena to each other;
– shaping and keeping knowledge of phenomena.

Each of them can make a separate direction of research in knowledge engineering.

References

1. The 2012 ACM Computing Classification System. http://www.acm.org/about/class/2012. Accessed 30 Apr 2018
2. Blei, D.: Probabilistic topic models. Commun. ACM **55**(4), 77–84 (2012)
3. Chernyak, E.: An approach to the problem of annotation of research publications. In: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, pp. 429–434. ACM (2015)
4. Frolov, D., Mirkin, B., Nascimento, S., Fenner, T.: Finding an appropriate generalization for a fuzzy thematic set in taxonomy. Working paper WP7/2018/04, Moscow, Higher School of Economics Publ. House (2018)
5. Gene Ontology Consortium: Gene ontology consortium: going forward. Nucleic Acids Res. **43**, D1049–D1056 (2015)
6. Grossi, R., Vitter, J.S.: Compressed suffix arrays and suffix trees with applications to text indexing and string matching. SIAM J. Comput. **35**(2), 378–407 (2005)
7. Lee, D., Cornet, R., Lau, F., De Keizer, N.: A survey of SNOMED CT implementations. J. Biomed. Inform. **46**(1), 87–96 (2013)
8. Lloret, E., Boldrini, E., Vodolazova, T., MartÃnez-Barco, P., Munoz, R., Palomar, M.: A novel concept-level approach for ultra-concise opinion summarization. Expert Syst. Appl. **42**(20), 7148–7156 (2015)
9. Mirkin, B., Nascimento, S.: Additive spectral method for fuzzy cluster analysis of similarity data including community structure and affinity matrices. Inf. Sci. **183**(1), 16–34 (2012)
10. Mirkin, B.: Clustering: A Data Recovery Approach. Chapman and Hall/CRC Press, Boca Raton (2012)
11. Mirkin, B., Orlov, M.: Three aspects of the research impact by a scientist: measurement methods and an empirical evaluation. In: Migdalas, A., Karakitsiou, A. (eds.) Optimization, Control, and Applications in the Information Age. PROMS, vol. 130, pp. 233–259. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-18567-5_12
12. Mueller, G., Bergmann, R.: Generalization of workflows in process-oriented case-based reasoning. In: FLAIRS Conference, pp. 391–396 (2015)

13. Pampapathi, R., Mirkin, B., Levene, M.: A suffix tree approach to anti-spam email filtering. Mach. Learn. **65**(1), 309–338 (2006)
14. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Inf. Process. Manage. **25**(5), 513–523 (1998)
15. Song, Y., Liu, S., Wang, H., Wang, Z., Li, H.: Automatic taxonomy construction from keywords. U.S. Patent No. 9,501,569. U.S. Patent and Trademark Office, Washington, D.C. (2016)
16. Vedula, N., Nicholson, P.K., Ajwani, D., Dutta, S., Sala, A., Parthasarathy, S.: Enriching taxonomies with functional domain knowledge. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 745–754. ACM (2018)
17. Waitelonis, J., Exeler, C., Sack, H.: Linked data enabled generalized vector space model to improve document retrieval. In: Proceedings of NLP & DBpedia 2015 Workshop in Conjunction with 14th International Semantic Web Conference (ISWC), vol. 1486. CEUR-WS (2015)
18. Wang, C., He, X., Zhou, A.: A short survey on taxonomy learning from text corpora: issues, resources and recent advances. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 1190–1203 (2017)