Parsimonious Generalization of Fuzzy Thematic Sets in Taxonomies Applied to the Analysis of Tendencies of Research in Data Science

Dmitry Frolov[a], Susana Nascimento[b], Trevor Fenner[c], Boris Mirkin[a,c,*]

[a]*Department of Data Analysis and Artificial Intelligence, National Research University Higher School of Economics, Pokrovsky Boulevard, 11, Moscow, Russian Federation*
[b]*Department of Computer Science and NOVA LINCS, Universidade Nova de Lisboa, Quinta da Torre 2829-516, Caparica, Portugal*
[c]*Department of Computer Science and Information Systems, Birkbeck University of London, Malet Street, London WC1E 7HX, UK*

Abstract

This paper proposes a novel method, referred to as ParGenFS, for finding a most specific generalization of a query set represented by a fuzzy set of topics assigned to leaves of the rooted tree of a taxonomy. The query set is generalized by "lifting" it to one or more "head subjects" in the higher ranks of the taxonomy. The head subjects should cover the query set, with the possible addition of some "gaps", taxonomy nodes covered by the head subject but irrelevant to the query set. To decrease the numbers of gaps, we admit some "offshoots", nodes belonging to the query set but not covered by the head subject. The method globally minimizes the total number of head subjects, gaps and offshoots, each suitably weighted. Our algorithm is applied to the structural analysis and description of a collection of 17685 abstracts of research papers published in 17 Springer journals related to Data Science for the 20-year period 1998-2017. Our taxonomy of Data Science (TDS) is extracted from the Association for Computing Machinery Computing Classification System 2012 (ACM-CCS), a six-level hierarchical taxonomy manually developed by a team of ACM experts. The TDS also includes a number of additional leaves that we added to cater for recent developments not represented in the ACM-CCS taxonomy. We find fuzzy clusters of leaf topics over the text collection, using specially developed machinery. Three of the clusters are indeed thematic, relating to the Data Science sub-areas of (a) learning, (b) information retrieval, and (c) clustering. These three clusters are then lifted in the TDS using

*Corresponding author
Email addresses: dfrolov@hse.ru (Dmitry Frolov), snt@fct.unl.pt (Susana Nascimento), trevor@dcs.bbk.ac.uk (Trevor Fenner), bmirkin@hse.ru (Boris Mirkin)

ParGenFS, which allows us to draw some conclusions about tendencies in developments in these areas.

Keywords: hierarchical taxonomy, parsimony, generalization, additive fuzzy cluster, spectral clustering, annotated suffix tree

---

1. Introduction

The issue of the automation of structuring 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. Among various meanings of the term, we focus on two definitions from the Merriam-Webster dictionary: the verb "generalize" means "to give a general form to" (1) or "to derive or induce (a general conception or principle) from particulars" (2a) (see [23]).

The existing approaches to computational analysis of structure of text collections do not usually involve generalization as a specific goal. The most popular tools for structuring text collections are clustering and topic modelling. Both operate with features at the same level of granularity as individual words or short phrases in the texts, and thus do not have generalization as an explicitly stated goal.

Nevertheless, publications on text analysis frequently point to the hierarchical nature of the universe of concepts and meanings, thus somehow involving generalization. One can distinguish between at least three directions in which the matter of generalization is addressed, be it more or less explicitly:

(i) Activities related to developing taxonomies, especially those involving what are referred to in linguistics as hyponymy/hypernymy relations: a hyponym is a concept whose semantic field is part of the semantic field of another concept, its hypernym (see, for example, [38, 39, 44], and references therein). A recent paper [41] should also be mentioned here, as it is devoted to supplementing a taxonomy with newly emerging research topics.

(ii) 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, [21, 30]).

(iii) "Operational" generalization. This refers to using generalized case descriptions involving taxonomic relations between generalized states and their parts to achieve tangible goals, such as improving the characteristics of text retrieval (see, for example, [29] and [42].)

We, however, neither develop nor change taxonomies. Rather, we use a taxonomy for straightforwardly deriving a general conception within the taxonomy from particulars that are also parts of the taxonomy, although of a greater granularity. We assume that a straightforward medium for such a derivation, a rooted tree of a taxonomy of the field, is available. The tree represents a main hyponymic/hypernymic relation in the domain, so that an A-tagged node is the parent of a B-tagged node if the relation "B is an A" is true. We are concerned with a case in which we wish to generalize a fuzzy set of leaves of the taxonomy that represent the essence of some empirically observed phenomenon. Specifically, one may be interested in patterns of novel research in a domain like Data Science, a newly emerging area of Computer Science. The most popular Computer Science taxonomy was manually developed by the world-wide Association for Computing Machinery, an authoritative representative body in the field. The latest release of this taxonomy was published in 2012 as the ACM Computing Classification System (ACM-CCS) [1]. We consider the part related to Data Science, in a slightly modified form obtained by adding in a few leaves, as described in [13]; see also a somewhat reduced version in [27].

Information on research being conducted in Data Science is taken from a collection of research papers published relatively recently in a number of journals representative of the domain of our concern. We extract tight clusters of ACM-CCS topics using the papers in the collection, each representing core tendencies of the development of the domain as reflected in the collection. It should be expected that the clusters are fuzzy, in accordance with the fuzzy nature of semantics. We are interested in finding a most appropriate generalization of such a cluster, following an approach illustrated in Figure 1(a).

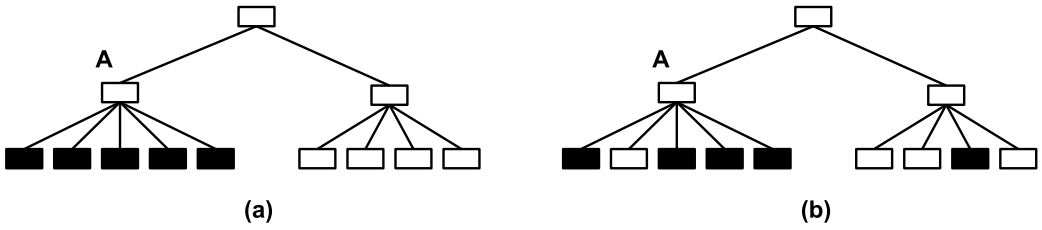

Figure 1: A taxonomy fragment with black boxes corresponding to a query set; a straightforward case (a) and a more complex case (b).

Figure 1(a) depicts a fragment of a taxonomy, with a query represented by a leaf cluster comprising all the children of the parental node A. Of course, the taxonomy concept assigned to A is a most natural generalization of the cluster, in this case suggesting an interpretation of the cluster as all those topics that fall in the concept A. We then extend this approach to generalize less obvious cases,

such as that in Figure 1(b), and consider how this approach can be applied to real-world scenarios.

The rest of the paper is organized accordingly. In Section 2, we present a mathematical formalization of the generalization problem as one of parsimoniously lifting a fuzzy set of leaves representing a given query to higher ranks of the taxonomy. We then provide a recursive algorithm that finds a globally optimal solution to the problem. In Section 3, we describe an application of this approach to deriving tendencies in the development of Data Science; these are discerned from the abstracts of the 17685 research papers published over a recent 20-year period in 17 Springer journals related to Data Science. After a brief description of the state of the art in the analysis of research papers, we describe our approach to finding and generalizing fuzzy clusters of research topics. Specifically, the taxonomy of Data Science (TDS) used in this paper is presented in subsection 3.2.2; a method for developing a matrix of topic-to-paper relevance values based on automated processing of the text collection is described in subsection 3.2.3; in subsections 3.2.4, 3.2.5 and 3.2.6, we describe a spectral method for finding fuzzy clusters of research topics from TDS using relevance and co-relevance data; the three most homogeneous of six fuzzy clusters found are presented in section 3.2.7; in subsection 3.2.8, we present the results of lifting these clusters in the TDS taxonomy; then, in subsection 3.2.9, we present our conclusions on the tendencies in the development of the corresponding parts of Data Science suggested by the lifting results. Section 4 concludes the paper.

2. Parsimoniously lifting a fuzzy leaf set in a taxonomy: model and method

2.1. Statement of the problem

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 $h(S)$ of higher rank in the taxonomy that covers the set S as tightly as possible. Such a "lifting" problem is a mathematical analogue 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 subset of leaves.

The problem is not as simple as it may seem to be. Consider, for the sake of simplicity, a crisp set S represented by the five black box leaves on the fragment of a tree shown in Figure 1(b). Figure 2 illustrates the situation in which the set of black box leaves is lifted to the root, which is shown by blackening the box at the root, and also its children. If we accept that set S may be generalized by the *head subject* at the root, this would lead to a number of white boxes (here four) being covered by the root, and thus being included in the same concept as S, even

though they do not belong in S. In such a situation, these (four) nodes will be referred to as *gaps*. Any gap should incur a "penalty", as well as there being a "charge" for any introduced head subject. Altogether, the number of conceptual elements introduced to generalize S here is one head subject, that is, the root to which we have assigned S, and the four resulting gaps. An alternative lifting decision is illustrated in Figure 3: here the set S 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 one. However, another anomaly has emerged: a black box leaf on the right, a node belonging to S but not covered by the root of the left branch to which the set S is mapped. This type of error will be referred to as an *offshoot*. With this lifting, only three items emerge: one head subject, one offshoot, and one gap. This is fewer than the number of items introduced by lifting S to the root of the tree (viz. five, one head subject and four gaps), which appears to make the former preferable. Of course, this conclusion is only valid if the weight of an offshoot is less than the total weight of three gaps.

Therefore, when lifting a leaf set to higher ranks in the taxonomy, there may emerge two types of errors: gaps and offshoots. These are completely determined by the choice of head subject. If one likens assigning a general concept to the process of classification, then gaps correspond to false positives and offshoots to false negatives.

The goal of finding a most appropriate generalization for S within the taxonomy can be formalized as that of finding one or more head subjects that cover S with the minimum number of elements introduced, viz. head subjects, gaps and offshoots. This goal realizes the principle of Maximum Parsimony (MP) in describing the phenomenon in question. This principle expresses a general idea, called Occam's razor, that the simplest explanation of the observations should be preferred [33].

Consider a rooted tree T representing a hierarchical taxonomy in which the 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 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 from I to the non-negative real numbers that assigns a membership value $u(i)$ to each $i \in I$. We refer to the set $S_u \subseteq I$, 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 that we seek. 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; these are illustrated in Figures 2 and 3. We are interested in finding such a generalization of a given fuzzy u in which the total number of head subjects, gaps and offshoots is as small as possible. To this end, we introduce penalties for each of these elements. Assuming, for the sake of simplicity, that the black box leaves in Figure 1(b) have membership values equal to unity, one can easily see that the total penalty when the head subject is raised to the root (Figure 2) is equal to $1 + 4\lambda$, where 1 is the penalty for a head subject and λ is the penalty for a gap, since the lift in Figure 2 involves one head subject, the root, and four gaps, the white box leaves. Similarly, the penalty for the lift to the root of the left-hand subtree (Figure 3) is equal to $1 + \gamma + \lambda$, where γ is the penalty for an offshoot, as there is exactly one head subject, one gap and one offshoot in Figure 3. Therefore, depending on the values of γ and λ, either the lift in Figure 2 or that in Figure 3 should be chosen. This will be the former if $3\lambda < \gamma$, or the latter otherwise.

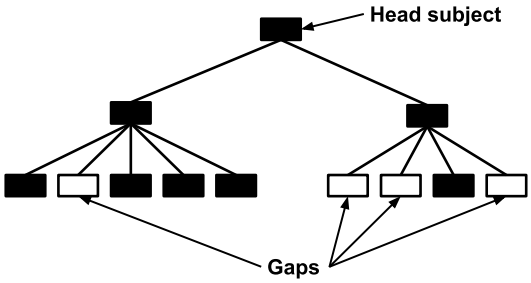

Figure 2: Generalization of the query set from Figure 1(b) by mapping it to the root, at the cost of four gaps.

To properly define the concept of gap in general, let us first consider *u-irrelevant* nodes in the tree T. A node $t \in T$ is referred to as *u-irrelevant* if its leaf-cluster $I(t)$ is disjoint from the base S_u. In other words, a node is u-irrelevant if all of its descendants are u-irrelevant too.

Consider a candidate head subject node h in T. As mentioned above, establishing node h as a head subject can be considered as a *gain* of the meaning of u at the node. An *h-gap* is a node g of $T(h)$, other than h, at which a *loss* of the

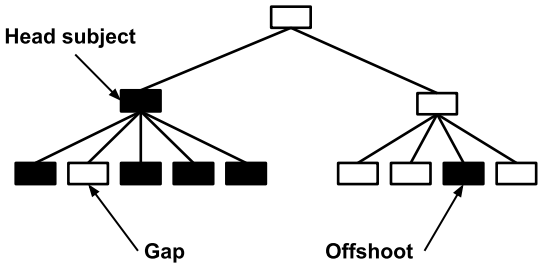

Figure 3: Generalization of the query set from Figure 1(b) by mapping it to the root of the left branch, at the cost of one gap and one offshoot.

meaning has occurred, that is, g is a maximal u-irrelevant node in the sense that its parent is not u-irrelevant. The set of all h-gaps will be denoted by $G(h)$.

An *h-offshoot* is a leaf $i \in S_u$ that is not covered by h, i.e., $i \notin I(h)$. The set of all h-offshoots is $S_u - I(h)$.

Some fuzzy topic sets u may refer to general concepts widely dispersed in the taxonomy. In such a case, defining just one head subject may be insufficient. Therefore, we permit cases in which two or more head subjects are needed to cover S_u accurately.

The concepts introduced motivate the following definition.

Given a fuzzy topic set u on 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. $I(h) \cap I(h') = \varnothing$ 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 associate a penalty with any u-cover H that takes into account the relative importance of head subjects, gaps and offshoots, using the following penalty weights: 1 for a head subject, λ for a gap, and γ for an offshoot.

To properly define the overall penalty, we extend the u-membership values from I to all the nodes in T. The algorithm ParGenFS (Parsimonious Generalization of Fuzzy Sets) for finding a parsimonious generalization of u, which we describe below, does not depend on the way the u-values are assigned, so any extension of the membership values to $u(t)$ for $t \in T - I$ is acceptable, provided the value of $u(t)$ is zero for all u-irrelevant nodes t.

Although every gap is assigned with a membership value of zero, we may consider some gaps more important than others, depending on the membership values assigned to their parents. 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 and be reflected in the penalty function. We assume here that the *gap*

importance $v(g)$ of a gap g is $u(par(g))$, where $par(g)$ is the parent of g.

The penalty function $p(H)$ for a u-cover H is then defined by:

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

These concepts are illustrated in Fig. 4.

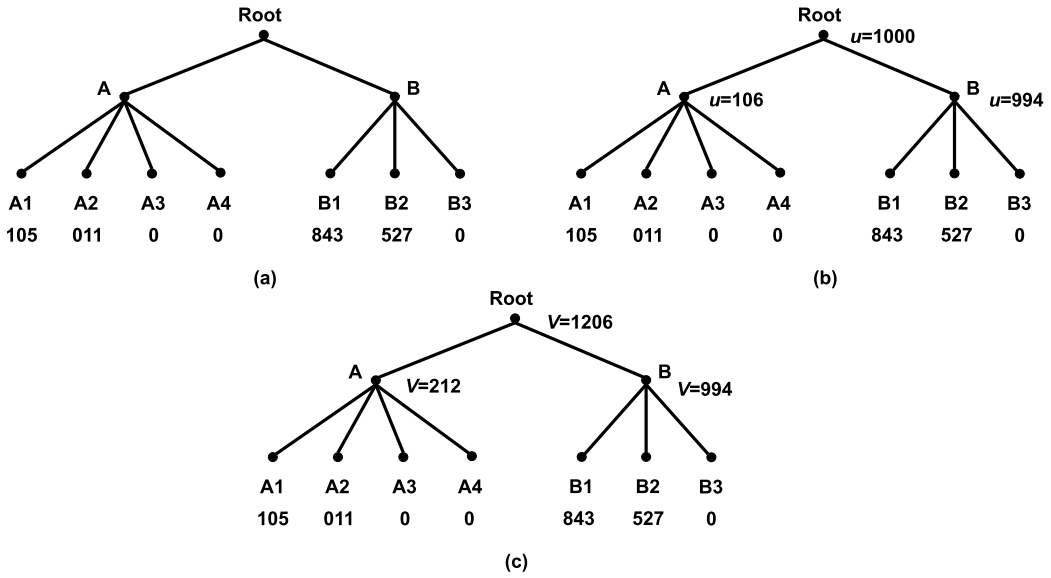

Figure 4: Illustration of concepts related to fuzzy leaf sets in a taxonomy: (a) taxonomy T with seven leaves A1, A2, A3, A4, B1, B2, B3 of which four, A1, A2, B1, B2 have positive membership values presented below in thousandths; they are normalized so that their squares total to unity; (b) T after extending the membership values to the interior nodes A, B, and Root with the same normalization: the sum of the squares of the membership values of the nodes at the same level of the hierarchy is unity; (c) summary gap importance values $V(\cdot)$.

Figure 4(a) presents a taxonomy with two interior nodes A and B, as well as the root, with a fuzzy query set u defined on the leaves. The membership values are given in thousandths and are normalized so that the sum of the squares of the membership values is unity (up to an admissible error). The normalization accords with the quadratic normalization of fuzzy clusters used in the algorithm FADDIS (described later, see section 3.2.4). Figure 4(b) extends the membership values to the interior nodes, so that 106 at A and 994 at B are close to the square roots of the sum of the squares of the membership values of the leaves covered by the corresponding node; i.e., $\sqrt{105^2 + 11^2} \approx 106$ and $\sqrt{843^2 + 527^2} \approx 994$. Figure 4(c) takes account of the gap importance values $v(t)$ and shows the summary gap importances at the interior nodes. Specifically, each potential gap g in function

(1) gets a v-value coinciding with the membership value of its parent: $v(A3) = v(A4) = 106$, $v(B3) = 994$. The V-value at a parental node t is calculated as the sum of the v-values of the gap leaves in $I(t)$. We consider this value as the *summary gap importance* at t, and denote it by $V(t)$; so $V(A) = 106 + 106 = 212$, $V(B) = 994$, and $V(Root) = 212 + 994 = 1206$.

2.2. Algorithm ParGenFS for finding a most specific generalization

Here we propose an algorithm for finding a u-cover H that globally minimizes the penalty $p(H)$. Such a u-cover will be a parsimonious generalization of the query set u.

Our algorithm is applied to the taxonomy tree after it has been preprocessed by pruning from it all non-maximal u-irrelevant nodes, i.e. descendants of gaps. Simultaneously, the sets of gaps $G(t)$ and the summary gap importance values, $V(t) = \sum_{g \in G(t)} v(g)$ in formula (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.

We assume that the tree T has already been pruned and that all of its nodes are annotated with the membership values $u(t)$. The sets $G(t)$, together with the gap importance values $v(t)$ and $V(t)$, are assigned as described above. This is shown for the example in Figure 5 in the next subsection.

We can now apply our lifting algorithm ParGenFS. For each node t, the algorithm ParGenFS computes two sets, $H(t)$ and $L(t)$, containing those nodes in $T(t)$ at which gains and losses, respectively, of head subjects occur (including offshoots). The associated penalty $p(t)$ is computed as described below.

An assumption of the algorithm is that no gain can happen after a loss, which is a consequence of the fact that all of the descendants of a gap are u-irrelevant. 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 $p(t)$ for each node t from the corresponding values for its children, the nodes in $\chi(t)$.

Specifically, for each leaf node i that is not in S_u, we set both $L(i)$ and $H(i)$ to be empty and the penalty $p(t)$ to be zero. For each leaf node $i \in S_u$, we set $L(i)$ to be empty, $H(i)$ to be the set $\{i\}$ containing just the leaf node, and the penalty $p(i)$ to be γu_i.

To compute $L(t)$ and $H(t)$ for any interior node t, we distinguish between 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:

$$\begin{aligned} H(t) &= \{t\} \\ L(t) &= G(t) \\ p(t) &= u(t) + \lambda V(t). \end{aligned} \tag{2}$$

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

$$\begin{aligned} H(t) &= \bigcup_{w \in \chi(t)} H(w) \\ L(t) &= \bigcup_{w \in \chi(t)} L(w) \\ p(t) &= \sum_{w \in \chi(t)} p(w). \end{aligned} \tag{3}$$

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 arbitrarily – in the formulation of the algorithm below, we always choose (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.

Algorithm ParGenFS

- **INPUT:** u, T
- **OUTPUT:** $H = H(Root)$, $L = L(Root)$, $p = p(Root)$

I **Base Case**

for each leaf $i \in I$

$L(i) = \oslash$

if $u(i) > 0$

$H(i) = \{i\}$

$p(i) = \gamma u(i)$

else

$H(i) = \oslash$

$p(i) = 0$

II **Recursion**

$p_{gain} = u(t) + \lambda V(t)$

$p_{nogain} = \sum_{w \in \chi(t)} p(w)$

if $p_{gain} \leq p_{nogain}$

$H(t) = \{t\}$

$L(t) = G(t)$

$p(t) = p_{gain}$

else

$H(t) = \bigcup_{w \in \chi(t)} H(w)$

$L(t) = \bigcup_{w \in \chi(t)} L(w)$

$p(t) = p_{nogain}$

It is not difficult to see that the algorithm ParGenFS does lead to an optimal lifting, as is shown in the following theorem.

Theorem 1. *Any u-cover H found by the algorithm ParGenFS is a (global) minimizer of the penalty p(Root).*

Proof of Theorem 1. We prove this result by induction on the number of nodes n in the tree. If $n = 1$, there is only one node i and, in the Base Case of ParGenFS, the definition of the sets $H(i)$ and $L(i)$ is such that the only possible non-empty set is $H(i) = \{i\}$, when $i \in S_u$. The penalty in this case is $\gamma u(i)$, which is clearly the correct minimum penalty. When $i \notin S_u$, the penalty is obviously zero.

Let us now assume that the statement is true for all rooted trees with fewer than n nodes. Consider a rooted tree $T(t)$ with n nodes, where $n > 1$. Each child w of the root t is itself the root of a subtree $T(w)$ with fewer than n nodes. So, by the induction hypothesis, $H(w)$ and $L(w)$ are the optimal sets for $T(w)$, and $p(w)$ is the minimal penalty.

If the head subject is not gained at t, then the optimal H- and L-sets at t are clearly the unions of the corresponding sets for the subtrees $T(w)$; this follows from the additive structure of the penalty function in eqn. (1). If, however, the head subject is gained at t, then $H(t) = \{t\}$ and $L(t) = G(t)$. Therefore, the minimum penalty for the subtree $T(t)$ must be the smaller of the penalty values $p(t) = u(t) + \lambda V(t)$ and $p(t) = \sum_{w \in \chi(t)} p(w)$, as determined in the algorithm. The result now follows by induction on n. □

A few words on complexity issues. Computationally, algorithm ParGenFS is rather straightforward. It processes each node of the taxonomy tree only once. At

each node it computes simple arithmetic expressions and unions of node subsets, which may take just a constant time or, at most, is proportional to the number of nodes. Therefore, the total number of operations is bounded above by a quadratic function of the number of leaves. We used a conventional Python 3.5 environment for the computations. All visualizations were obtained with the help of the ETE toolkit for Python (*http://etetoolkit.org/*).

2.3. An illustrative example

We now apply the algorithm ParGenFS to the taxonomy and query set shown in Figure 5(a). This shows a three-level tree, whose nodes are labeled A, B, C, A1, A2, ..., etc., together with a fuzzy query set u having base $S_u = \{\text{A1, A2, B1, B2}\}$, and membership values as shown. The membership values are given in thousandths for convenience of reading and are normalized according to the quadratic condition defined later in eqn. (15).

Figure 5(b) shows the pruned tree, with the membership values extended to all nodes and the associated summary gap importance values $V(\cdot)$ shown at the non-leaf nodes.

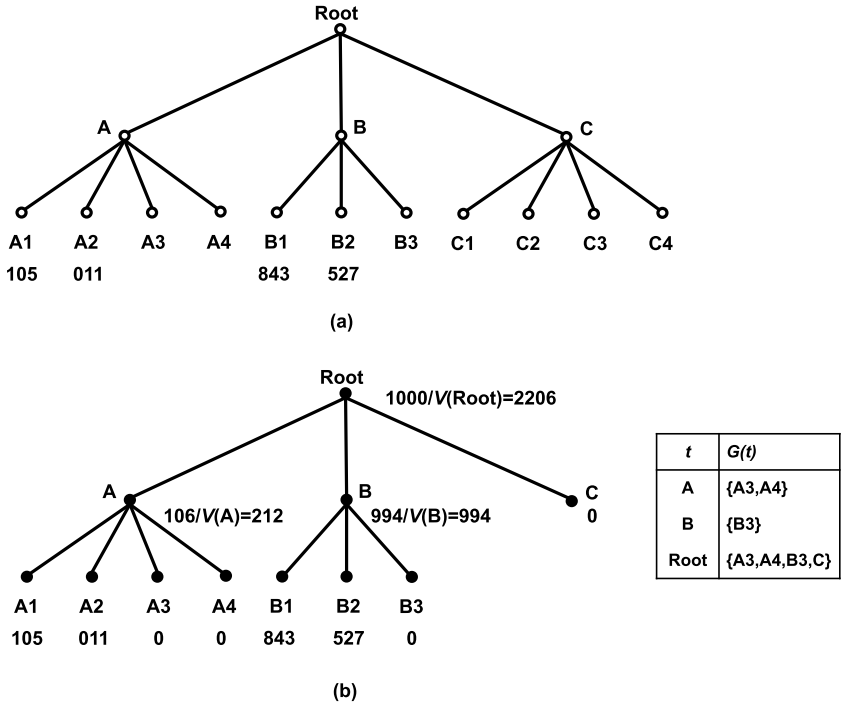

Figure 5: (a) Illustrative taxonomy T and a fuzzy query set with membership values assigned to leaves in thousandths; (b) T after pruning and annotating with membership values and summary gap importance values $V(\cdot)$.

Tables 1 and 2 illustrate successive steps of the algorithm ParGenFS at the interior nodes A, B (Table 1) and the root (Table 2), with penalty weights $\lambda = 0.2$ and $\gamma = 0.9$. The computations leading to the smaller penalty values are highlighted in boldface.

Table 1: Computational results of ParGenFS at nodes A and B of the pruned tree for the fuzzy query set u in Figure 5; the gap and offshoot penalty weights are $\lambda = 0.2$ and $\gamma = 0.9$.

i	$H(i)$	$L(i)$	$p(i)$	t		$H(t)$	$L(t)$	$p(t)$
$A1$	$\{A1\}$	$\oslash$	$0.9 \times 0.105 = 0.095$		Gain	$\{A\}$	$\{A3, A4\}$	$0.106 + 0.2 \times 0.212 = 0.148$
$A2$	$\{A2\}$	$\oslash$	$0.9 \times 0.011 = 0.010$	A				
$A3$	$\oslash$	$\oslash$	0		**No Gain**	**{A1,A2}**	$\oslash$	**0.095 + 0.010 = 0.105**
$A4$	$\oslash$	$\oslash$	0					
$B1$	$\{B1\}$	$\oslash$	$0.9 \times 0.843 = 0.759$		**Gain**	**{B}**	**{B3}**	**0.994 + 0.2×0.994 = 1.193**
$B2$	$\{B2\}$	$\oslash$	$0.9 \times 0.527 = 0.474$	B				
$B3$	$\oslash$	$\oslash$	0		No Gain	$\{B1, B2\}$	$\oslash$	$0.759 + 0.474 = 1.233$
C	$\oslash$	$\oslash$	0					

Table 2 shows that the "No Gain" scenario at the root corresponds to the smaller penalty value 1.298. The query set u thus leads to an optimal u-cover with just node B being a head subject, leaving A1 and A2 as offshoots, and a gap at B3.

Table 2: Computational results at the root for the fuzzy query set u in Figure 5, following the computations shown in Table 1; the gap and offshoot penalty weights are $\lambda = 0.2$ and $\gamma = 0.9$.

t	$H(t)$	$L(t)$	$p(t)$	t		$H(Root)$	$L(Root)$	$p(Root)$
A	$\{A1, A2\}$	$\oslash$	0.105		Gain	$\{Root\}$	$\{A3, A4, B3, C\}$	$1 + 0.2 \times 2.206 = 1.441$
B	$\{B\}$	$\{B3\}$	1.193	*Root*	**No Gain**	**{A1,A2,B}**	**{B3}**	**0.105 + 1.193 = 1.298**
C	$\oslash$	$\oslash$	0.00					

3. Applying ParGenFS to the analysis of a collection of research papers

3.1. Analysis of research publications: state of the art

For a few past decades, the analysis of research paper collections has reached a degree of maturity exemplified by two recent reviews related to (a) science of science [12] and (b) science mapping [7]. These studies encompass various types of data analysis, such as collaboration network analysis, author or paper intercitation or co-citation data analysis, cluster-analysis of text-to-keyword tables, and text

visualization. Most popular are data on intercitation and co-citation networks, as well as on similarity between papers. An intercitation network is defined by links to cited papers (or authors) forming, thus, a directed graph. A co-citation network is a weighted undirected graph in which the weight of the link between, say, papers i and j is determined by the number of cited papers common to both. A (content) similarity score between papers is defined by counting common keywords or, more rarely, by similarity between their structures (see, for instance, [2]).

A reference can be found in the literature for virtually any topic related to the structure and content analysis of research papers. The following three directions deserve a special mention, however: (a) general properties of citation networks, (b) attempts at deriving taxonomies of specific domains, and (c) the analysis of emergence and evolution of research directions and fields.

The general properties of citation networks, item (a) in the list above, typically relate to various manifestations of power law and locally Gaussian distributions [3, 12].

Deriving taxonomy or taxonomy-like structures from the stream of research papers, item (b) in the list above, is a subject of specific concerns because this method of summarization of enormous flow of publications can be extremely helpful for both individual interest and organizational planning [19, 16, 44, 37]. Manually developed taxonomies, such as the ACM-CCS [1], take enormous effort and time to create. However, so far, no sound technology for automated taxonomy generation has been developed.

Discovering major similarity groupings and their contents, item (c) in the list above, potentially allows the user to see trends in the evolution of a research domain; see, for example, [7, 18, 34] for recent results and references. Especially interesting to us are papers like [8], in which specific thematic co-citation clusters were found in Information Science, including some areas of our particular interest, such as "Information retrieval". Summarization of contents in a similarity text grouping is being pursued by various pathways. One pathway tries to discern main categories from the texts, as exemplified by Latent Dirichlet Allocation (LDA) based topic modeling (see, for example, [43, 5]). Yet another pathway involves using knowledge of the domain, usually represented by an expert-driven taxonomy such as ACM-CCS [1] (see, for example, [17, 31]). One may say that our analysis in this paper also follows this knowledge-driven approach.

3.2. Finding and lifting thematic fuzzy clusters over a text collection

Our analysis involves finding and generalizing fuzzy clusters of taxonomy leaf topics. We proceed sequentially through the following stages:

- 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 by using the algorithm ParGenFS to conceptualize them;
- Drawing conclusions from the generalizations found by ParGenFS.

Each of these stages is described in one or more of the following subsections.

3.2.1. Scholarly text collection

We have downloaded a collection of 17685 research papers, together with their abstracts, published in 17 Springer journals related to Data Science over the 20 years, 1998–2017. We take the abstracts of these papers as a representative collection. The list of the journals is shown in Table 3. The text collection is publicly available on GitHub, URL: *https://github.com/dmitsf/ParGenFS/blob/master/datasets/paper_abstracts.zip*.

Table 3: List of Springer journals related to Data Science used as the source for our text collection. Some journals start later than 1998 because of unrelated issues.

#	Title	Volumes	Years
1	Pattern Analysis and Applications	1–20	1998–2017
2	Journal of Mathematical Imaging and Vision	14–29	2001–2017
3	World Wide Web	1–20	1998–2017
4	Artificial Intelligence Review	22–48	2004–2017
5	Annals of Mathematics and Artificial Intelligence	23–80	1998–2017
6	Journal of Classification	15–34	1998–2017
7	Knowledge and Information Systems	1–52	1999–2017
8	Machine Learning	30–106	1998–2017
9	Swarm Intelligence	1–11	2007–2017
10	Applied Intelligence	14–47	1998–2017
11	Neural Processing Letters	7–45	1998–2017
12	Data Mining and Knowledge Discovery	2–31	1998–2017
13	Machine Vision and Applications	15–28	2004–2017
14	Social Network Analysis and Mining	1–7	2011–2017
15	International Journal on Document Analysis and Recognition	1–20	1998–2017
16	International Journal of Multimedia Information Retrieval	1–6	2012–2017
17	Pattern Recognition and Image Analysis	16–27	2006–2017

3.2.2. TDS Taxonomy

Taxonomy is a form of knowledge engineering that is currently growing in popularity. Best known are taxonomies such as those within the bioinformatics Genome Ontology project (GO) [14], the health and medicine SNOMED CT project [20], and the Interactive Advertising Bureau (IAB) industrial taxonomy (see *https://www.iab.com*). Mathematically, a taxonomy is a rooted tree, the nodes of which are labeled by main concepts of a domain. The tree corresponds to a relation of inclusion: the fact that node A is the parent of node B means that B is part, or a special case, of A. An important characteristic of a rooted tree is that each node has only one parent.

There are two major approaches for developing a domain taxonomy: automated and manual. The latter currently is by far the more mature and developed. However, even this approach suffers from deficiencies, as summarized in [40]: "Taxonomy design decisions regarding the used classification structures, procedures and descriptive bases are usually not well described and motivated." The automatic approach can exploit a multitude of digital resources and methods for semantic analysis. A rather comprehensive attempt is described in [40]. Our preferences, at this moment, are for taxonomies manually established by a representative body of specialists. Indeed, such a taxonomy does not depend on the purely empirical data utilized by automatic methods. Moreover, a manual taxonomy usually balances the theoretical insight and practical experience accumulated in the community represented by the body issuing it. Such is the Computing Classification System (ACM-CCS 2012) produced by the world-wide Association for Computing Machinery [1]. The subdomain of our choice is Data Science, comprising such areas as machine learning, data mining, data analysis, etc. We take the part of the ACM-CCS 2012 taxonomy that is related to Data Science and add a few leaves related to more recent Data Science developments. Our taxonomy of Data Science (TDS) is presented in full in [13], with all of its 317 leaves. The higher ranks of the taxonomy are presented in Table 4.

3.2.3. Evaluation of relevance between texts and keyphrases

After ground-breaking discoveries of methods for automatically developing sets of topics relevant to collections of documents [4, 5], it has become popular to concentrate on keywords taken from the documents being analyzed. We, however, prefer using topics produced manually by committees of experts because of the obvious advantages: comprehensiveness and stability. Therefore, our list of keyphrases comprises the leaf topics from the TDS taxonomy. Accordingly, we focus on the relevance between taxonomy keyphrases and the texts. The most popular and well-established approaches to scoring keyphrase-to-document relevance include the so-called vector-space approach [35] and the probabilistic text model approach [33]. These, however, concentrate on individual words and rely

Table 4: 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

on text pre-processing.

We utilize a method first developed by R. Pampapathi et al. [32] and subsequently extended by Chernyak and Mirkin [9,10]: the AST method for evaluating keyphrase-to-text relevance scores using just string frequency information. An advantage of the method, as described by the authors, is that it requires no manual effort, but works rather reliably.

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. The k-suffix of a string $x = x_1x_2 \dots x_N$ of length N is the contiguous end fragment $x^k = x_{N-k+1}x_{N-k+2} \dots x_N$. For example, the 3-suffix of the string $INFORMATION$ is the substring ION, and $ATION$ is the 5-suffix. Each AST node is assigned a symbol and the so-called *frequency 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 an AST has no symbol or annotation (see Figure 8). An algorithm for building an AST for any given string $x = x_1x_2 \dots x_N$ is given below.

1. Initialize the AST T to consist of a single node, the root R.
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, the longest path in T from the root that coincides with a beginning fragment $\bar{x}^{k_{max}}$ of x^k, say of

length $kmax$. At each node of the path for $\bar{x}^{k_{max}}$ add 1 to the annotation. If the length of the overlap $\bar{x}^{k_{max}}$ is less than k, the path is extended by adding new nodes corresponding to symbols from the remaining part of this suffix x^k. The 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, [15]).

Having built an AST T, we can score the string-to-document relevance over the AST. To do this, we follow [25] by computing the *conditional probability* $p(u)$ of node u in T:

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

where $f(u)$ is the frequency annotation of the node u.

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

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

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 or 3 consecutive words, create an empty AST for the document and use the above algorithm to sequentially add all these strings into the AST.

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 suffixes: “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 Levenstein-distance based approaches.

Let us build an AST, for example, for a document consisting of the single string ‘inference’. We start with the AST being a single node (the root), split the document into short strings and sequentially add suffixes of these strings to the AST. In this example, we just add all the suffixes of the only string to the AST: from ‘inference’ to ‘nference’ to ‘ference’, etc., down to ‘e’ (see Figures 6 and 7). The final AST for the string ‘inference’ is shown in Figure 8.

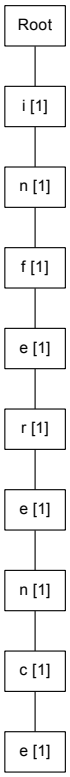

Figure 6: AST for string ‘inference’: step 1.

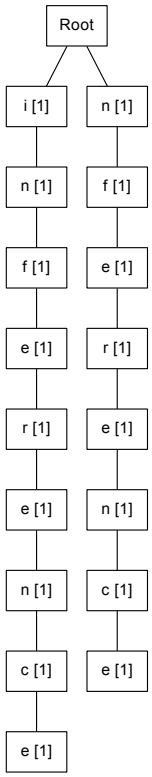

Figure 7: AST for string ‘inference’: step 2.

Let us calculate the relevance score of the string ‘fence’ for the AST in Figure 8. The string has five suffixes: ‘fence’, ‘ence’, ‘nce’, ‘ce’, ‘e’. Relevance scores for these suffixes according to formula (6) are given in Table 5. According to formula (7), the score for the whole string is:

$$S(\text{‘fence’}, T) = \frac{1}{5} \cdot (0.5555 + 0.6667 + 0.5740 + 0.5555 + 0.3333) \approx 0.537.$$

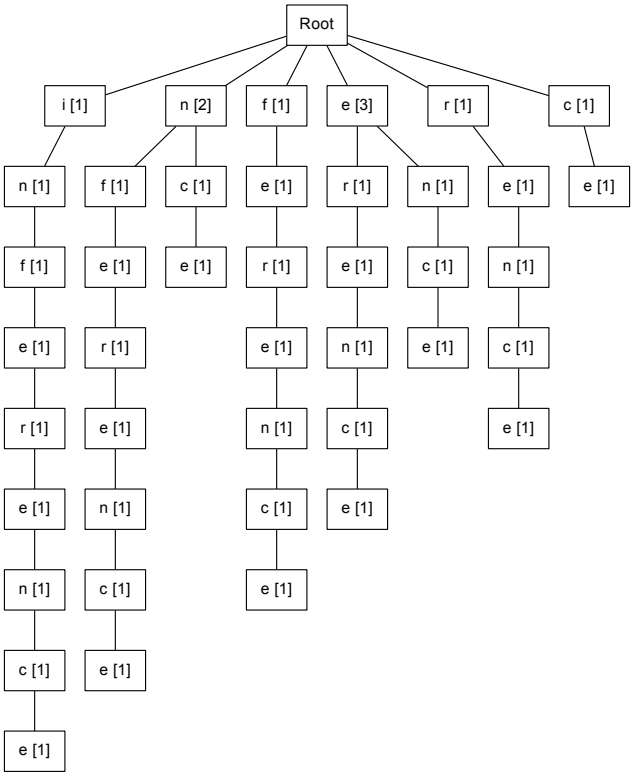

Figure 8: Final AST for string ‘inference’.

Table 5: Computing the relevance scores for suffixes of the string ‘fence’

Suffix	Match	Score
‘fence’	‘fe’	$\frac{1}{2} \cdot \left(\frac{1}{9} + \frac{1}{1}\right) \approx 0.5555$
‘ence’	‘ence’	$\frac{1}{4} \cdot \left(\frac{3}{9} + \frac{1}{3} + \frac{1}{1} + \frac{1}{1}\right) \approx 0.6667$
‘nce’	‘nce’	$\frac{1}{3} \cdot \left(\frac{2}{9} + \frac{1}{2} + \frac{1}{1}\right) \approx 0.5740$
‘ce’	‘ce’	$\frac{1}{2} \cdot \left(\frac{1}{9} + \frac{1}{1}\right) \approx 0.5555$
‘e’	‘e’	$\frac{1}{1} \cdot \left(\frac{3}{9}\right) \approx 0.3333$

Consider another example, the string ‘since’, which looks less similar to the string ‘inference’ that generated the AST. The suffix scores are given in Table 6.

For the whole string, we have:

$$S(\text{`since'}, T) = \frac{1}{5} \cdot (0 + 0.5555 + 0.5740 + 0.5555 + 0.3333) \approx 0.404.$$

Table 6: Computing the relevance scores for suffixes of string 'since'

Suffix	Match	Score
'since'	''	0
'ince'	'in'	$\frac{1}{2}\left(\frac{1}{9} + \frac{1}{1}\right) \approx 0.5555$
'nce'	'nce'	$\frac{1}{3}\left(\frac{2}{9} + \frac{1}{2} + \frac{1}{1}\right) \approx 0.5740$
'ce'	'ce'	$\frac{1}{2}\left(\frac{1}{9} + \frac{1}{1}\right) \approx 0.5555$
'e'	'e'	$\frac{1}{1}\left(\frac{3}{9}\right) \approx 0.3333$

3.2.4. 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 i and i' are relevant, the greater the inter-relation between i and i', the greater the chance for topics i and i' to fall in the same cluster. We have tried several popular clustering algorithms. Unfortunately, no satisfactory results have been obtained. Therefore, we present here results obtained with the Fuzzy ADDItive Spectral clustering (FADDIS) algorithm from [26] developed specifically for finding thematic clusters. This algorithm implements assumptions that are relevant to the task:

- LN Laplacian Pseudo-Inverse Normalization [22, 26]: A similarity data transformation that models – to an extent – heat distribution and, in this way, makes the hidden cluster structure sharper.
- AA Additivity [36]: Thematic clusters behind the texts are additive, so the similarity values are sums of the contributions from different hidden themes.
- AN Non-Completeness [24]: Clusters do not necessarily cover all the keyphrases available, as some of them may not be relevant to the text collection under consideration.

3.2.5. Co-relevance topic-to-topic similarity scores

Given a keyphrase-to-document matrix R of relevance scores, it is converted to a keyphrase-to-keyphrase similarity matrix A that scores the "co-relevance" of keyphrases according to the text collection. The similarity score $a_{ii'}$ between topics

i and i' can be computed as the inner product of vectors of scores $r_i = (r_{iv})$ and $r_{i'} = (r_{i'v})$ 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 of n_v over all $v = 1, 2, \ldots, V$. A topic is considered relevant to v if its relevance score is greater than 0.2 (a threshold found experimentally, see [9]). The numbers of texts in our collection to which different numbers of topics are relevant are shown in Table 7.

Table 7: The distribution of numbers of relevant topics in our text collection

Number of texts	Number of relevant topics
1237	0
2353	1
7114	2,3, or 4
6124	5–11
857	12 or more

This topic-to-topic similarity measure has the following properties [26]:

- The similarity matrix is positive semi-definite.
- The similarity between two topics is positive if and only if there is at least one text to which both are relevant.
- The greater the individual relevance scores, the greater the similarity.
- Given a pair of topics, the greater the set of texts to which both are relevant, the greater the similarity.

3.2.6. Additive fuzzy spectral clustering

Let us denote the total set of leaf topics by I and assume that a fuzzy cluster on I is represented by a fuzzy membership vector $\vec{u} = (u_i)$, such that $0 \leq u_i \leq 1$ for all $i \in I$, together with an intensity $\mu > 0$, a scale parameter that relates the membership scores to the similarity scores. For a set of research topics I and a membership vector $\vec{u} = (u_i)$, representing a semantic substructure of the corpus of research papers under consideration, the product $(\mu u_i)(\mu u_{i'}) = \mu^2 u_i u_{i'}$ can be considered as the contribution of the research direction represented by the cluster under consideration to the total similarity score $a_{ii'}$ between topics i and i'. In the additive fuzzy clustering model in [26], 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 that are to be minimized:

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

where $\vec{u}_k = (u_{ki})$ is the membership vector of cluster k and μ_k is its intensity. These assumptions require that clusters are extracted according to an additive model. The method developed in [26], Fuzzy ADDItive Spectral (FADDIS), finds clusters in eqn. (8) one-by-one, which accords with the assumptions above. Some theoretical and experimental computation results in [26] demonstrate that FADDIS is competitive with popular fuzzy clustering approaches.

A fuzzy cluster $(\vec{u},\ \mu)$ is extracted to minimize the one-cluster least-squares criterion

$$E = \sum_{i,i' \in I} (w_{ii'} - \xi u_i u_{i'})^2 \tag{9}$$

with respect to unknown positive $\xi = \mu^2$ and fuzzy membership vector $\vec{u} = (u_i)$, for a given similarity matrix $W = (w_{ii'})$.

Initially, $W = A$. Then the matrix W is changed by subtracting from it that part of the similarities that is accounted for by the found cluster: $W \leftarrow W - \mu^2 \vec{u}\vec{u}^T$, where μ and $\vec{u}$ are the intensity and membership vector of the cluster, respectively.

In this way, A is additively decomposed in accordance with the model (8) in a step-by-step fashion; so the number of clusters K can be determined during the process. The first-order optimality condition for the criterion E in eqn. (9) states that an optimal ξ satisfies:

$$\xi = \frac{\sum_{i,i' \in I} w_{ii'} u_i u_{i'}}{\sum_{i \in I} u_i^2 \sum_{i' \in I} u_{i'}^2},$$

or, in matrix terms,

$$\xi = \frac{\vec{u}^T W \vec{u}}{(\vec{u}^T \vec{u})^2}. \tag{10}$$

By substituting this value of ξ into eqn. (9), we obtain

$$E = S(W) - \xi^2 \left(\vec{u}^T \vec{u}\right)^2,$$

where $S(W) = \sum_{i,i' \in I} w_{ii'}^2$ is the similarity data scatter. Denoting the last term by $G(\vec{u})$, we have

$$G(\vec{u}) = \xi^2 \left(\vec{u}^T \vec{u}\right)^2 = \left(\frac{\vec{u}^T W \vec{u}}{\vec{u}^T \vec{u}}\right)^2. \tag{11}$$

This provides for a decomposition of the similarity data scatter into explained and unexplained parts, according to the equation above:

$$S(W) = G(\vec{u}) + E. \tag{12}$$

Therefore, to minimize E, one has to maximize $G(\vec{u})$ in eqn. (11), or equivalently its square root:

$$g(\vec{u}) = \xi \vec{u}^T \vec{u} = \frac{\vec{u}^T W \vec{u}}{\vec{u}^T \vec{u}}. \tag{13}$$

The value $g(\vec{u})$ in eqn. (13) is the so-called Rayleigh quotient, whose maximum is the maximum eigenvalue of W reached at the corresponding eigenvector. This recalls the celebrated spectral clustering approach (see a tutorial in [22]). According to this approach, one first solves the unconstrained problem of maximizing $g(\vec{u})$ with respect to any $\vec{u}$, to obtain the normalized eigenvector $\vec{z}$ corresponding to the maximum eigenvalue of W. Then its projection $\vec{u}$ onto the set of non-negative fuzzy membership vectors is found:

$$u_i = \begin{cases} 0, & \text{if } z_i \leq 0; \\ z_i, & \text{if } 0 < z_i \leq 1. \end{cases} \tag{14}$$

As the criterion in eqn. (13) involves division by the squared Euclidean norm of $\vec{u}$, the Euclidean normalization condition is most natural and the extracted cluster $\vec{u}$ is normalized accordingly, so that

$$\vec{u}^T \vec{u} = 1. \tag{15}$$

The process of extracting fuzzy clusters one-by-one stops when any of the following conditions is satisfied:

1. The value of ξ given by eqn. (10) at the current step is negative.
2. The contribution of the last extracted cluster is too low. For example, a cluster should contribute at least as much as an average entity, so that the threshold $1/N$ should be considered a well-defined conservative value for the threshold.
3. The residual scatter E becomes smaller than, say, 5% of the original similarity data scatter.
4. A pre-specified number $K_{\max}$ of clusters have been extracted.

To make the hidden cluster structure in similarity data sharper, we apply the so-called Laplacian normalization [22]. This normalization is usually applied with the so-called minimum normalized cut criterion and thus involves the minimum eigenvalue, whereas FADDIS uses the maximum one. This is why this normalization is modified in [26] to involve the inverse eigenvalues. Specifically, the so-called Laplacian pseudo-inverse transformation (Lapin, in short) is applied. Given a similarity matrix W, the $N \times N$ diagonal matrix D is computed, with (i,i)-th entry equal to $d_i = \sum_{i' \in I} w_{ii'}$, the sum of row i of W, and the Laplacian is defined as $L_n = I - D^{-1/2} W D^{-1/2}$. Then the Laplace Pseudo INverse transformation (Lapin) is defined as

$$L_n^+ = Z \tilde{\Lambda}^{-1} Z^T,$$

where Z is the matrix of eigenvectors corresponding to non-zero eigenvalues, from the spectral decomposition $L_n = Z \Lambda Z^T$, and the diagonal matrix $\tilde{\Lambda}^{-1}$ is obtained

from Λ by removing zero eigenvalues and replacing each non-zero eigenvalue λ by $1/\lambda$. Lastly, the eigenvector corresponding to the maximum eigenvalue of L_n^+ is taken to generate the fuzzy cluster for the model (9).

3.2.7. FADDIS thematic clusters

After computing the 317×317 topic-to-topic co-relevance matrix, converting it 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, as can be seen from the lists of most represented topics in these clusters, which are shown in Tables 8, 9 and 10. We denote the clusters using the letters L (Learning), R (Retrieval) and C (Clustering).

Table 8: Cluster L (Learning): topics with membership values greater than 0.15

$u(t)$	Code	Topic
0.300	5.2.3.8.	rule learning
0.282	5.2.2.1.	batch learning
0.276	5.2.1.1.2.	learning to rank
0.217	1.1.1.11.	query learning
0.216	5.2.1.3.3.	apprenticeship learning
0.213	1.1.1.10.	models of learning
0.203	5.2.1.3.5.	adversarial learning
0.202	1.1.1.14.	active learning
0.192	5.2.1.4.1.	transfer learning
0.192	5.2.1.4.2.	lifelong machine learning
0.189	1.1.1.8.	online learning theory
0.166	5.2.2.2.	online learning settings
0.159	1.1.1.3.	unsupervised learning and clustering

Table 9: Cluster R (Retrieval): topics with membership values greater than 0.15

$u(t)$	Code	Topic
0.211	3.4.2.1.	query representation
0.207	5.1.3.2.1.	image representations
0.194	5.1.3.2.2.	shape representations
0.194	5.2.3.6.2.1	tensor representation
0.191	5.2.3.3.3.2	fuzzy representation
0.187	3.1.1.5.3.	data provenance
0.173	2.1.1.5.	equational models
0.173	3.4.6.5.	presentation of retrieval results
0.165	5.1.3.1.3.	video segmentation
0.155	5.1.3.1.2.	image segmentation
0.154	3.4.5.5.	sentiment analysis

Table 10: Cluster C (Clustering): topics with membership values greater than 0.15

$u(t)$	Code	Topic
0.327	3.2.1.4.7	biclustering
0.286	3.2.1.4.3	fuzzy clustering
0.248	3.2.1.4.2	consensus clustering
0.220	3.2.1.4.6	conceptual clustering
0.192	5.2.4.3.1	spectral clustering
0.187	3.2.1.4.1	massive data clustering
0.159	3.2.1.7.3	graph based conceptual clustering
0.151	3.2.1.9.2.	trajectory clustering

3.2.8. The generalization step: Lifting clusters L, R, and C within TDS

The three clusters above are lifted in the TDS taxonomy using the algorithm ParGenFS with penalty weights $\lambda = 0.1$ and $\gamma = 0.9$. The offshoot penalty $\gamma = 0.9$ is chosen to be slightly less than 1.0, the penalty for a head subject. The gap penalty $\lambda = 0.1$ reflects our desire to lift a cluster to a parental node even when the cluster membership values of the topics at the children are quite modest. Consider, for example, a crisp cluster in which just 2 of the 10 children of a node t are members of the cluster under consideration. Lifting the head subject to t would create 8 gaps and cost $1 + 8 * \lambda$, giving a penalty cost of 1.8 when $\lambda = 0.1$ or 2.6 when $\lambda = 0.2$. Not lifting the two member children, but making them head subjects themselves, leads to a penalty cost equal to 2. Therefore, in the case when $\lambda = 0.1$, this would lead here to lifting the cluster whereas, when $\lambda = 0.2$, it would not.

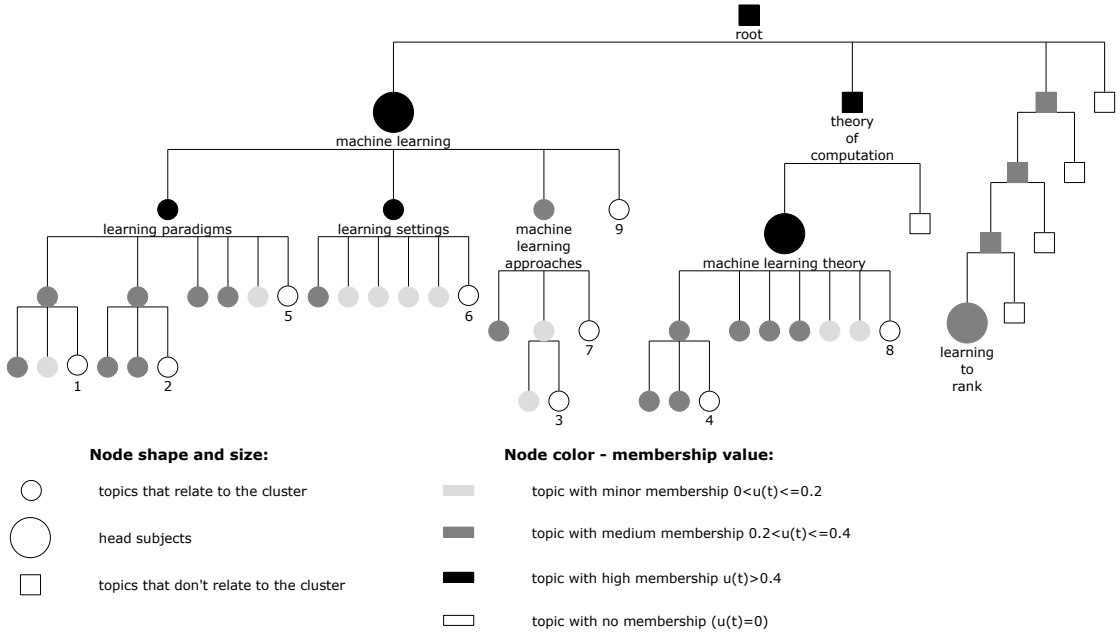

Figure 9: Lifting results for Cluster L (Learning). The gaps are numbered as in Table 11.

Another issue deserving attention is the metod of extending leaf membership values to interior nodes of the taxonomy. Since our fuzzy clusters are found using the FADDIS algorithm described above, the leaf membership values are normalized according to the Euclidean norm condition in eqn. (15). This normalization is extended to the interior leaves, as shown in the example in Figure 5 (b), where the membership values of the interior nodes, A, B and C are 0.106, 0.994 and 0, respectively; the square root of the sum of their squares is 0.9996 which is indeed unity up to a small admissible error.

The results of lifting Cluster L (Learning) are shown in Figure 9. The cluster has been lifted to three head subjects: Machine Learning, Machine Learning Theory, and Learning to Rank. These head subjects reflect the structure of the general concept "Learning" according to our text collection. The list of gaps is less instructive, probably reflecting a relatively modest coverage of the domain by the publications in the collection (see Table 11).

Table 11: 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

Similar comments can be made with respect to the results of lifting Cluster R (Retrieval), which are shown in Figure 10. The two head subjects, Information Systems and Computer Vision, reflect the structure of "Retrieval" in the set of publications under consideration. The gaps for Cluster R are listed in Table 12.

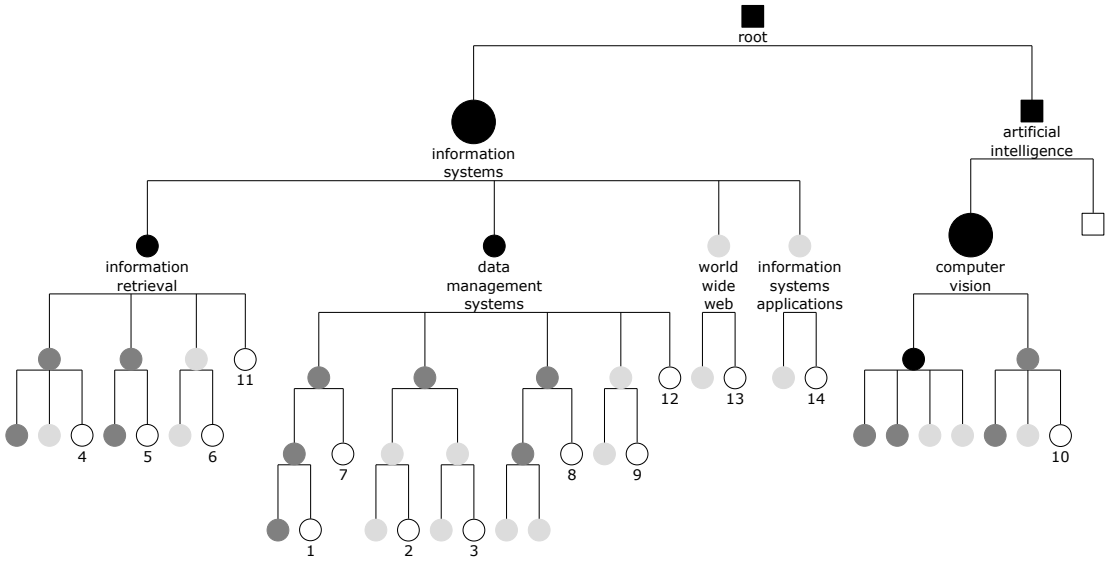

Figure 10: Lifting results for Cluster R (Retrieval). The gaps are numbered as in Table 12. The elements in the figure are explained in the legend to Fig. 9

We show no figure for the results of lifting Cluster C, because the corresponding

Table 12: Gaps at the lifting of Cluster R

Number	Topics
1	data streams, incomplete data, temporal data, uncertainty, inconsistent data
2	multidimensional range search, point lookups, unidimensional range search, proximity search
3	data compression, record and block layout
4	query intent, query suggestion, query reformulation
5	test collections, relevance assessment, retrieval effectiveness, retrieval efficiency
6	question answering, document filtering, recommender systems, information extraction, expert search, clustering and classification, summarization, business intelligence
7	relational database model, entity relationship models, hierarchical data models, network data models, physical data models
8	relational database query languages, mapreduce languages, call level interfaces
9	data exchange, data cleaning, mediators and data integration, entity resolution, data warehouses
10	interest point and salient region detections, shape inference, object detection, object recognition, object identification, tracking, reconstruction, matching
11	document representation, users and interactive retrieval, retrieval models and ranking, specialized information retrieval
12	database management system engines
13	site wrapping, data extraction and integration, traffic analysis, knowledge discovery
14	data cleaning, collaborative filtering, association rules, clustering, data stream mining, graph mining, process mining, text mining, data mining tools, sequence mining

taxonomy fragment is too large and the lifting results are too fragmentary. There are 16 head subjects, which are presented in Table 13 together with their TDS codes.

One can see from the list in Table 13 that the core clustering subjects are supplemented by methods and environments – this indicates that the ever-increasing role of clustering activities should be better reflected in the taxonomy.

3.2.9. Drawing conclusions

We can see that the topic clusters found from the text collection do indeed highlight areas of current and nascent developments. The three clusters under consideration closely relate to the following developments:

- theoretical and methodical research in learning, as well as integrating the subject of Learning to Rank within the mainstream;
- representation of various types of data for information retrieval, including annotated visual data; and

Table 13: Head subjects at the lifting of Cluster C

Number	Topic	Code
1	clustering	3.2.1.4
2	graph based conceptual clustering	3.2.1.7.3
3	trajectory clustering	3.2.1.9.2
4	clustering and classification	3.4.5.8
5	unsupervised learning and clustering	1.1.1.3
6	spectral methods	5.2.4.3
7	document filtering	3.4.5.2
8	language models	3.4.3.3
9	music retrieval	3.4.7.2.4
10	collaborative search	3.4.3.4
11	database views	3.1.3.7
12	stream management	3.1.3.12
13	database recovery	3.1.3.3.3
14	MapReduce languages	3.1.4.3.1
15	logic and databases	1.1.2.10
16	language resources	5.1.1.9

- various types of clustering in different branches of the taxonomy related to a variety of applications and methods.

In particular, one can see from the "Learning" head subjects (see Figure 9) that the main work here still concentrates on theory and method rather than applications. The good news is that the field of learning, formerly focused mostly on tasks of learning subsets and partitions, is currently expanding towards learning ranks and rankings. Of course, many subareas remain to be covered: these can be discerned from the list of gaps in Table 11.

Moving to the lifting results for the information retrieval cluster R (see Figure 10 and Table 9), we can clearly see the current tendencies. Rather than relating the term "information" to texts only, as was the case in previous stages of the process of digitalization, visual data is becoming an integral part 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 human communication via language, there is no comparable hierarchy of meanings for images. One may only guess that the elements in this cluster related to segmentation of images and videos, as well as those related to data management systems, are going to be fundamental to a future multilevel system of meanings for images and video data. This is a direction for future developments clearly seen from Figure 10.

We cannot help but compare this conclusion with conclusions made in [8] regarding similarly named clusters, viz. "Information Retrieval". One was found as

an author co-citation cluster over a collection of 5963 records and a corresponding 633-strong author co-citation network, and the other as a document co-citation cluster over a network of 655 references for the period 1996–2008. The authors point out that the interpretation of the latter cluster should be less ambiguous than the former. Two aspects of each cluster have been pointed out in [8]: "(1) prominent members of a cluster as the intellectual basis and (2) themes identified in the citers of the cluster as research fronts."

Prominent members of the Information Retrieval cluster in [8] are books on the subject by G. Salton and C. Van Rijsbergen, as well as a paper by S. Robertson - all well known to specialists. The main themes identified in "main citers" are summarized as title words: "information retrieval", "probabilistic model", "query expansion", and "using heterogeneous thesauri" [8]. Another summarization comes from sentences selected according to algorithms for summarization. These sentences are: "The relationships identified between the five best terms selected by the users for query expansion and the initial query terms were that: (a) 34 % of the query expansion terms have no relationship or other type of correspondence with a query term; (b) 66 % of the remaining query expansion terms have a relationship to the query terms. We provide data on: (i) sessions – changes in queries during a session, number of pages viewed, and use of relevance feedback; (ii) queries - the number of search terms, and the use of logic and modifiers; and (iii) terms - their rank/ frequency distribution and the most highly used search terms." [8].

One can see how different is our description of the cluster from that in [8]. The latter cites aspects found in a few representative papers, whereas our description is based on a taxonomy integrating various aspects of the research. We do not discuss the differences related to the difference in the text collections. It is probably worth noting that using a collection of publications from a later period, 2009-2016, the authors of a follow-up work [16] claim that the Information Retrieval cluster is widening to "Information Behavior" to cover more aspects of Information Sciences.

Regarding the "Clustering" cluster C with its far too many head subjects, one may conclude that perhaps the time has already come, or will imminently, when the subject of clustering must be raised to a higher level in the taxonomy that embraces all these head subjects. 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, viz. unsupervised learning. Perhaps, we are soon 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 major part of knowledge engineering. When discussing the role of classification as a knowledge engineering phenomenon, one encounters three conventional aspects of classification:

- structuring phenomena;
- relating different aspects of phenomena to each other;

- shaping and maintaining knowledge of phenomena.

Each of these can become a separate direction of research in knowledge engineering.

4. Conclusion

This paper presents a formalization of the concept of generalization, an important part of the human ability for conceptualization. According to the Merriam-Webster Dictionary, to generalize is "to derive or induce (a general conception or principle) from particulars". We assume that such an operation may require the availability of a coarser granularity of the structuring of the domain. This is captured by the idea of lifting a query set to higher ranks in a hierarchical taxonomy of the domain.

The hierarchical structure of a taxonomy entails the possibility of inconsistencies between a query set and the taxonomy structure. These inconsistencies can be of one of two types – gaps or offshoots, which potentially arise at a higher rank "head subject" in order to cover the query set. A gap is a node of the taxonomy that is covered by a head subject but does not belong in the query set. An offshoot is a node of the taxonomy that does belong in the query set even though it is not covered by a head subject. The higher the rank of a candidate for a conceptual head subject, the larger is the number of gaps. The lower the rank of the head subject, the larger is the number of offshoots. Our algorithm ParGenFS finds a globally optimal lifting that balances the numbers of head subjects, gaps, and offshoots depending on the associated penalties for each of these elements.

We cannot compare our approach to any similar ones because, to the best our knowledge, this is the first attempt at formalizing the notion of generalization as applied to a set of concepts.

We illustrate the usefulness of this approach using a set of 17685 abstracts of research papers, published by Springer in 17 journals related to Data Science during the 20 years from 1998 to 2017. We use this set to obtain six fuzzy clusters of taxonomy topics according to their co-relevance. We can easily interpret only three out of the found clusters, which probably reflects the inherent randomness of research processes and some tendencies that we failed to notice because they have not yet been explicitly formulated. It should be pointed out that our approach to finding interpretable clusters of taxonomy topics over the textual data requires using rather sophisticated methods, including spectral clustering, the weighting of publications, and Laplacian transformation. This complexity perhaps comes, at least partly, from the way we estimate the similarity between concepts – by considering them as just strings and consequently ignoring any imposed pre-structuring deriving from pre-selected keywords or NLP constructions. The lifting of these clusters in the TDS taxonomy suggests several general conclusions about current

research in Data Science. These conclusions, even if not entirely unexpected, give a glimpse into the hidden research processes, as captured by the authorship of the Springer journals. We emphasise that the essence of our interpretations differs significantly from that in the approach based on the analysis of citation networks. The difference stems from the fact that our approach involves a coarser granulation than that in the texts under consideration, whereas the analysis of citation networks remains on the same level of granularity.

The proposed approach to generalization can be utilised in a number of similar tasks, such as the positioning of a research project, the interpretation of a concept that is not present in the taxonomy, or the annotation of a set of research articles. These all are parts of the processes of long-term research analysis and planning within which our approach should be positioned.

Among major issues requiring further development in this direction, three of the most relevant are:

- Specifying penalty weights
- Taxonomy modification
- Taxonomy development

The subject of automation of the choice of penalty weights in ParGenFS algorithm is of key importance in our approach. We think that reasonable computational progress on the penalty weights can be achieved by replacing the criterion of maximum parsimony by the criterion of maximum likelihood. The latter criterion can be formulated and applied if each node of the taxonomy is assigned with probabilities of "gain" and "loss" of topic events. The ParGenFS algorithm could then straightforwardly be modified to use the criterion of maximum likelihood in place of the current criterion because of the additivity of the log-likelihoods. We intend pursuing this approach for the TDS taxonomy in the near future.

The issue of taxonomy modification can be addressed, in our opinion, in the framework of the approach to generalization developed in this paper. Consider, for example, our results on lifting the "Clustering" fuzzy cluster C: we obtained 16 head subjects for this cluster, resulting in 16 penalty units for the head subjects. However, we could allow, within the same framework, one other admissible event, in addition to those of introducing head subjects, gaps and offshoots. This additional event could be the introduction of a new node into the tree as a head subject. For our example of Cluster C, this new node could be the node "clustering" as the parent of all the 16 head subjects, thus decreasing the number of head subjects by 15, although possibly increasing the number of gaps. Of course, the introduction of a new node must be penalized too, say, by 5 penalty units – still, a likely reduction in the penalty if the number of new gaps is small. We plan to develop the necessary machinery in a follow-up paper.

The issue of taxonomy development needs more attention, both from research communities and planning committees. Specifically, most urgent directions for further advancements here are:

- developing better methods of automating the process of creating taxonomies (an interesting approach to this is described in [19]); and

- practical usage of domain taxonomies by specialists, for example, at conferences and meetings of research communities and committees (the need for this is pointed out in [28]).

Acknowledgment. D.F. and B.M. acknowledge continuing support by the Academic Fund Program at the National Research University Higher School of Economics (grant 19-04-019 in 2018–2019), and by the International Decision Choice and Analysis Laboratory (DECAN) NRU HSE, in the framework of a subsidy granted to the HSE by the Government of the Russian Federation for the implementation of the the Russian Academic Excellence Project "5-100". S.N. acknowledges support by FCT/MCTES, NOVA LINCS (UID/CEC/04516/2019).

The authors are indebted to the reviewers for their helpful queries and comments, which have been taken into account in preparing the final version of the paper.

References

[1] The 2012 ACM Computing Classification System. [Online]. Available: *http://www.acm.org*/about/class/2012 (Accessed 2018, 30 November).

[2] D. R. Amancio, "Comparing the topological properties of real and artificially generated scientific manuscripts," *Scientometrics*, 105, pp. 17631779, 2015.

[3] D.R. Amancio, O.N. Oliveira Jr., L.F. Costa. "Three-feature model to reproduce the topology of citation networks and the effects from authors visibility on their h-index," *Journal of Informetrics*, 6, pp. 427434, 2012.

[4] D.M. Blei, D.M. Ng, M.I. Jordan, "Latent Dirichlet allocation," *The Journal of Machine Learning Research*, 3, pp. 993–1022, 2003.

[5] D. Blei, "Probabilistic topic models, " *Communications of the ACM*, 55 (4), pp. 77–84, 2012.

[6] R.K. Brouwer, "A method of relational fuzzy clustering based on producing feature vectors using FastMap," *Information Sciences*, 179, pp. 3561-3582, 2009.

[7] C. Chen, "Science mapping: A systematic review of the literature," *Journal of Data and Information Science*, vol. 2, no. 2, pp 1- 40, 2017.

[8] C. Chen, F. Ibekwe-SanJuan and J. Hou, "The structure and dynamics of co-citation clusters: A multiple-perspective co-citation analysis", *Journal of the American Society for information Science and Technology*, vol 61, no. 7, 1386-1409, 2010.

[9] E. Chernyak, "An approach to the problem of annotation of research publications," *Proceedings of the eighth ACM international conference on web search and data mining*, ACM, 429-434, 2015.

[10] E. Chernyak, B. Mirkin, "Refining a Taxonomy by Using Annotated Suffix Trees and Wikipedia Resources," *Annals of Data Science*, 2(1), pp. 61-82, 2015.

[11] W.A. Firestone, "Alternative arguments for generalizing from data as applied to qualitative research," *Educational Researcher*, vol. 22, 4 pp. 16-23, 1993.

[12] S. Fortunato, C.T. Bergstrom, K. Brner, J.A. Evans, D. Helbing, S. Milojevi, A.M. Petersen, F. Radicchi, R. Sinatra, B. Uzzi, A. Vespignani, L. Waltman, D. Wang, A.-L. Barabsi. "Science of science," *Science*, 359 (6379), eaao0185, 2018 (*http://science.sciencemag.org/content/359/6379/eaao0185*).

[13] D. Frolov, B. Mirkin, S. Nascimento, and T. Fenner, "Finding an appropriate generalization for a fuzzy thematic set in taxonomy," Working paper WP7/2018/04 – Moscow: Higher School of Economics Publ. House, Series WP7 "Mathematical methods for decision making in economics, business and politics." 60 p., 2018.

[14] Gene Ontology Consortium, "Gene ontology consortium: going forward," *Nucleic Acids Research*, vol. 43 D1049-D1056, 2015.

[15] R. Grossi and J.S. Vitter, "Compressed suffix arrays and suffix trees with applications to text indexing and string matching," *SIAM Journal on Computing*, 35, 2 pp. 378-407, 2005.

[16] J. Hou, X. Yang and C. Chen, "Emerging trends and new developments in information science: A document co-citation analysis (20092016)," *Scientometrics*, 115, 2, pp. 869-892, 2018.

[17] S. Kapur, O. F. Ayman, and R. E. Chatwin. "Method and apparatus for representing text using search engine, document collection, and hierarchal taxonomy." U.S. Patent No. 7,580,926. 2009.

[18] T. Kawamura, K. Watanabe, N. Matsumoto, S. Egami, and M. Jibu, "Funding map using paragraph embedding based on semantic diversity," *Scientometrics*, vol. 116, no. 2, pp. 941958, 2018.

[19] R. Klavans, and K. W. Boyack, "Which type of citation analysis generates the most accurate taxonomy of scientific and technical knowledge?", *Journal of the Association for Information Science and Technology*, 68(4), pp. 984-998, 2017.

[20] D. Lee, R. Cornet, F. Lau, N. De Keizer, "A survey of SNOMED CT implementations," *Journal of Biomedical Informatics*, vol. 46, no. 1, pp. 87-96, 2013.

[21] E. Lloret, E. Boldrini, T. Vodolazova, P. Martinez-Barco, R. Munoz, and M. Palomar, "A novel concept-level approach for ultra-concise opinion summarization," *Expert Systems with Applications*, 42(20), pp. 7148-7156, 2015.

[22] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, pp. 395-416, 2007.

[23] Merriam-Webster website. [Online]. Available: *https://www.merriam-webster.com/dictionary/generalize* (Accessed 2018, 28 November).

[24] B. Mirkin, *Clustering: A Data Recovery Approach*, Chapman and Hall/CRC Press, 2d Edition, 2012.

[25] B.G. Mirkin, E.L. Chernyak, O.N. Chugunova, *Metod annotirovannogo suffiksnogo dereva dlja ocenki stepeni vhozhdenija strok v tekstovye dokumenty*. [Annotated Suffix Tree as a Way of String-To-Document Score Evaluating]. Business Informatics, 3 (21), pp. 31–41, 2012 (in Russian).

[26] B. Mirkin, S. Nascimento, "Additive spectral method for fuzzy cluster analysis of similarity data including community structure and affinity matrices," *Information Sciences*, vol. 183, no. 1, pp. 16-34, 2012.

[27] B. Mirkin, M. Orlov, "Three aspects of the research impact by a scientist: measurement methods and an empirical evaluation." In *Optimization, Control, and Applications in the Information Age*, Springer, Cham, pp. 233-259, 2015.

[28] F. Murtagh, M. Orlov, and B. Mirkin, "Qualitative judgement of research impact: Domain taxonomy as a fundamental framework for judgement of the quality of research." *Journal of Classification*, 35(1), pp. 5-28, 2018.

[29] G. Mueller and R. Bergmann, "Generalization of Workflows in Process-Oriented Case-Based Reasoning,", In FLAIRS Conference, pp. 391-396, 2015.

[30] D. Nallaperuma, and D. De Silva, "A participatory model for multi-document health information summarisation," *Australasian Journal of Information Systems*, 21.

[31] S. Nascimento, T. Fenner, B. Mirkin, "Representing research activities in a hierarchical ontology," in *Procs. of 3rd International Workshop on Combinations of Intelligent Methods and Applications* (CIMA 2012), Montpellier, France, August, 28, pp. 23-29, 2012.

[32] R. Pampapathi, B. Mirkin and M. Levene, "A suffix tree approach to anti-spam email filtering," *Machine Learning*, 65(1), pp. 309-338, 2006.

[33] P. Robinson and S. Bauer, *Introduction to Bio-Ontologies*, Chapman and Hall/CRC Press, 2011.

[34] A.A. Salatino, F. Osborne, and E. Motta, "How are topics born? Understanding the research dynamics preceding the emergence of new areas", *Peer J Computer Science*, 3, e119 , 2017.

[35] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing and Management*, vol. 25, no 5, pp. 513-523, 1998.

[36] R.N. Shepard and P. Arabie, "Additive clustering: representation of similarities as combinations of discrete overlapping properties," *Psychological Review*, vol. 86, pp. 87-123, 1979.

[37] F.N. Silva, D.R. Amancio, M. Bardosova, L.F. Costa, O.N. Oliveira Jr. "Using network science and text analytics to produce surveys in a scientific topic," *Journal of Informetrics*, 10, pp. 487502, 2016.

[38] R. Snow, D. Jurafsky, and A.Y. Ng, "Semantic taxonomy induction from heterogenous evidence." In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, pp. 801-808, 2006.

[39] Y. Song, S. Liu, H. Wang, Z. Wang, and H. Li, "Automatic taxonomy construction from keywords," U.S. Patent No. 9,501,569. Washington, DC: U.S. Patent and Trademark Office, 2016.

[40] M. Usman, R. Britto, J. Boerstler, and E. Mendes, "Taxonomies in software engineering: A Systematic mapping study and a revised taxonomy development method," *Information and Software Technology*, 85, pp. 43-59, 2017.

[41] N. Vedula, P.K. Nicholson, D. Ajwani, S. Dutta, A. Sala, and S. Parthasarathy, "Enriching Taxonomies With Functional Domain Knowledge," In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, ACM, pp. 745-754, 2018.

[42] J. Waitelonis, C. Exeler, and H. Sack, "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)*, CEUR-WS, vol. 1486, 2015.

[43] C. Wang, D.M. Blei, "Collaborative topic modeling for recommending scientific articles," In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 448-456, 2011.

[44] C. Wang, X. He, and A. Zhou, "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.