# A Hybrid Approach to the Analysis of a Collection of Research Papers

Boris Mirkin[1,2][0000−0001−5470−8635], Dmitry Frolov[1,4][0000−0002−0370−3559], Alex Vlasov[1], Susana Nascimento[3], and Trevor Fenner[2]

[1] Department of Data Analysis and Artificial Intelligence, HSE University, Moscow, Russia {bmirkin,dfrolov}@hse.ru

[2] Department of Computer Science and Information Systems, Birkbeck University of London, London

[3] UK Department of Computer Science and NOVA LINCS, Universidade Nova de Lisboa, Caparica, Portugal

[4] Natimatica, Ltd., Moscow, Russia.

**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". Our method involves two more automated analysis techniques: a fuzzy clustering method, FADDIS, involving both additive and spectral properties, and a purely structural string-to-text relevance measure based on suffix trees annotated by frequencies. We apply this to extract research tendencies from two collections of research papers: (a) about 18000 research papers published in Springer journals on data science for 20 years, and (b) about 27000 research papers retrieved from Springer and Elsevier journals in response to data science related queries. We consider a taxonomy of Data Science based on the Association for Computing Machinery Classification of Computing System (ACM-CCS 2012). Our findings allow us to make some comments on the tendencies of research that cannot be derived by using more conventional techniques.

**Keywords:** Hybrid approach · Generalization · Fuzzy cluster · Annotated suffix tree · Research tendency.

## 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. There are many papers tackling various aspects of this. In our view, however, the mainstream of all the efforts currently constitute approaches based on the analysis of structure and dynamics of graphs/networks of interrelations between papers, or articles, (sometimes, between authors) or between research concepts. Paper [5] exemplifies the former, more recent papers

[3, 9] – the latter. Arguably, the latter, analysis of concept networks is less computationally intensive than the former, because the sizes of concept graphs are much smaller than those of graphs of articles. Yet results of structural analyses are frequently unstable, much dependent on the datasets involved, and, also, difficult to use for knowledge engineering.

Consider, for illustration, a result from [9]: three sets of keywords returned by three different methods as response to query "Economic growth" in Table 1. One cannot help but noticing how different and, sometime, arbitrary are keywords returned by algorithms. This type of return is difficult to interpret and automate.

**Table 1.** Three sets of keywords returned by three different topic modeling methods in response to query "Economic growth" in [9], Table 3 on page 228.

| Management information system | Economic adjustment | Stages of growth model |
|---|---|---|
| Tobacco | Economic policy | Growth policy |
| Internet Usage | Growth policy | Resource wealth |
| Eurobond | Economic development | Kuznets curve |
| Automobile engine | Economic reform | Export-led growth |

The goal of this paper is developing a coherent methodology for conceptual analysis of research paper collections that would lead to unified conceptual representations more suitable for automated analysis. The very first provision is to restrict the arbitrariness of keywords, be they supplied by authors, like in [3], or extracted from texts, like in [9]. To achieve that, we use a domain taxonomy, so that the set of keywords is a subset of the taxonomy leaf topics. A taxonomy, in this paper, is a rooted tree whose nodes are annotated by domain concepts in such a way that parental nodes are tagged by concepts more general than concepts assigned to the children nodes. In spite of the recent surge in efforts for automated taxonomy building (see, for a review, [15]), no sound automated taxonomy making method has been developed so far. We definitely prefer using a manually developed taxonomy such as ACM Classification of Computing Systems 2012 by the international Association for Computing Machinery [2].

Therefore, the set of keywords here is constant. This would shield us from empirical biases which are immanent to the approaches that use keywords derived from the texts under analysis. There is a negative side too: some of our leaf-related keywords may appear little relevant or even irrelevant to this or that article from the collection. Therefore, we need a method for assessment of relevance between keywords and texts, which would provide us with robust relevance scoring independently of the way at which keywords appear in the text. Such a method has been proposed and substantiated, with our participation, to evaluate similarity between texts and keywords considered as strings of symbols, the so-called Annotated Suffix Tree approach (see in [6, 13]).

Our next step will be for obtaining clusters of keywords so that those keywords that tend to co-occur in the same texts would tend to belong to the same clusters. The clusters sought should be fuzzy to reflect semantic relations between keywords. Therefore, the next stage of our approach is in using the obtained keyword-to-text relevance scores for finding fuzzy clusters of keywords, that tend to co-occur in the same texts.

Conventionally, obtaining such a cluster or set of clusters would be considered "the end of the story", like it is in popular methods for topic modeling [4, 1]. We, however, consider it is imperative to use the knowledge embodied in the domain taxonomy, of which the keywords are part, for further interpretation of the clusters. Specifically, given a fuzzy cluster of taxonomy leaves, we propose to find a most specific generalization of that in higher ranks of the taxonomy and use thus obtained higher ranks concept(s) as a general description of the cluster. To this end, we develop a method for finding the most parsimonious generalization of fuzzy leaf clusters. Therefore, our method consists of the following stages:

1. Obtaining a domain taxonomy.
2. Obtaining a collection of research papers in the domain.
3. Obtaining a matrix of relevance scores between taxonomy leaf topics and the papers.
4. Finding thematic fuzzy clusters of "co-relevant" taxonomy leaf topics.
5. Finding most parsimonious generalizations of (some of) the thematic fuzzy clusters.
6. Making conclusions out of the generalizations.

We apply this strategy to two collections of research papers in Data Science that we have downloaded using different criteria. We use a taxonomy of Data Science derived by us [12, 7] from the most popular Computer Science taxonomy, manually developed by the world-wide Association for Computing Machinery in 2012 as the ACM Computing Classification System (ACM-CCS) [2]. Our generalizations and interpretations of the two sources are mutually consistent. Moreover, they cannot be found with the existing approaches because they are based on different levels of conceptual granularity, whereas other approaches involve the same granularity level.

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 Data Science, that can be discerned from two sets of research papers: (a) about 18000 research papers published by the Springer Publishers in data science 17 journals for the past 20 years, and (b) about 27000 research papers published in 80 data science journals by Springer and Elsevier, and retrieved using 17 query terms such as "clustering" and "artificial intelligence". 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 4 concludes the paper.

## 2 Parsimoniously lifting a fuzzy thematic cluster: model and method

Mathematically, a taxonomy is a rooted tree whose nodes are annotated by taxonomy topics. The problem of our concern is this. 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 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 Figure 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. Gaps at lifting 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. Another lifting decision is illustrated in Figure 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 general concept at 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 would make it more preferable if the relative weight of an offshoot is less than the total relative weight of three gaps.

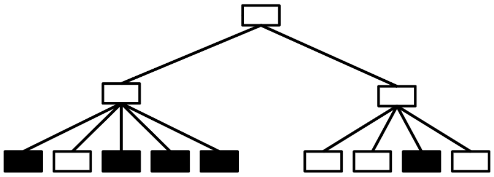

**Fig. 1.** A crisp query set, shown by black boxes, to be conceptualized in the taxonomy.

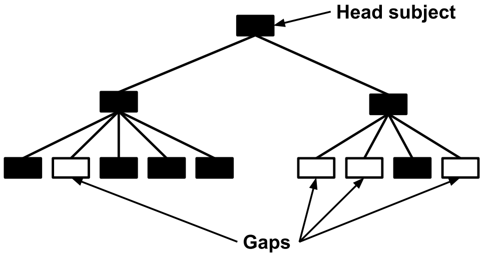

**Fig. 2.** Generalization of the query set from Figure 1 by mapping it to the root, with the price of four gaps emerged at the lift.

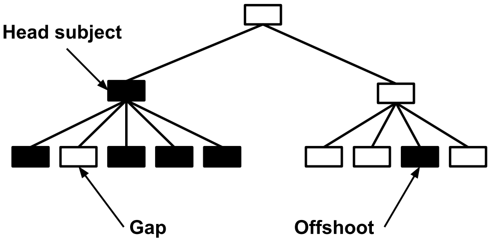

**Fig. 3.** Generalization of the query set from Figure 1 by mapping it to the root of the left branch, with the price of one gap and one offshoot emerged at this lift.

We are interested to see whether a fuzzy set $S$ can be generalized by a node $h$ from higher ranks of the taxonomy, so that $S$ can be thought of as falling within the framework covered by the node $h$. 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" $h$ 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. We give here a short introduction to the solution proposed by the authors in [8].

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 $I$. 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 Figures 2 and 3. Altogether, the total number of head subjects, gaps, and offshoots is 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.

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:

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

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 $u$. Our algorithm ParGenFS [8] recursively computes $H(t)$, $L(t)$ and $p(t)$ from the corresponding values for the child nodes in $\chi(t)$. 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 $t$. To obtain a parsimonious lift, whichever case gives the smaller value of $p(t)$ is chosen. 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. The algorithm ParGenFS leads to an optimal lifting indeed [8].

## 3 Application to collections of research papers

This section describes application of the method described above.

### 3.1 Scholarly text collection

We have downloaded two collections: (a) a collection of 17685 research papers together with their abstracts published in 17 Data Science journals by the Springer Publisher in 1998-2017, see [8] (Collection A); (b) a collection of 26 799 research papers published in 80 Data Science journals by the Springer and Elsevier Publishers and retrieved by using such keywords as clustering, machine learning, deep learning, artificial intelligence, etc. as queries (Collection B). We use abstracts to these papers.

### 3.2 DST Taxonomy

Taxonomy is a form of knowledge engineering which is getting more and more popular. 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 domain of our choice is Data Science, comprising such areas as machine learning, data mining, data analysis, big data, computational intelligence, 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 [12]. The higher ranks of the taxonomy are presented in Table 2 and its full version in [7].

**Table 2.** 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 [4]. These, however, rely on individual words and text preprocessing. We utilize an in-house method [6, 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 $x = x_1 x_2 \dots x_N$ 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. We use efficient versions of AST building algorithms (see, for example, [10]).

Having an AST $T$ built, one can score the string-to-document relevance over the AST as the average frequency of a symbol conditioned by the previous substring coinciding in both the string nd document [8].

### 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 our FADDIS algorithm developed specifically for finding thematic clusters [11]. 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 $A$ for 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 $v = 1, 2, \dots, V = 17685$ or $V = 26799$ at Collection A or B, respectfully. 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 $v = 1, 2, \ldots, V$. A topic is considered relevant to $v$ if its relevance score is greater than 0.2 (a threshold found experimentally, see [6]). Our algorithm, FADDIS, [11] finds fuzzy clusters one by one under the assumption that each of the clusters is represented by its fuzzy membership vector $\boldsymbol{u} = (u_t)$, $t \in T$, where $T$ is the leaf set of our taxonomy so that the product $(\mu u_t)(\mu u_{t'}) = \mu^2 u_t u_{t'}$ approximates $a_{tt'}$ as closely as possible. Here $\mu_k$ stands for the cluster's intensity value determined according to the approximation task [11].

**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, at Collection A, 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 3.

**Table 3.** 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 |

### 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 $\gamma = 0.9$.

Lifting Cluster L gave three head subjects: machine learning, machine learning theory, and learning to rank. These represent the structure of the general concept “Learning” according to text Collection A.

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 consideration. Lifting of Cluster C leads to 16 (!) head subjects/offshoots at which the core clustering subjects are supplemented by methods and environments in the cluster – demonstrating in this way that the ever increasing role of clustering activities should be better reflected in the taxonomy.

### 3.6 Fuzzy clusters at Collection B

Among many fuzzy clusters found by FADDIS algorithm among the DTS taxonomy over the Collection B, there are seven interpretable clusters. These are described in Table 4.

**Table 4.** Generalizations of interpretable clusters found at the Collection B. Symbol ⊙ denotes an offshoot.

| Interpretation | Head subjects and offshoots | Gaps | Leaves |
|---|---|---|---|
| “Learning” | 1.1.1. – Machine learning theory<br>5.2. – Machine learning<br>⊙ 3.4.4.5. – Learning to rank | 38 | 32 |
| “Clustering” | 3.2.1.4. – Clustering<br>and 8 offshoots | 0 | 17 |
| “Probabilistic representations” | 2.1.1. – Probabilistic representations<br>5.2.1.2. – Unsupervised learning<br>5.2.3.5. – Learning in probabilistic graphical models<br>and 8 offshoots | 11 | 31 |
| “Retrieval” | 3.1.4. – Query languages<br>3.4. – Information retrieval<br>⊙ 5.1.1.9. – Language resources | 27 | 28 |
| “Structuring” | 3.1.1.5. – Data model extensions<br>5.1.3. – Computer vision<br>⊙ 1.1.1.12. – Structured prediction<br>⊙ 3.1.4.1.1. – Structured Query Language<br>⊙ 3.4.1.1. – Document structure<br>⊙ 3.4.2.1. – Query representation<br>⊙ 3.4.7.1.1. – Structured text search<br>⊙ 5.2.1.1.5. – Structured outputs<br>and 11 other offshoots | 11 | 34 |
| “Computer vision representations” | 5.1.3.2. – Computer vision representations<br>⊙ 4.1.4.1. – Visualization toolkits<br>and 3 more offshoots | 0 | 13 |
| “Querying” | 3.1.3.2. – Database query processing<br>3.4.2. – Information retrieval query processing<br>and 5 offshoots more | 3 | 15 |

The first two of them one-to-one correspond to clusters L and C over collection A, whereas the third cluster over A, R (Retrieval), corresponds to five other clusters over Collection B. These five are not incompatible with Cluster R, but rather appear to be its facets. The “Computer vision” head subject over A, has received now two complementary aspects: “Structuring” and “Computer vision representations”.

### 3.7 Making conclusions

One can see that the topic clusters found with the text collections do highlight areas of soon-to-be developments. One cannot help but relate them 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.

Most impressive here is the information retrieval cluster R. Rather than conventionally relating the term “information” to texts only, visuals are becoming parts of the concept of information. 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-related five clusters linked to data representation and 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 many 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 dawn of the Data Science era clustering was usually considered a more-or-less auxiliary part of machine 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.

## 4 Conclusion

The paper describes a hybrid method for the analysis of a collection of research papers based on a domain taxonomy. The method involves the following original developments by the authors:

i A taxonomy of Data Science derived from ACM-CCS 2012;
ii A method for scoring relevance between taxonomy leaf topics and texts which requires no manually texts pre-processing;
iii A spectral method for one-by-one deriving fuzzy clusters of taxonomy leaf topics;
iv A method for parsimoniously generalization of fuzzy leaf clusters in the taxonomy;
v Consistent conclusions of tendencies of research in Data Science.

## References

1. Amado, A., Cortez, P., Rita, P., Moro, S.: Research trends on big data in marketing: A text mining and topic modeling based literature analysis. European Research on Management and Business Economics **24**(1), 1–7 (2018)
2. Association for Computing Machinery (ACM): The 2012 ACM computing classification system (2012), http://www.acm.org/about/class/2012
3. Ba, Z., Cao, Y., Mao, J., Li, G.: A hierarchical approach to analyzing knowledge integration between two fields—a case study on medical informatics and computer science. Scientometrics **119**(3), 1455–1486 (2019)
4. Blei, D.M.: Probabilistic topic models. Communications of the ACM **55**(4), 77–84 (2012)
5. Chen, C., Ibekwe-SanJuan, F., Hou, J.: The structure and dynamics of cocitation clusters: A multiple-perspective cocitation analysis. Journal of the American Society for information Science and Technology **61**(7), 1386–1409 (2010)
6. Chernyak, E., Mirkin, B.: Refining a taxonomy by using annotated suffix trees and wikipedia resources. Annals of Data Science **2**(1), 61–82 (2015)
7. Frolov, D., Mirkin, B., Nascimento, S., Fenner, T.: Finding an appropriate generalization for a fuzzy thematic set in taxonomy. Tech. rep., Moscow, Russia (2018)
8. Frolov, D., Nascimento, S., Fenner, T., Mirkin, B.: Parsimonious generalization of fuzzy thematic sets in taxonomies applied to the analysis of tendencies of research in data science. Information Sciences **512**, 595–615 (2020)
9. Galke, L., Melnychuk, T., Seidlmayer, E., Trog, S., Förstner, K.U., Schultz, C., Tochtermann, K.: Inductive learning of concept representations from library-scale bibliographic corpora. INFORMATIK 2019: 50 Jahre Gesellschaft für Informatik–Informatik für Gesellschaft (2019)
10. Grossi, R., Vitter, J.S.: Compressed suffix arrays and suffix trees with applications to text indexing and string matching. SIAM Journal on Computing **35**(2), 378–407 (2005)
11. Mirkin, B., Nascimento, S.: Additive spectral method for fuzzy cluster analysis of similarity data including community structure and affinity matrices. Information Sciences **183**(1), 16–34 (2012)
12. Mirkin, B., Orlov, M.: Three aspects of the research impact by a scientist: measurement methods and an empirical evaluation. In: Optimization, Control, and Applications in the Information Age, pp. 233–259. Springer (2015)
13. Pampapathi, R., Mirkin, B., Levene, M.: A suffix tree approach to anti-spam email filtering. Machine Learning **65**(1), 309–338 (2006)
14. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information processing & management **24**(5), 513–523 (1988)
15. 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)