Globally Optimal Parsimoniously Lifting a Fuzzy Query Set Over a Taxonomy Tree

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

[1] Department of Data Analysis and Artificial Intelligence, National Research University Higher School of Economics, Moscow, Russian Federation
dfrolov@hse.ru

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

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

Abstract. This paper presents a relatively rare case of an optimization problem in data analysis to admit a globally optimal solution by a recursive algorithm. We are concerned with finding a most specific generalization of a fuzzy set of topics assigned to leaves of domain taxonomy represented by a rooted tree. The idea is to “lift” the set to its “head subject” in the higher ranks of the taxonomy tree. The head subject is supposed to “tightly” cover the query set, possibly bringing in some errors, either “gaps” or “offshoots” or both. Our method globally minimizes a penalty function combining the numbers of head subjects and gaps and offshoots, differently weighted. We apply this to a collection of 17645 research papers on Data Science published in 17 Springer journals for the past 20 years. We extract a taxonomy of Data Science (TDS) from the international Association for Computing Machinery Computing Classification System 2012. We find fuzzy clusters of leaf topics over the text collection, optimally lift them to head subjects in TDS, and comment on the tendencies of current research following from the lifting results.

Keywords: Hierarchical taxonomy · Parsimony · Generalization · Additive fuzzy cluster · Spectral clustering · Annotated suffix tree

1 Introduction

The issue of automation of structurization and interpretation of digital text collections is of ever-growing importance because of both practical needs and theoretical necessity. This paper concerns an aspect of this, modeling generalization as a unique feature of human cognitive abilities. The existing approaches to computational analysis of structure of text collections usually involve no generalization as a specific aim. The most popular tools for structuring text collections are cluster analysis and topic modelling. Both involve items of the same

H. A. Le Thi et al. (Eds.): WCGO 2019, AISC 991, pp. 779–789, 2020.
https://doi.org/10.1007/978-3-030-21803-4_78

level of granularity as individual words or short phrases in the texts, thus no generalization as an explicitly stated goal.

Nevertheless, the hierarchical nature of the universe of meanings is reflected in the flow of publications on text analysis. We can distinguish between at least three directions at which the matter of generalization is addressed.

First of all, there are activities related to developing taxonomies, especially those involving hyponymic/hypernymic relations (see, for example, [11,14], and references therein). A recent paper [12] is devoted to supplementing a taxonomy with newly emerging research topics.

Another direction is part of conventional activities in text summarization. Usually, summaries are created using a rather mechanistic approach of sentence extraction. There is, however, also an approach for building summaries as abstractions of texts by combining some templates such as subject-verb-object (SVO) triplets (see, for example, [5]).

One more direction is what can be referred to as "operational" generalization: the authors use generalized case descriptions involving taxonomic relations between generalized states and their parts to achieve a tangible goal such as improving characteristics of text retrieval (see, for example, [8,13].)

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

The rest of the paper is organized accordingly. Section 2 presents a mathematical formalization of the generalization problem as of parsimoniously lifting of a given fuzzy leaf set to higher ranks of the taxonomy and provides a recursive algorithm leading to a globally optimal solution to the problem. Section 3 describes an application of this approach to deriving tendencies in development of the data science, that are discerned from a set of about 18,000 research papers published by the Springer Publishers in 17 journals related to Data Science for the past 20 years. Its subsections describe our approach to finding and generalizing fuzzy clusters of research topics. In the end, we point to tendencies in the development of the corresponding parts of Data Science, as drawn from the lifting results.

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

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

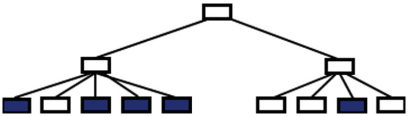

Fig. 1. A crisp query set, shown by black boxes, to be conceptualized in the taxonomy.

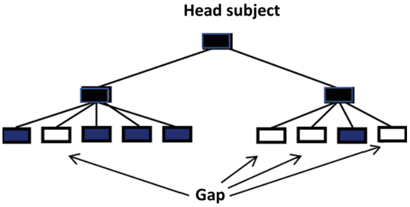

Fig. 2. Generalization of the query set from Fig. 1 by mapping it to the root, with the price of four gaps emerged at the lift.

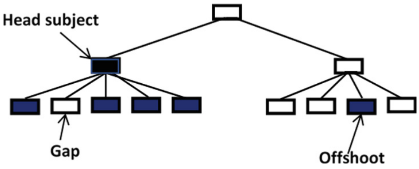

Fig. 3. Generalization of the query set from Fig. 1 by mapping it to the root of the left branch, with the price of one gap and one offshoot emerged at this lift.

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

We are interested to see whether a fuzzy set S can be generalized by a node t from higher ranks of the taxonomy, so that S can be thought of as falling within the subtree rooted at the node t. The goal of finding an interpretable pigeonhole for S within the taxonomy can be formalized according to the Maximum Parsimony (MP) principle: find one or more "head subjects" t to cover S with the minimum number of the elements introduced at the generalization: head subjects and gaps and offshoots.

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 all 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 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 set u defined on the set of leaves I of the tree T, one may 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 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 of u precisely, are false positives and false negatives, referred to in this paper, as *gaps* and *offshoots*, respectively. They are illustrated in Figs. 2 and 3. Given a head subject node h, a gap is a node t covered by h but not belonging to the base of u, so that $u(t) = 0$. In contrast, an offshoot is a node t such that $u(t) > 0$ but not covered by h. Altogether, the total number of head subjects, gaps, and offshoots has to be as small as possible. To this end, we introduce a penalty for each of these elements. Assuming for the sake of simplicity, that the black box leaves on Fig. 1 have membership function values equal to unity, one can easily see that the total penalty at the head subject raised to the root (Fig. 2) is equal to $1 + 4\lambda$ where 1 is the penalty for a head subject and λ, the penalty for a gap, since the lift on Fig. 2 involves one head subject, the root, and four gaps, the blank box leaves. Similarly, the penalty for the lift on Fig. 3 to the root of the left-side subtree is equal to $1 + \gamma + \lambda$ where γ is the penalty for an offshoot, as there is one copy of each, head subject, gap, and offshoot, in Fig. 3. Therefore, depending on the relationship between γ and λ either lift on Fig. 2 or lift on Fig. 3 is to be chosen. That will be the former, if $3\lambda < \gamma$, or the latter, if otherwise.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

ParGenFS Algorithm

```
– INPUT: $u$, $T$
– OUTPUT: $H = H(root)$, $L = L(root)$, $p = p(root)$
I Base Case
  for each leaf $i \in I$
    if $u(i) > 0$
      $H(i) = \{i\}$, $L(i) = \oslash$, $p(i) = \gamma u(i)$
    else
      $H(i) = \oslash$, $L(i) = \oslash$, $p(i) = 0$
II Recursion
    if $u(t) + \lambda V(t) \leq \sum_{w \in \chi(t)} p(w)$
      $H(t) = \{t\}$, $L(t) = G(t)$, $p(t) = u(t) + \lambda V(t)$
    else
      $H(t) = \bigcup_{w \in \chi(t)} H(w)$, $L(t) = \bigcup_{w \in \chi(t)} L(w)$, $p(t) = \sum_{w \in \chi(t)} p(w)$
```

The algorithm ParGenFS leads to an optimal lifting indeed:

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

Proof. We prove this result by induction over 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, and 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.

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 (1). Clearly, 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 it is in the algorithm. The result now follows by induction on n.

3 Structuring and Generalizing a Collection of Research Papers

To apply the ParGenFS algorithm, we follow the steps described below.

3.1 Scholarly Text Collection

We downloaded a collection of 17685 research papers together with their abstracts published in 17 journals related to Data Science for 20 years from 1998–2017. We take the abstracts to these papers as a representative collection.

3.2 DST Taxonomy

We consider a taxonomy of Data Science, comprising such areas as machine learning, data mining, data analysis, etc. We take that part of the ACM-CCS 2012 taxonomy, which is related to Data Science, and add a few leaves related to more recent Data Science developments. The taxonomy under consideration is presented, for example, in [4].

3.3 Scoring the 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 [10] and probabilistic text model approach [2]. These, however, rely on individual words and text preprocessing. We utilize a method, first developed by Pampapathi et al. [9] and further advanced in [3], the AST method for evaluating keyphrase-to-text relevance score using purely string frequency information. An advantage of the method is that it requires no manual work, but works rather reliably, as claimed by these authors.

3.4 Deriving Fuzzy Clusters of Taxonomy Topics

Clusters of topics should reflect co-occurrence of topics: the greater the number of texts to which both t and t' topics 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 at our data. Unfortunately, no satisfactory results have been found. Therefore, we present here results obtained with the FADDIS algorithm developed in [7] specifically for finding thematic clusters. This algorithm implements assumptions that are relevant to the task:

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

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

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

Co-Relevance Topic-to-Topic Similarity Score Given a keyphrase-to-document matrix R of relevance scores 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 $v = 1, 2, \ldots, V = 17685$. The inner product is moderated by a natural weighting factor assigned to texts in the collection. The weight of text v is defined as the ratio of the number of topics n_v relevant to it and n_{max}, the maximum n_v over all v = 1,2,...,V. A topic is considered relevant to v if its relevance score is greater than 0.2 (a threshold found experimentally, see [3]).

FADDIS Thematic Clusters After computing the 317 × 317 topic-to-topic co-relevance matrix, converting in to a topic-to-topic Lapin transformed similarity matrix, and applying FADDIS clustering, we sequentially obtained 6 clusters, of which three clusters are obviously homogeneous. They relate to 'Learning', 'Retrieval', and 'Clustering'. These clusters, L, R, and C, respectively, are presented in Table 1.

3.5 Results of Lifting Clusters L, R, and C

The clusters above are lifted in the DST taxonomy using ParGenFS algorithm with the gap penalty $\lambda = 0.1$ and off-shoot penalty $\gamma = 0.9$ defined to correspond specifics of the DST tree.

The results of lifting of Cluster L are shown in Fig. 4. There are three head subjects: Machine Learning, Machine Learning Theory, and Learning to Rank.

Table 1. Clusters L, R, C: topics with largest membership values.

Cluster L			Cluster R			Cluster C		
$u(t)$	Code	Topic	$u(t)$	Code	Topic	$u(t)$	Code	Topic
0.300	5.2.3.8	Rule learning	0.211	3.4.2.1	Query representation	0.327	3.2.1.4.7	Biclustering
0.282	5.2.2.1	Batch learning	0.207	5.1.3.2.1	Image representations	0.327	3.2.1.4.7	Biclustering
0.276	5.2.1.1.2	Learning to rank	0.194	5.1.3.2.2	Shape representations	0.286	3.2.1.4.3	Fuzzy clustering
0.217	1.1.1.11	Query learning				0.248	3.2.1.4.2	Consensus clustering
0.216	5.2.1.3.3	Apprenticeship learning				0.220	3.2.1.4.6	Conceptual clustering
		...			...			...

These represent the structure of the general concept “Learning” according to the text collection under consideration. One can see from these head subjects that main work here still concentrates on theory and method rather than applications.

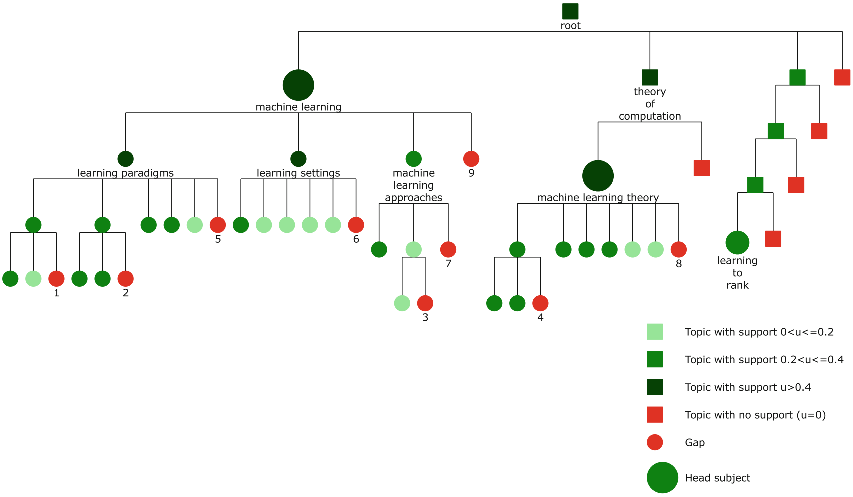

Fig. 4. Lifting results for Cluster L: Learning. Gaps are numbered.

Similar comments can be made with respect to results of lifting of Cluster R: Retrieval. The obtained head subjects: Information Systems and Computer Vision show the structure of “Retrieval” in the set of publications under considerations. We can clearly see the tendencies of the contemporary stage of the process. Rather than relating the term “information” to texts only, as it was in the previous stages of the process of digitalization, visuals are becoming parts of the concept of information.

For the results of lifting of Cluster C the corresponding taxonomy fragment is too large, whereas the lifting results are too fragmentary. 16 (!) head subjects was obtained: clustering, graph based conceptual clustering, trajectory clustering, clustering and classification, unsupervised learning and clustering, spectral methods, document filtering, language models, music retrieval, collaborative search, database views, stream management, database recovery, mapreduce languages, logic and databases, language resources. As one can see, the core clustering subjects are supplemented by methods and environments in the cluster – this shows that the ever increasing role of clustering activities perhaps should be better reflected in the taxonomy. At the beginning of the Data Science era, a few decades ago, clustering was usually considered a more-or-less auxiliary part of machine learning, the unsupervised learning. Perhaps, soon we are going to see a new taxonomy of Data Science, in which clustering is not just an auxiliary instrument but rather a model of empirical classification, a big part of the knowledge engineering. When discussing the role of classification as a knowledge engineering phenomenon, one encounters three conventional aspects of classification: structuring the phenomena; relating different aspects of phenomena to each other; and shaping and keeping knowledge of phenomena. Each of them can make a separate direction of research in knowledge engineering.

References

1. The 2012 ACM Computing Classification System. http://www.acm.org/about/class/2012. Accessed 30 Apr 2018
2. Blei, D.: Probabilistic topic models. Commun. ACM **55**(4), 77–84 (2012)
3. Chernyak, E.: An approach to the problem of annotation of research publications. In: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, pp. 429–434. ACM (2015)
4. Frolov, D., Mirkin, B., Nascimento, S., Fenner, T.: Finding an appropriate generalization for a fuzzy thematic set in taxonomy. Working paper WP7/2018/04, Moscow, Higher School of Economics Publ. House, 58 p. (2018)
5. Lloret, E., Boldrini, E., Vodolazova, T., MartÃnez-Barco, P., Munoz, R., Palomar, M.: A novel concept-level approach for ultra-concise opinion summarization. Expert. Syst. Appl. **42**(20), 7148–7156 (2015)
6. Mei, J.P., Wang, Y., Chen, L., Miao, C.: Large scale document categorization with fuzzy clustering. IEEE Trans. Fuzzy Syst. **25**(5), 1239–1251 (2017)
7. Mirkin, B., Nascimento, S.: Additive spectral method for fuzzy cluster analysis of similarity data including community structure and affinity matrices. Inf. Sci. **183**(1), 16–34 (2012)
8. Mueller, G., Bergmann, R.: Generalization of workflows in process-oriented case-based reasoning. In: FLAIRS Conference, pp. 391–396 (2015)
9. Pampapathi, R., Mirkin, B., Levene, M.: A suffix tree approach to anti-spam email filtering. Mach. Learn. **65**(1), 309–338 (2006)
10. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Inf. Process. Manag. **25**(5), 513–523 (1998)
11. Song, Y., Liu, S., Wang, H., Wang, Z., Li, H.: Automatic taxonomy construction from keywords. US Patent No. 9,501,569. Washington, DC, US Patent and Trademark Office (2016)

12. Vedula, N., Nicholson, P.K., Ajwani, D., Dutta, S., Sala, A., Parthasarathy, S.: Enriching taxonomies with functional domain knowledge. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 745–754. ACM (2018)
13. Waitelonis, J., Exeler, C., Sack, H.: Linked data enabled generalized vector space model to improve document retrieval. In: Proceedings of NLP & DBpedia 2015 Workshop in Conjunction with 14th International Semantic Web Conference (ISWC), vol. 1486. CEUR-WS (2015)
14. 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)