Data & Knowledge Engineering 72 (2012) 103–125

Contents lists available at SciVerse ScienceDirect

Data & Knowledge Engineering

journal homepage: www.elsevier.com/locate/datak

Editorial

Narrative-based taxonomy distillation for effective indexing of text collections ☆

Mario Cataldi [a], K. Selçuk Candan [b,*], Maria Luisa Sapino [a]

[a] Università di Torino, Italy
[b] Arizona State University, Tempe, AZ 85283, USA

ARTICLE INFO

Article history:
Received 1 October 2010
Accepted 23 September 2011
Available online 7 October 2011

Keywords:
Metadata
Information Retrieval and Filtering
Taxonomy Summarization
Taxonomy Classification

ABSTRACT

Taxonomies embody formalized knowledge and define aggregations between concepts/categories in a given domain, facilitating the organization of the data and making the contents easily accessible to the users. Since taxonomies have significant roles in data annotation, search and navigation, they are often carefully engineered. However, especially in domains, such as news, where content dynamically evolves, they do not necessarily reflect the content knowledge. Thus, in this paper, we ask and answer, in the positive, the following question: "*is it possible to efficiently and effectively adapt a given taxonomy to a usage context defined by a corpus of documents*?"

In particular, we recognize that the primary role of a taxonomy is to describe or *narrate* the natural relationships between concepts in a given document corpus. Therefore, a corpus-aware adaptation of a taxonomy should essentially distill the structure of the existing taxonomy by appropriately segmenting and, if needed, summarizing this narrative relative to the content of the corpus. Based on this key observation, we propose *A Narrative Interpretation of Taxonomies for their Adaptation* (ANITA) for re-structuring existing taxonomies to varying application contexts and we evaluate the proposed scheme using different text collections. Finally we provide user studies that show that the proposed algorithm is able to adapt the taxonomy in a new compact and understandable structure.

1. Introduction

While there are many strategies for organizing text documents, hierarchical categorization — usually implemented through a predetermined taxonomical structure — is often the preferred choice. In a taxonomy-based information organization, each category in the hierarchy can index text documents that are relevant to it, facilitating the user in the navigation and access to the available contents. For example, many on-line news aggregators, such as Google News,[1] Yahoo News,[2] and educational web sites, such as NSDL[3] (Fig. 1), present resources in hierarchical structures to help the user locate resources relevant to her interests.

Despite their many advantages as navigation support structures, taxonomies also have certain disadvantages. A key disadvantage is that, for a document collection whose content changes over time, a given initial taxonomy may soon loose its effectiveness in guiding users to relevant documents. In such cases, we can either rely on a domain expert that provides a new taxonomy or consider and revise the existing taxonomy in the light of the new data. In this paper, we aim to address this second alternative.

☆ This work is partially supported by an NSF Grant #1043583 — MiNC: NSDL Middleware for Network- and Context-aware Recommendations.
* Corresponding author. Tel.: +1 480 965 2770; fax: +1 480 965 2751.
E-mail addresses: cataldi@di.unito.it (M. Cataldi), candan@asu.edu (K.S. Candan), mlsapino@di.unito.it (M.L. Sapino).
[1] http://news.google.com.
[2] http://news.yahoo.com.
[3] http://nsdl.org.

0169-023X/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.datak.2011.09.008

Fig. 1. A scientific categorization example (used by NSF's National Science Digital Library web site, http://nsdl.org, to organize digital resources.

In particular, we propose a novel method for distilling a taxonomical domain categorization from an existing one, *within the context of* a given set of text documents that have to be represented and indexed by it.

1.1. Taxonomy adaptation

A taxonomy can be adapted to a new context in many ways. One approach would be to start from a very rich taxonomy and, for each context, identify taxonomy nodes that are not relevant to the current context and eliminate them to obtain a reduced taxonomy consisting of nodes that are contextually relevant. While being simple, this approach assumes that there is broad initial taxonomy, which can be reduced effectively to various different contexts. Researchers, such as [1], have noticed that to capture new contexts, richer adaption strategies, including ones allowing subtrees to move within the given taxonomy, may be needed. In an educational system, for example, this can occur when an initial scientific taxonomy that includes the concept "*entropy*" under ""*thermodynamics*" is used for managing a collection of computer science documents. In this case, due to the content of the collection, it may make sense to move the concept "*entropy*" under "*information theory*". Intuitively, this is due to the fact that knowledge is rarely hierarchical and a given concept may be linked to many others (Fig. 2). Thus, knowledge can be hierarchically organized in different ways by focusing on those relationships that are most relevant in a given context.

The challenge is that in many cases we do not have access to the underlying knowledge graph, but are given an initial taxonomy which is nothing but a single hierarchical-organization of the underlying knowledge graph corresponding to the initial context. Therefore, a contextually relevant adaptation of the initial taxonomy needs to discover the "*missing relationships*" and distill a new and more contextually appropriate structure from the initial taxonomy, possibly also removing concepts that are redundant in the considered context.

1.2. Desiderata

Let us be given an initial taxonomical hierarchy, $H(N,E)$, a document corpus, D, and a target taxonomy size, $k \leq |N|$. The goal is to identify a new taxonomy, $H'(N',E')$, where $|N'| = k$, the concepts in H' come from the concepts in H, and the following desiderata are best satisfied.

1.2.1. Desideratum 1: coverage

H' should reflect the content of the corpus, D. In other words, users should associate as many documents in the given corpus as possible with the nodes of the taxonomy.

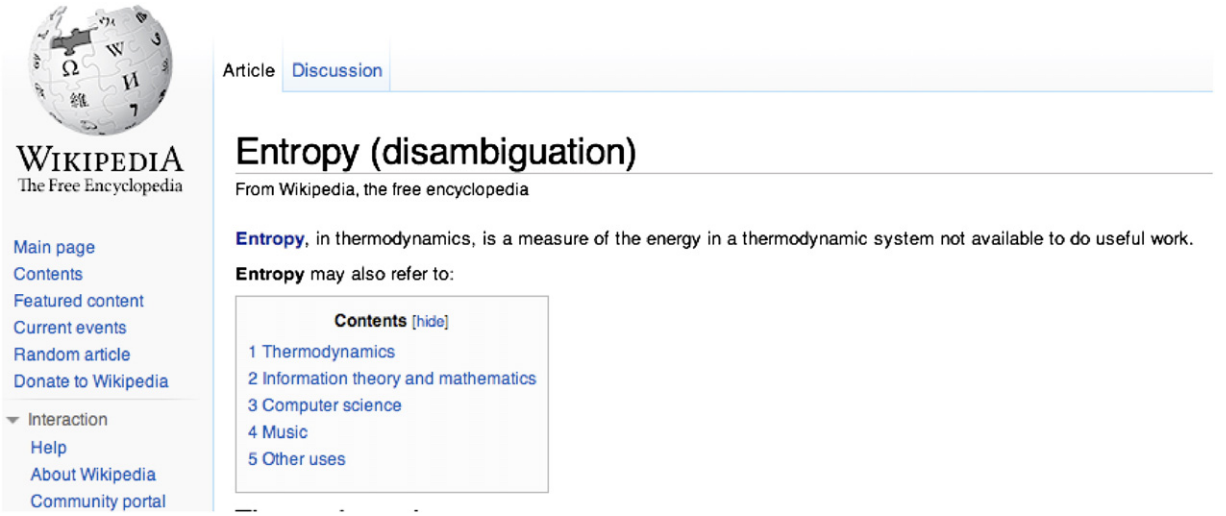


Fig. 2. The possible categorizations of the concept "Entropy" proposed by the Wikipedia categorization system. The concept can be related to different concepts depending on the focus of interest.

1.2.2. Desideratum 2: redundancy

The nodes of the adapted taxonomy, H', should be as distinct as possible and should discriminate the documents in the corpus. This would minimize redundant associations of the documents to the nodes of the taxonomy and, thus, help improve the organization of the documents in the corpus.

1.2.3. Desideratum 3: specificity

If new concept labels need to be introduced (by possibly combining the concepts in H) during the adaptation, then given two alternative adaptations which provide similar coverage and redundancy, the one which present the least ambiguous concept labels to the user is more desirable.

1.3. Contributions of this paper

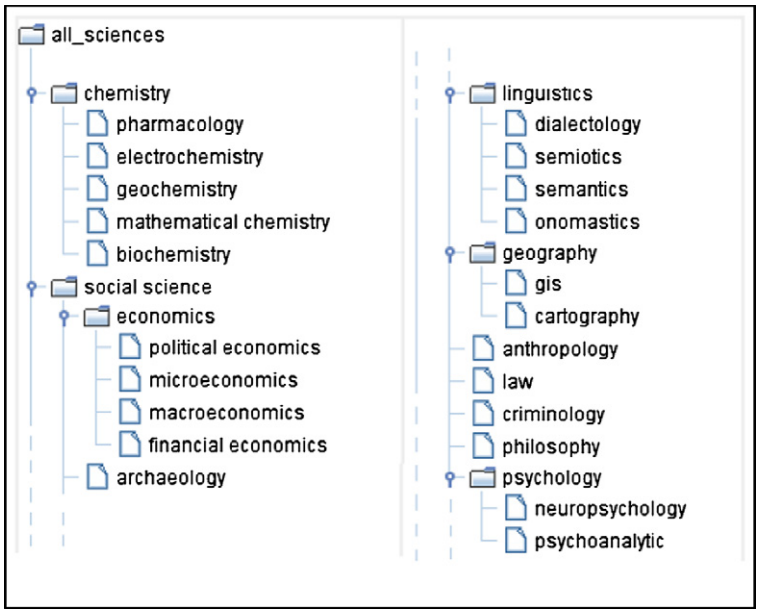
In this paper, we propose "*A Narrative Interpretation for Taxonomy Adaptation* (ANITA)," a novel distillation approach for adapting existing taxonomies to varying application contexts. As we formally define in Section 3.1.1, we view a given taxonomy as a narrative mechanism introducing the (hierarchical) relationships of the relevant concepts. This alternative view, which constrains the knowledge representation into a one-dimensional ordering of concepts (with possible repetitions) helps segment the given taxonomy into groups of concepts that are similar to each other (and thus not discriminative) within the given context. Fig. 3 provides an example visualizing how ANITA adapts a taxonomy based on the context defined by a given collection of documents. In this example, most children of "*chemistry*", except for "*biochemistry*", have been unclassified because they are found to be unnecessary within the context of the NSF data set (described in Section 3.3.2). In contrast, the four children of "*economics*" have been collapsed into a single category node since in the given document context they are found to be useful, but not sufficiently distinguished from each other.

The specific contributions of this paper are as follows:

- *Narrative view of a taxonomy*: as described in Section 3.1.1, we transform each category in the original taxonomy into a *sentence* by associating to each concept *a vector of weighted terms* extracted from the current corpus. Then, we order these sentence-vectors (Section 3.1.2) in such a way to reflect both the semantical relationships among the categories and the structural constraints expressed by the hierarchy.
- *Segmentation of the narrative*: this narrative, which preserves the structure of the taxonomy (e.g., structural-relationships between the concepts), is then segmented based on a narrative-development analysis, highlighting where the narrative significantly drifts from one concept-topic to another (Section 3.2).
- *Re-construction* (*or distillation*) *of an adapted taxonomy based on the segmentation results*: the resulting narrative segments (each describing a group of concepts/categories that collectively act as a single topic) are re-organized into a hierarchical structure, linking each concept-segment to others that are structurally related to it (Section 3.3).

The result of this process is a contextually-relevant *adapted taxonomy*, where details are highlighted where they matter, suppressed where they do not support the context and re-organized to be more adherent to the considered context. In this paper, we extend our preliminary work in [2], where we presented the outlines of the narrative-based approach to dynamically adapt taxonomies to varying domains, and we propose alternative distance-preserving ordering approaches that lead to narratives of different structures. In Section 3.3.2, we evaluate the impact of these different narrative structures on the qualities of the resulting

(a) Original taxonomy fragment



(b) Adapted taxonomy fragment

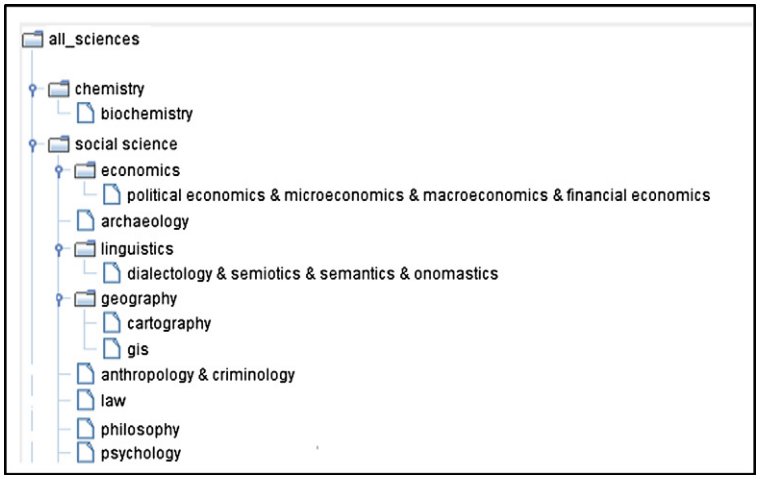


Fig. 3. (a) Scientific taxonomy fragment extracted from DMOZ (accessible at the link http://www.dmoz.org/) and (b) its adaptation based on the context defined by the NSF data set (described in Section 3.3.2).

adaptations. Section 3.3.2 also includes (a) comparisons of the ANITA approach with other alternative methods on different data sets and (b) results from user studies.

2. Related work

Text collections, growing in number and size, are creating new challenges within the data mining community about their organization, summarization, and presentation to the users through extracted hierarchical categorizations.

Ontologies and hierarchical categorizations, when available, can play significant roles in the organization and summarization of data. Consider, for example, snippet generation: today, most major search engines display search results as a ranked list, accompanied by the page titles and small text fragments, or snippets that summarize the content relative to the search keywords. However, statistically generated snippets are not always representative of the documents' contents or related to the query intention and many works tried to optimize their generation [3,4]. In particular, [4] showed that relying on a background ontology for deriving the possible senses a query might have and for selecting the sense that is most likely to represent the query intention, may improve snippet generation.

Alternatively, the hierarchy may not be an input of the process, but may be created on-demand to organize the results. [5,6], for example, focus on generation of hierarchies in order to organize documents retrieved by a search engine. Similarly, in [7], authors propose a hierarchical clustering algorithm to build a topic hierarchy for a collection of documents retrieved in response to a query.

While in this paper, and the rest of this section, we focus on works related to extraction, adaptation, and use of hierarchical organizations for the purpose of organization and presentation of text corpora, we also note that hierarchical organizations can be beneficial not only for organizing documents, but also queries themselves. For example, [8] relies on hierarchical organization of queries for efficient filtering of document streams. Authors propose a two-stage model for information filtering of document streams.

The first "topic filtering" stage quickly filters out the most likely irrelevant documents based on term-based profiles. The second, "pattern taxonomy" based stage, on the other hand, structures query patterns into a taxonomy based on subset relationships and uses this pattern taxonomy to solve the problem of overload during real-time filtering of document streams.

2.1. Automatic extraction of hierarchies

In the literature, many authors tried to automatically extract hierarchical categorizations from text corpora.

[9] presents an overview about the many methodologies that have been proposed to automatically extract structured information from texts (reporting also procedures and metrics for quantitative evaluations). In [10] authors present an unsupervised method to automatically derive from a set of documents a hierarchical organization of concepts (salient words and phrases extracted from the documents), using co-occurrence information. [11] organized the extracted concepts by analyzing the syntactic dependencies of the terms in the considered text corpus. [12] also considers multiple and heterogeneous sources of evidence to improve the taxonomical relations between the selected terms. Many methods rely on preliminary supervised operations to limit the noise in the retrieved concepts: in [13], the user sketches a preliminary ontology for a domain by selecting the vocabulary associated to the desired elements in the ontology (this phase is called lexicalisation).

One way to extract hierarchies is to apply hierarchical clustering on the hierarchy along with the documents. In [14], the centroids of each class are used as the initial seeds and then a projected clustering method is applied to build the hierarchy. In [15] a linear discriminant projection is applied to the data first and then the hierarchical clustering method UPGMA [16] is exploited to generate a binary tree. [17] applies a divisive hierarchical clustering: authors generate a taxonomy where each node is associated to a list of categories. [18] associates word distribution conditioned on classes to each node: the method uses a variance of the EM algorithm to cluster nodes. Similarly, [19] presents a method in which concepts are probabilistically modeled. The probabilistic classes are organized in hierarchies by relying on the KL divergence measure between the probability distributions associated to the concepts.

In the last few years, with the increase of semi-structured information repositories, many other authors tried to leverage the information guided by these sources to reduce the imprecision in the retrieved hierarchies: for example, in [20,21], authors have investigated the problem of automatic knowledge acquisition from Wikipedia repositories. [22] leverages the tag vocabulary extracted by Flickr to induce an ontology by using a subsumption-based model.

Since hierarchies have significant roles in the data annotation, indexing and exploration, they are often carefully designed. [23], for example, makes a distinction between *domain specific ontologies* that capture concepts about a particular application domain and *upper level* ontologies that are domain independent. The authors identify various problems, including functionality, usability, portability, and reliablity, that arise when using upper level ontologies. However, we see that, even for domain specific ontologies or hierarchies, it may be necessary to restructure them in order to better reflect the specific content knowledge (and improve the efficiency of the retrieval process). [24] addresses the problem of how to adapt a topic taxonomy in order to reflect the change of a group's interest to achieve dynamic group profiling. In this work the authors assume that there is a sequence of edits that can lead the adaptation process and aim to identify this sequence of edits. Instead, in our paper, we consider a holistic approach; instead of searching for a sequence of edits, we look for partitions that group the semantically related hierarchy nodes in a given domain.

2.2. Faceted search and navigation

It is widely observed that the standard search interface (most of the times consisting of a text query box and a list of retrieved items) is inadequate for navigation and exploration in large text collections. User interfaces which filter, group and organize retrieval results, on the other hand, have been demonstrated to be preferred by users [25] over the straight result-list model when used for exploratory purposes [26].

A representation known as hierarchical faceted meta-data is becoming popular within the information architecture and enterprise search communities [27]. Faceted search, navigation and browsing [28,29], is a popular information filtering technique for accessing a data collection represented using a faceted classification. A faceted classification system allows the assignment of multiple classifications to a data item, enabling the classifications to be ordered in multiple ways, rather than in a single, predetermined, taxonomic order. But, a considerable impediment to this meta-data approach (and therefore, the hierarchical faceted meta-data) is the need to create a hierarchy that can effectively organize the information contents. For this reason, usually, the meta-data structures are manually created by information architects [30]. While manually created meta-data is considered of high quality, it is costly in terms of time and effort to produce, which makes it difficult to scale and keep up with the vast amounts of new content being produced. In this paper, we describe ANITA, an algorithm that makes a considerable progress in automating

meta-data adaptation. In fact, ANITA permits to generate domain-specific hierarchies by starting from existing generic taxonomical meta-data structure.

2.3. Evaluation of taxonomies

Evaluation of the quality of automatically generated taxonomies is a very important and non-trivial task. In the literature, many evaluation measures have been introduced. In [31], authors determine the precision of the clustering algorithm by manually assigning a relevance judgment to the documents associated to the clusters. In [32], authors use the F-Score to evaluate the accuracy of the document associations (but the approach requires a ground truth, which is hard to determine in many cases). In [33] authors perform a user study to evaluate the qualities of the relationships between concepts and their children and parent concepts. In [5], authors measure the quality of the concepts by evaluating their ability to find documents within the hierarchy (the "reach time" criterion measures the time taken to find a relevant document). In this paper, we use similar objective metrics as well as subjective user studies to evaluate our distillation approach.

3. Narrative-driven taxonomy adaptation process

Given an input taxonomy $H(C,E)$ (also called hierarchy in the paper) defined as a directed tree, where $C=\{c_1,\ldots,c_n\}$ represents the set of n nodes (also called concepts or categories) and E is the set of structural directed edges (where each edge represents an ISA relationship between two nodes in C), our goal is to create an adapted taxonomy $H'(C',E')$, based on a given context defined by a corpus, D, of text documents.

As described before, ANITA relies on a "narrative" interpretation of the input taxonomy to achieve this goal; unlike the original taxonomy, which is hierarchical, the narrative is linear, but created in a way that reflects the structure of the hierarchy.

A (linear) narrative, $N(S)$, where $S=\{s_1,\ldots,s_m\}$ is a sequence of sentences, is a permutation (possibly with repetitions) of the set S. To obtain a narrative representation, $N_H(S)$, of a given hierarchy $H(C,E)$, ANITA represents each concept $c_i \in C$ as a sentence $s_i \in S$, which *describes* the concept c_i in terms of other concepts in the taxonomy as well as relevant concepts emerging from the corpus of interest. To obtain the narrative, $N_H(S)$, ANITA then selects a permutation (possibly with repetitions) that captures both the structural information (coming from the original structure described by E) as well as the content of the considered corpus.

This alternative interpretation, which constrains the knowledge representation into a one-dimensional ordering of concepts (with possible repetitions) helps segment the given taxonomy into groups of concepts that are similar to each other within the given context. Experiments reported in Section 3.3.2 show that ANITA is able to leverage this narrative to improve the effectiveness of the adaptation process with respect to more generic clustering-based approaches, which cannot represent the structural context.

3.1. Step I: narrative view of a taxonomy

In this section, we first introduce the narrative interpretation and then describe the taxonomy adaptation process in detail.

3.1.1. Step Ia: concept-sentences

Whereas a taxonomy is a hierarchy of concept-nodes, a *narrative* is a sequence of sentences. Therefore, in order to create a narrative corresponding to the taxonomy, we need to map concept-nodes of the input taxonomy into *concept-sentences*. What we refer to as concept-sentences are not natural language sentences, but vectors obtained by analyzing the structure of the given taxonomy and the related corpus of documents. Intuitively, these sentence-vectors can be thought of as being analogous to *keyword-vectors* commonly used in representing documents in IR systems.

Concept-sentences associate to each concept a coherent set of semantically related keywords, extracted from the associated text corpus. Thus, for each concept c_i in the considered hierarchy, we associate a sentence-vector $\vec{sv}_{c_i}$ as

$$\vec{sv}_{c_i} = \left\{ w_{i,1}, w_{i,2}, w_{i,3} \cdots w_{i,v} \right\}$$

where v represents the total number of considered terms (the corpus vocabulary and labels in the taxonomy), and $w_{i,j}$ represents the semantical correlations between the j-th term and the i-th taxonomical concept. Fig. 4 shows a sample taxonomy fragment and Table 1 shows the corresponding sentence-vectors which include concepts from the taxonomy as well as keywords from the data set.

Concept-sentences can be obtained in many different ways; [34,11,35,36] propose various approaches that leverage semantic similarities between concepts in a given context for obtaining such vectors. In this paper, we use the approach proposed in [34] (and reported in detail in Appendix A) to associate to each concept a keyword-vector, that integrates terms extracted from text documents and labels of concepts obtained from the considered domain taxonomy. The resulting vectors reflect both the structural context (imposed by the taxonomy) and the documents content (imposed by the corpus).

3.1.2. Step Ib: sentence ordering

After the vector-based encoding of the *concept-sentences*, the next step is the creation of the narrative by selecting a permutation (possibly with repetitions) which captures the structure of the taxonomy as well as the content of the considered corpus.

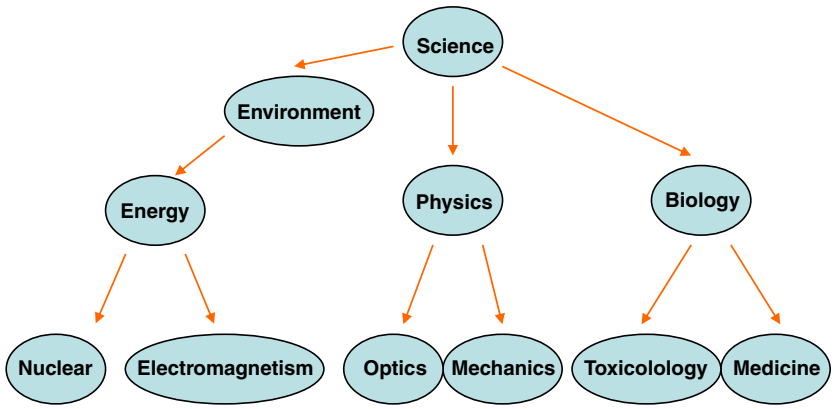


Fig. 4. A portion of a hierarchical meta-data structure about *science*, extracted from DMOZ.

Table 1
(a) The sentence-vectors (sv), referred to the taxonomy fragment in Fig. 4, obtained applying the method described in [34] using the NSF document set (described in Section 3.3.2). The sentence-vectors are ordered based on the corresponding weights which are omitted in the figure for clarity. Terms that are not in bold are picked from the NSF document corpus.

$\overrightarrow{sv}_{science}$	{**science**, **student**, education, **physics**, teacher ···}
$\overrightarrow{sv}_{environ.}$	{**environment**, **science**, ecology, **energy**, earth ···}
$\overrightarrow{sv}_{physics}$	{**physics**, quantum, particle, **mechanics**, theory ···}
$\overrightarrow{sv}_{biology}$	{**biology**, **energy**, genetic, cell, ecology, student, biochemistry ···}
$\overrightarrow{sv}_{energy}$	{**energy**, **environment**, **electromagnetism**, thermodynamics, conservation ···}
$\overrightarrow{sv}_{optics}$	{**optics**, **physics**, light, **science**, radiation ···}
$\overrightarrow{sv}_{mechanics}$	{**mechanics**, **physics**, force, **science**, quantum···}
$\overrightarrow{sv}_{toxicology}$	{**toxicology**, **biology**, department, student, **science** ···}
$\overrightarrow{sv}_{medicine}$	{**medicine**, safety, disease, **science**, policy ···}
$\overrightarrow{sv}_{nuclear}$	{**nuclear**, cell, power, **physics**, particle ···}
$\overrightarrow{sv}_{electromag.}$	{**electromagnetism**, interaction, **physics**, **science**···}

Given a set, $S=\{s_1,\ldots,s_n\}$, of concept sentences corresponding to the input taxonomy $H(C,E)$, where $C=\{c_1,\ldots,c_n\}$, the permutation (with possible repetition) $N_H(S)$ needs to satisfy a set of constraints, implied by the structure of $H(C,E)$, for each (a) ancestor–descendant and (b) sibling concept pairs in C.

3.1.2.1. Ancestor–descendant ordering. In this paper we consider three alternative ancestor–descendant ordering constraints: the pre-order, parenthetical and post-order constraints.

- *Pre-order constraints*: a hierarchy (especially a concept hierarchy) is structured in a way that the most general concept is used as the root of the hierarchy and the most specific ones are the leaves. In a sense, each node provides more specialized knowledge within the context defined by all its ancestors. We leverage this aspect to define a narrative in which the sentences associated to the nodes of the taxonomy are read in pre-order; i.e., given any ancestor and descendant pair in H, the ancestor appears *earlier* in the narrative, N_H.
- *Post-order constraints*: in contrast, in this alternative, given any ancestor and descendant pair in H, the ancestor appears *later* in the narrative, N_H. This alternative generates a narrative in which the different concepts are presented bottom-up: after presenting the most specific concepts, their super-concept is narrated — any super-concept presented after the narration of its children can be seen as summarizing the description of its sub-concepts.
- *Parenthetical constraints*: in this case, given any ancestor and descendant pair in H, the ancestor appears both *before* and *after* the descendant in the narrative, N_H. This implies that the ancestor is repeated twice in the narrative. Intuitively, the parenthetical traversal is analogous to a narrative where each passage is presented with an *introduction* and goes in *details* until a general *conclusion*. In parenthetical traversal of the tree, each parent node is visited twice, representing both the general introduction and the conclusion to the argument that the children specialize.

3.1.2.2. Distance-preserving sibling ordering. While pre-order, post-order and parenthetical traversals of the tree help us decide in which order ancestors and descendants are to be considered, we also need additional information to choose the order in which the siblings in the hierarchy are to be included in the narrative.

Let us consider a node c_0 with m children $\{c_1, c_2 \cdots c_m\}$. Our primary goal is to ensure that the narrative is ordered in a way that reflects the similarities — or dissimilarities — among these m siblings (as well as their parent c_0). In fact, in a narrative, each argument is introduced by smoothly contextualizing its topic (reporting before sentences that introduce it) and drifts to the other topics by introducing and defining the context of the next argument. Therefore, each concept-sentence should be anticipated by the concept-sentence that best introduces it and followed by the concept-sentence that can best deepen its knowledge. For

example, in Fig. 4 "*biology*" has two children, "*toxicology*" and "*medicine*"; if "*biology*" is more semantically related to "*medicine*" than "*toxicology*", we would like to order the narrative in such a way to preserve this information.

For this purpose, we first compute the distance matrix M based on the sentence-vectors corresponding to all $m+1$ concepts (the parent and the m children)[4]:

$$M[i][j] = 1 - sim\left(\vec{s\upsilon}_{c_i}, \vec{s\upsilon}_{c_j}\right).$$

Then, we use a distance-preserving embedding technique to map these concepts onto a one-dimensional ordering. In particular, without loss of generality, we use multi-dimensional scaling (MDS [37]), to embed the concepts onto a 1-dimensional order. MDS works as follows: given as inputs (1) a set of N objects, (2) a matrix of size $N \times N$ containing pairwise distance values and (3) the desired dimensionality k, MDS tries to map each object into a point in the k-dimensional space in such a way that the stress value

$$stress = \sqrt{\frac{\sum_{i,j}\left(d'_{i,j} - d_{i,j}\right)^2}{\sum_{i,j} d_{i,j}^2}},$$

where $d_{i,j}$ is the actual distance between two objects o_i and o_j and $d'_{i,j}$ is the distance between the corresponding points in the resulting k-dimensional space, is minimized. Therefore, by providing as input $N = m+1$ input concepts and $k = 1$ target dimension, the resulting order of concepts would preserve the semantic ordering between the concepts as best as possible. We constrain the stress minimization process in a way that forces the position of c_0 at the beginning of the list. This way, the resulting order of the children concepts will reflect the concept similarities with respect to the position of the parent concept in the narrative.

As an example, let us re-consider the taxonomy fragment presented in Fig. 4. In order to decide in which order the children of "*biology*", "*medicine*" and "*toxicology*", should be included in the narrative, we first calculate a distance matrix, $M_{biology}$, of these three nodes (Table 2(a)). Then, we apply the (slightly modified) MDS algorithm to obtain the ordering of the children with respect to the parent (Table 2(b)): first "*medicine*" is included in the narrative and, then, "*toxicology*".

Fig. 5(a), (b), and (c) show the three distance preserving ordering approaches.

3.2. Step II: segmentation of the narrative

At this point the narrative is a sequence of sentences (or more precisely sentence-vectors), each including the information coming from the structural knowledge (hierarchy) and the context knowledge (documents), defining a global discourse that covers all the topics addressed by the taxonomy, according to the knowledge expressed by the contents.

In the next step, we analyze this narrative to identify segments (or partitions) that are highly correlated. The idea is that if, in the given corpus, two concepts are highly correlated, they may not need two separate nodes in the adapted taxonomy. In contrast, if there is a significant difference between two portions of the narrative, then these two portions (or segments) do necessitate different concepts in the resulting taxonomy. In the literature, there are various techniques for segmenting a narrative into coherent units. Textile [38,39] and Vectile [40] algorithms, for example, plot similarity scores (based on lexical co-occurrence and distribution analysis) of neighboring portions of the text. The dips (i.e., local minima) in the similarity curve correspond to regions of the text where there is a significant change in the content. Therefore, these dips are identified as text segment boundaries.

In this paper, in order to partition the narrative $\vec{s\upsilon}_1, \vec{s\upsilon}_2, \ldots, \vec{s\upsilon}_n$ into coherent segments, we use a similar strategy. More specifically, instead of searching for local minima of similarities, given the desired size, k, of the adapted taxonomy, we seek k partitions with similar high internal coherence (defined in terms of the total amount of topic drift).

1. Given the narrative (i.e., ordered sequence of sentence-vectors), we first compare each pair of neighboring vectors, $\vec{s\upsilon}_i$ and $\vec{s\upsilon}_{i+1}$ $(1 \le i \le n-1)$ by computing their *distance*:

$$\Delta_{i,i+1} = 1 - cos\left(\vec{s\upsilon}_i, \vec{s\upsilon}_{i+1}\right).$$

2. The sequence of vectors is then analyzed for *topic drifting*. We say that a topic drift occurs for a given segment of the narrative when the degree of change between its starting and ending points is above a given threshold. If $Seg_{i,j}$ denotes a segment from the vector $\vec{s\upsilon}_i$ and $\vec{s\upsilon}_j$, the corresponding degree of drift is defined as $drift_{i,j} = \sum_{h=i}^{j-1} \Delta_{h,h+1}$. A segment $S_{i,j}$ is said to be *coherent* if it holds that $drift_{i,j} < \lambda_{max}$, where $\lambda_{max} = \frac{drift_{1,n}}{k}$ is the *coherence threshold*, and k is the target size of the summarized taxonomy.

At the end of the process, we obtain a set of segments, or partitions, $P = \{P_1, P_2, \cdots, P_k\}$ that represent sequences of coherent narrative components. Note that, each partition is a sequence of concepts from the original taxonomy and defines a single concept in the revised taxonomy.[5]

[4] See Appendix B for more details regarding similarity computation.

[5] Notice from Fig. 5(c) that, the parenthetical traversal introduces each parent concept twice; in this case, if a parent node is associated to two different partitions, it is removed from the partition whose drift value (with respect to neighbor nodes in the sequence) is higher.

Table 2
(a) The distance matrix M obtained using the sentence-vectors for the concept-nodes "*biology*", "*medicine*" and "*toxicology*" and (b) the MDS-ordering of the children of "*biology*".

(a)

	biology	toxic.	medicine
biology	0	0.4	0.2
toxicology	0.4	0	0.2
medicine	0.2	0.2	0

(b)

biology	1st
medicine	2nd
toxicology	3rd

Let us reconsider the taxonomy presented in Fig. 4; based on the NSF data corpus (described in Section 3.3.2), the taxonomy is partitioned in four groups of nodes (Fig. 6). The segmentation process also alters the structure of the hierarchy, since the relationships among concepts could change from one domain to another one. For example, in a popular/scientific magazine context, two concepts "*nuclear*" and "*environment*" may be strongly related, while in the context of a scientific professional journal, the concept "*nuclear*" might be more rigorously related to the concept of "*physics*" (in fact, as shown in Fig. 7, when considering the NSF awarded abstracts, "*nuclear*" has been connected to "*physics*"). Therefore, ANITA tries to preserve the original relationships among concepts, but alters the structure when there is sufficient evidence in the corpus that a different structure would reflect the content better.

3.3. Step III: taxonomy distillation from the partitions

In order to construct the adapted taxonomy from the partitions created in the previous step, we need to re-assemble the partitions in the form of a tree structure. Furthermore, for each partition, we need to pick a *label* that will be presented to the user and will describe the concepts in the partition.

3.3.1. Step IIIa: partition linking

The adapted taxonomy, $H'(C',E')$ with $C'=\{c'_1,\ldots,c'_k\}$ (where each node c'_i represents the partition P_i) should preserve the original structure of $H(C,E)$ as much as possible. Thus,

- the root of H' is c_{root} $(1\leq root\leq k)$ such that the corresponding partition P_{root} contains the root node of H.
- Let us consider a pair, P_i and P_j, of partitions in P. The decision on whether (and how) the corresponding concepts c'_i and c'_j should be connected is based on the following analysis. Let $E_{i,j}$ be the set of edges in E linking any concept in P_i to any concept in P_j. Similarly, let $E_{j,i}$ be the set of edges in E linking any concept in P_j to any concept in P_i. With the goal of preserving to the best the structure of H, we measure the strength of the structural constraints implied by E in H, and we propose as our solution the adapted taxonomy which maximally preserves such constraints.
  Let $e=\langle c_a,c_b\rangle$ be an edge in H that connects two different partitions P_i and P_j (i.e. $c_a\in P_i$, $c_b\in P_j$). The strength of the structural constraint e, *strength(e)*, (i.e., the strength of the structural constraints induced by e) is $1+d_b$, being d_b the number of descendants of c_b in H that also belong to P_j. Based on this, the decision of having the corresponding c'_i as the ancestor of c'_j is supported by the strength of the structural constraints associated to the edges in $E_{i,j}$. Thus, the taxonomy H', is constructed by maximally preserving such constraints as follows:

1. create a complete weighted directed graph, $G_P(V_P,E_P,w_P)$, of partitions, where
   - $V_P=P$,
   - E_P is the set of edges between all pairs of partitions, and
   - $w_P(\langle P_i,P_j\rangle)=\sum_{e\in E_{i,j}} strength(e)$;
2. find a *maximum spanning tree* of G_P rooted at the partition P_{root}.

In our running example shown in Fig. 6 the structural constraints imposed by the original hierarchy (dictated by the edges of the taxonomy) may imply that the partition containing the concept "*physics*" (*partition* 3) should be attached to the *partition* 1 (containing the node "*science*") or the *partition* 2 (containing the concept "*environment*"). However, as shown in Fig. 7, the proposed approach decides to attach *partition* 3 to the *partition* 1, because the *strength* of this correlation is higher than the one with *partition* 2 (3 structural constraints vs. 2). In Table 3, the *strength* of all the structural constraints among the partitions retrieved in Fig. 6 is shown.

For example, let us consider the taxonomy fragment and its partitions shown in Fig. 6. In the adapted hierarchy (Fig. 7), ANITA picks as root the partition containing the root node ("*science*"). Then, the remaining three partitions have been attached to it by analyzing the constraints given by the structural original edges.

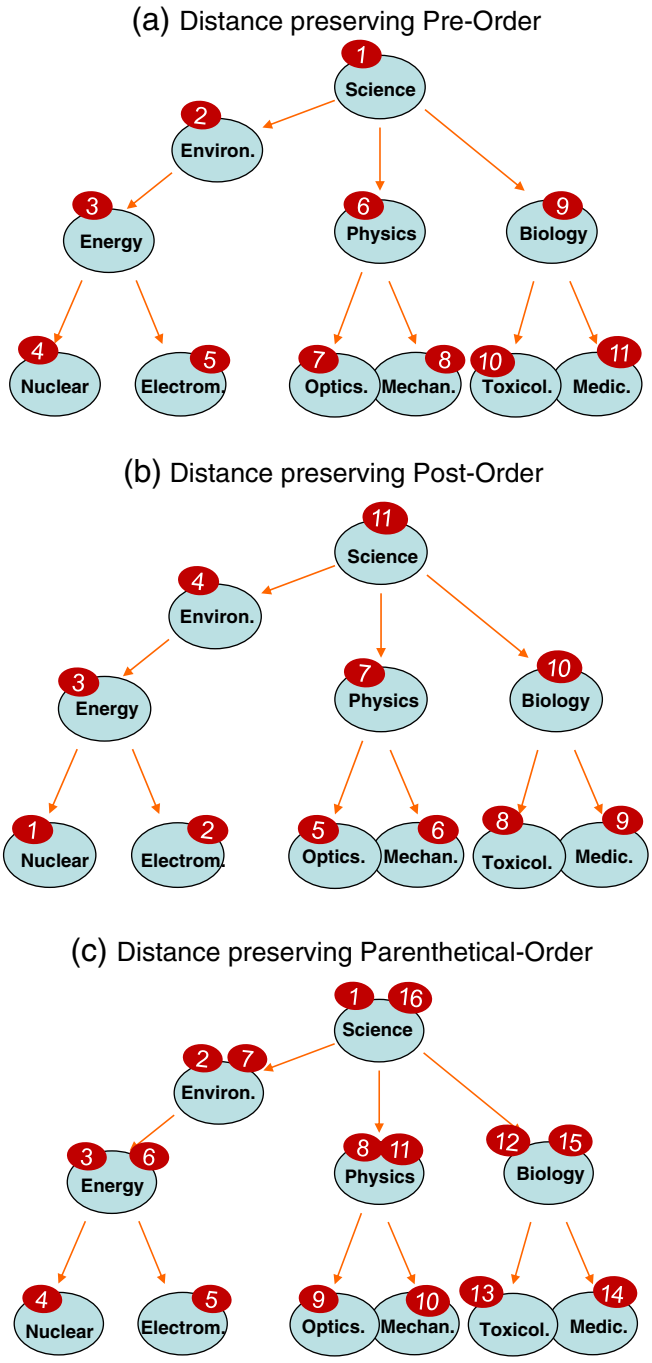


Fig. 5. The narrative order (denoted by the circled number) for the hierarchy presented in Fig. 4 based on a distance preserving pre-order, (b) distance preserving post-order and (c) distance preserving parenthetical ordering approach.

3.3.2. Step IIIb: Partition Labeling

In order to select a representative label for each partition we need to analyze the obtained partitions in the context of the original structure. In order to pick a label for the node c_i' associated to P_i, we consider the structural relationships in the original hierarchy H among the nodes in P_i. If there is a concept $c_l \in P_i$ that dominates all the other nodes in the partition (i.e., $\forall c_j (\neq c_l) \in P_i\ c_j$ is a descendant of c_l), then the label of c_l is selected as the label for c_i'. If there is no such single node, it means that P_i contains a pair of nodes whose least common ancestor in the original hierarchy H does not belong to P_i. Intuitively, the pair belongs to two disjoint subtrees of H in P_i: (a) since these subtrees belong to a single partition, in the given context, the corresponding root concepts are not sufficiently distinguished from each other; on the other hand, (b) none of the root concepts can individually represent the

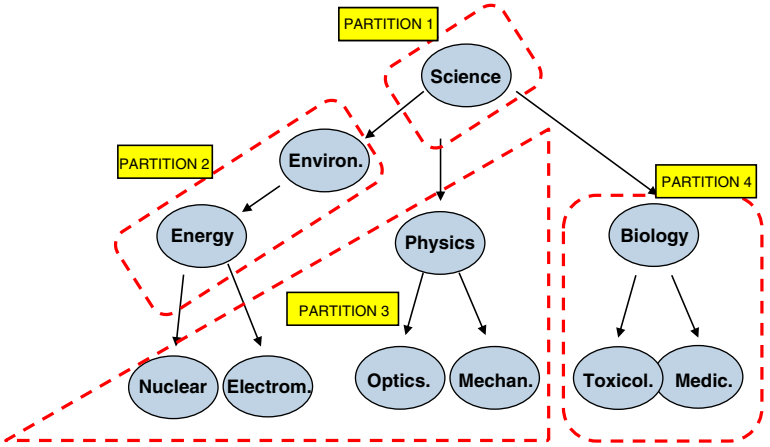


Fig. 6. Narrative-based adaptation of the taxonomy fragment presented in Fig. 4: based on the structural constraints and the available contents, the hierarchy nodes are grouped in 4 partitions.

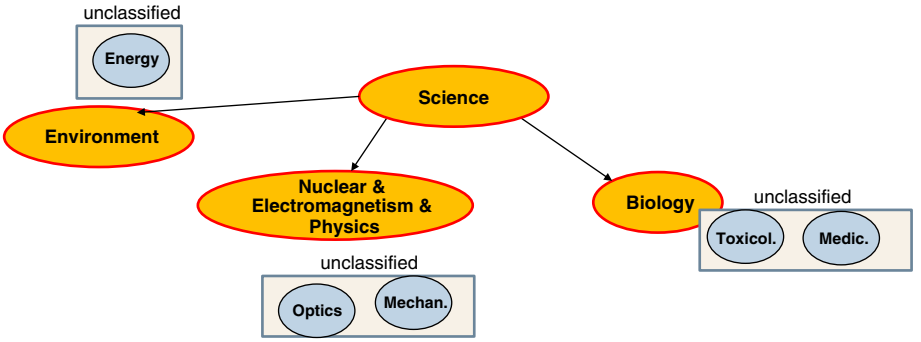


Fig. 7. Taxonomy reconstruction process: based on the partitions shown in Fig. 6(a) the taxonomy is reconstructed by linking the partitions to each other. Finally, each partition is labeled by selecting a representative label.

entire partition. Thus, to handle these cases where a partition contains multiple disjoint subtrees of H, we first identify all maximal subtrees of H in P_i and then concatenate the concept labels of the roots of these subtrees to obtain the partition label for P_i. Fig. 7 shows an example.

Note that the time complexity of identifying the maximal subtrees in P_i is $O(r \times |P_i|)$, where $|P_i|$ is the number of nodes in $|P_i|$ and r is the number of disjoint subtrees in P_i.

4. Evaluation

In our experiments, we used two different data sets: a corpus of news articles from New York Times (NY Times) data set [6] (~64 K text entries with over ~100 K unique keywords) and a set of scientific abstracts from National Science Foundation[7] (~50 K article abstracts describing NSF awards for basic research, with over ~30 K unique keywords).

For each data set, we used a corresponding domain taxonomy extracted from the *DMOZ* categorization[8] by considering the most relevant terms, in the considered domains, extracted from the corpora. Specifically, we considered a taxonomy of science (with 72 nodes) which we used to index the NSF abstracts, and geographical taxonomy (181 nodes), against which we classified the articles from the NY Times. To increase the diversity of the input taxonomies we selected different subsets of these original hierarchies by randomly removing some of their nodes. These modifications permit us to test the approach with diverse domain hierarchies and also help avoid any bias that may be inherent in the original hierarchies; deletions of internal nodes of the structure can sensibly alter the hierarchy and provide different cases to consider.

Specifically, we created a total of 18 distinct taxonomies for each domain, obtained by removing anywhere between 10% to 60% (with 10+ increments, three different cases per percentage) of the concepts of the considered DMOZ taxonomy fragments. Moreover, for each of them, we considered different target taxonomy sizes. The results in these Sections are averages for all these taxonomies.

[6] http://archive.ics.uci.edu/ml/datasets/Bag+of+Words.
[7] http://kdd.ics.uci.edu/databases/nsfabs/nsfawards.html.
[8] Accessible at the link http://www.dmoz.org/.

Table 3
Strength of the structural constraints among the partitions shown in Fig. 6. These values reflect the number of edges that will be broken if two partitions will be not directly linked to each other.

strength of the structural constraints among the partitions				
	partition 1	partition 2	partition 3	partition 4
partition 1	–	0	0	0
partition 2	2	–	0	0
partition 3	3	2	–	0
partition 4	3	0	0	–

4.1. Effectiveness measures

As in [32], we are using the classification effectiveness as a measure of taxonomy quality. In our experiments, for each concept c_i in the considered hierarchy, we obtain a set of associated documents A_{c_i} that best match it through a classification process (Appendix C). In order to better understand the behavior of ANITA under different settings and to compare its performance to other algorithms on a concrete basis, we quantify the quality of the adapted taxonomies using the following three measures:

- *Domain coverage*: an important role of taxonomies in many applications is to help provide search and access to text documents. Thus, it is essential that they properly reflect the content of the corpus. Given a corpus of documents D and a taxonomy $H(C,E)$, the coverage of D by H is defined by the percentage of documents in D that can be associated to at least one concept in C using some classification process. Let A_{c_i} D be the set of documents associated to the concept $c_i \in C$. We define the domain coverage measure as

$$cover(H,D) = \frac{\left|\cup_{c_i \in C} A_{c_i}\right|}{|D|}.$$

The main idea of the proposed method is to minimize the loss in terms of domain coverage while we potentially reduce the size of the given hierarchy. Thus, the higher the domain coverage, the more effective the taxonomy in covering the knowledge expressed by the considered corpus.

- *Redundancy*: note that it would be trivial to increase the domain coverage simply by concatenating more and more labels. This would not result in a desirable taxonomy. Therefore, we also define a *redundancy* measure

$$redundancy(H,D) = \frac{|overlap(D,H)|}{\left|\cup_{c_i \in C} A_{c_i}\right|},$$

where $overlap(D,H)$ returns the set of documents in D associated to *at least* two concepts in H. This formula quantifies the discrimination power of the concepts in the resulting taxonomy, i.e., the degree of overlapping in the sets of documents associated to different concepts. The lower the redundancy, the higher the discrimination power, and thus the more effective the taxonomy in helping search and access text documents.

- *Label term-length*: finally, the *label term-length* (ltl) measure reports the average number of labels in the original taxonomy included in the labels of the adapted hierarchy. Given two hierarchies which provide similar domain coverage and redundancy, a more concise label is more desirable. Intuitively, a concept with a concatenated list of labels corresponds to a composite concept. Since longer compositions will induce potentially more ambiguity than shorter compositions, we can argue that the more concise the label length, the better is the label (of course as long as the domain coverage and redundancy stay intact). If we consider for example the adapted taxonomy fragment in Fig. 3(b), the composite concept "*political economics & microeconomics & macroeconomics & financial economics*", composed of 4 original labels, will be less precise than each individual concept in the list. Therefore, we roughly quantify this ambiguity by counting the labels that compose each concept name. Thus, given an initial taxonomy $H(C,E)$ and its adapted version $H'(C',E')$, $length(label_{c'_i},H,H')$ counts the number of original node labels in H that have been concatenated to form the label of c'_i in H'. Then, the label term-length is defined as

$$ltl(H,H') = \frac{\sum_{c'_i \in H'} length\left(label_{c'_i},H,H'\right)}{|C'|}.$$

In Sections 4.2 through 4.5, we present experiment results that rely on these three measures. In Section 4.6, we report the execution times. In Section 4.7, we then report user study results that quantify the impact of ANITA on the users' navigation experience.

4.2. Impact of the narrative orders

Tables 4 and 5 present the values of the effectiveness measures for the three proposed narrative orderings, with and without distance preserving sibling ordering. The values are averages of the performance results for five different target taxonomy sizes (from 10% to 50% of the original number of concepts, with 10+ increments).

From these two tables, we observe that sibling ordering results in slightly higher label term-length. This behavior is due to the fact that the ordering of siblings is likely to lead to longer sequences of similar siblings, which will be concatenated if the sequence does not contain the parent. It is important to note that this lengthening of the labels does not result in any increase in the redundancy of the resulting taxonomies. In all cases, the versions with sibling ordering have significantly smaller redundancies than the corresponding versions with the random ordering of siblings. The differences in terms of their domain coverages are negligible.

Considering the different traversal strategies, we observe that, for both data sets, parenthetical traversal provides lower redundancies and lower label term-lengths. Parenthetical traversal also provides the highest coverages, especially when distance preserving sibling ordering is used; intuitively, unlike pre- and post-order traversals where an internal node may be separated from a sibling significantly in the narrative, this method provides a higher coherence for the retrieved clusters by ensuring that siblings will always appear next to each other.

Thus, in the rest of the section, we only consider the parenthetical traversal with distance preserving sibling ordering.

4.3. Comparison wrt. the original taxonomy

In this section, we quantify how much difference in coverage and redundancy with respect to the original taxonomy occurs for varying target taxonomy sizes. Fig. 8 shows the ratios between the considered effectiveness measures on the adapted and the original taxonomies, referring to the NSF and NY Times data sets.

Fig. 8 shows that, for both data sets, the relative domain coverage is very close to 1.0 for adaptations with ≥30% of the nodes; this means that the adapted taxonomies can index the same amount of contents as the original taxonomies. As expected, the coverage drops when the size of the adapted taxonomy is pushed further down, even though the label length increases to compensate for this drop. Note that, despite this increase in the label lengths, ANITA is still able to lower the redundancy in the taxonomy, even when the compression rates are lowered down to 10% range. Finally, note that the similarities between the NSF and NY Times redundancy and label term-length curves on these charts highlight that the performance of ANITA in redundancy and label term-length is largely independent of the data set. One major difference among the two data sets is the coverage behavior: in the case of the NSF data set, the original taxonomy appears to have many unnecessary nodes (i.e., many nodes have very few documents associated in the considered corpus and therefore are less useful for navigation purposes within the taxonomy); thus, the relative domain coverage stays unaffected even when the target taxonomy has only 40% of the original nodes; after this point, there is a sharp drop implying that most documents are represented by only few nodes in the original taxonomy. In contrast, in the NY Times data set, the drop in

Table 4
Impact of different narrative orders.

Context: NSF corpus			
	cover.	redund.	Ltl
Pre-order (sibling ord.)	0.123	0.551	1.724
Parenth. (sibling ord.)	**0.128**	**0.510**	**1.681**
Post-order (sibling ord.)	**0.128**	0.530	1.702
Pre-order (no sibling ord.)	0.125	0.729	1.423
Parenth. (no sibling ord.)	**0.128**	**0.725**	**1.402**
Post-order (no sibling ord.)	**0.128**	0.736	1.463

Table 5
Impact of different narrative orders.

Context: NY Times corpus			
	cover.	redund.	Ltl
Pre-order (sibling ord.)	0.752	0.634	2.289
Parenth. (sibling ord.)	**0.759**	**0.573**	**2.204**
Post-order (sibling ord.)	0.755	0.612	2.277
Pre-order (no sibling ord.)	0.755	0.792	2.063
Parenth. (no sibling ord.)	**0.758**	**0.789**	1.966
Post-order (no sibling ord.)	0.756	0.792	**1.809**

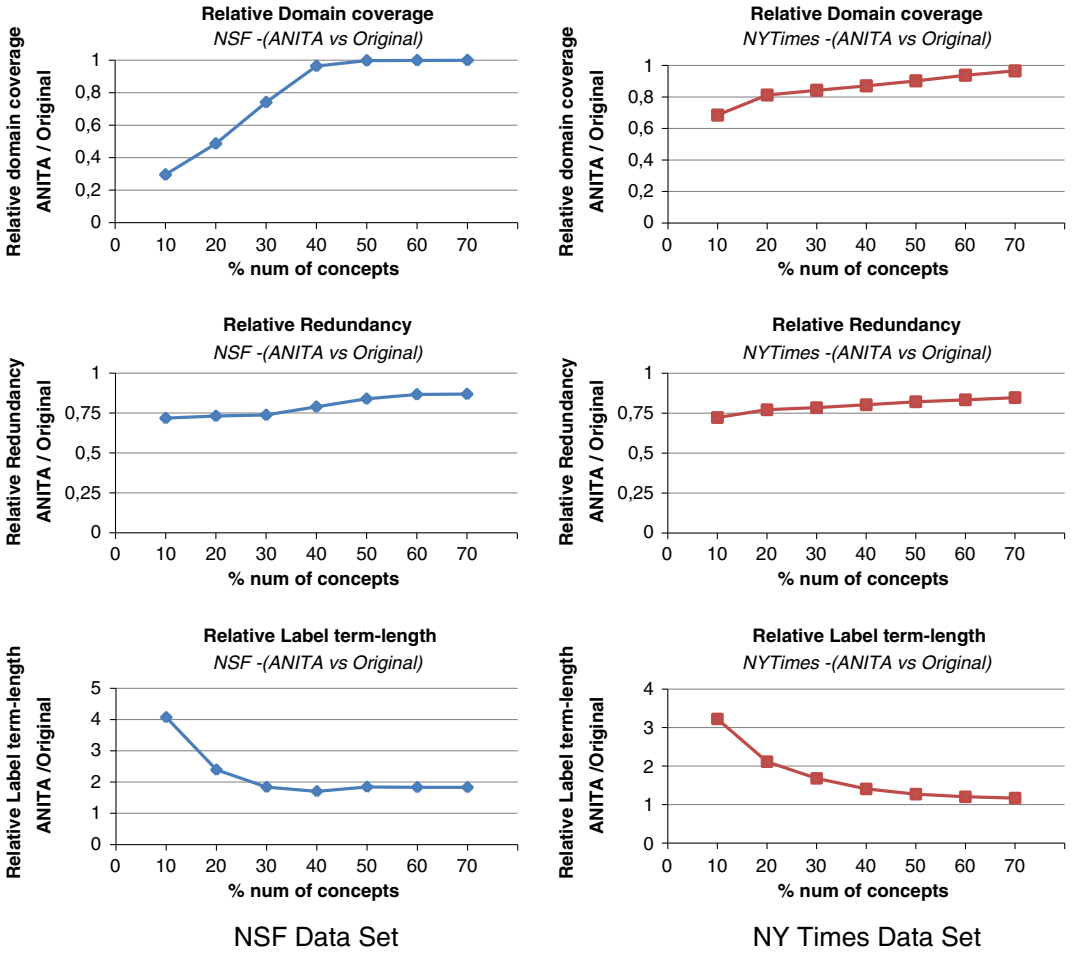


Fig. 8. Domain coverage, redundancy, and label term-length ratio $(\frac{ANITA}{Original})$ curves using NSF data set and NY Times data set.

coverage is slight, but relatively constant, indicating that (a) most of the taxonomy nodes are significantly represented in the data set, but (b) the documents have more geographical taxonomy nodes under which they can be classified. Note that a news article commenting about the war in Afghanistan may include the names of many nearby countries as well as countries that have sent military support, resulting in the article being associated to many geographic nodes. In NSF abstracts, topics are more focussed and therefore, while some concepts in the input taxonomy happen to have large numbers of associated documents, some other lack any.

4.4. Impact of document context in taxonomy adaptation

One of the key motivations of the proposed taxonomy adaptation approach is that a taxonomy that properly reflects the knowledge expressed by the considered corpus of contents can more precisely guide the user for exploration of those documents than another structure not properly informed about the corpus. In this section, we verify this hypothesis by comparing the effectiveness of taxonomies obtained by considering the entire data set, D, to the effectiveness of those adapted considering $D - D^*$ for a selected subset, $D^* \subset D$. We expect that having ignored D^* during the adaptation process will negatively impact the effectiveness of the resulting taxonomy in indexing D.

Thus, we consider the 18 original taxonomies and target taxonomy sizes between 10% and 70%. For each original taxonomy, H, we compare the full corpus informed adaptations, H' (obtained using the entire NSF data set, D), against $H'_{\neg bio}$ and $H'_{\neg astr}$ obtained as follows:

- Each $H'_{\neg bio}$ is a taxonomy *uninformed* about the documents concerning "*biology*"; i.e., it is adapted from the given H without considering the set, D_{bio}, of documents containing the term "*biology*".
- Similarly, each $H'_{\neg astr}$ is a taxonomy *uninformed* about the documents concerning "*astronomy*"; i.e., it is adapted from the given H without considering the set, D_{astr}, of documents containing the term "*astronomy*".

Fig. 9 compares the effectiveness of H' (referred to as "*informed*") to the effectiveness of $H'_{\neg bio}$ and $H'_{\neg astr}$ (collectively referred to as "*uninformed*").

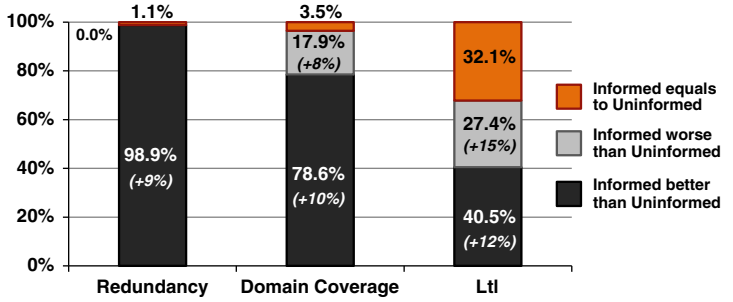


Fig. 9. Comparison in terms of Domain coverage, redundancy, label term-length between informed and uniformed taxonomies (NSF data sets). The values in parentheses are the average gains by the winning scheme.

The comparison in terms of domain coverage clearly demonstrates the benefits of using informed taxonomies instead of uninformed ones: in 78.6% of the cases the informed structure permits to obtain higher coverages (with an average gain of ~10%). Moreover, the benefit in terms of redundancy is also more evident, providing in 98.9% of the cases a lower redundancy in terms of associated documents (with an average gain of ~9%). Finally, in terms of label term-length, the informed clustering tends to provide lower label-length (with an average gain of ~12%). It is important to notice that, even in the small portion of cases in which the uninformed approach reports better performances, its relative gains are similar to those obtained when the informed approach reports better results.

4.5. ANITA vs. concept clustering methods

In Fig. 10 we compare the narrative-based partitioning approach against *k*-Means clustering, with *k* also being equal to the target taxonomy size requested from ANITA. In both cases, sentence-vector representation of the taxonomy nodes are used to support partitioning. Also, in both cases, once the partitions are obtained, the same taxonomy re-construction and labeling strategies (described in Section 3.3) are used to stitch the taxonomy back.

In these experiments, we considered target taxonomy size between 10% and 70% (with 10+ increments). The results in Fig. 10 simply report the percentages of cases in which one approach provides better performances than the other; as it is possible to notice, ANITA provides a clear gain in terms of lowering the amount of redundancy in the taxonomy (in 95.2% of the cases ANITA provides lower redundancy, with an average gain of ~14%). Moreover, ANITA also provides a gain in terms of domain coverage (in 61.9% of the cases, with an average gain of ~16%) and lower values in terms of label term-length (in 63.1% of the cases, with an average gain of ~22%), highlighting the global benefits of using the proposed adaptation approach. Again, it is important to notice that, even when *k*-Means reports better performances, its relative gains wrt. ANITA are similar to the relative gains obtained when ANITA reports better results.

Table 6 reports the results obtained by comparing ANITA against other clustering algorithms, such as EM, X-Means, and Hierarchical-EM (Hierarchical-EM method applies EM clustering strategy to each sibling group). Since these algorithms do not take target number of clusters as input, we first apply these algorithms and then use ANITA with the number of clusters returned by them. As these results show, ANITA provides better results in terms of all measures against these alternative clustering strategies; ANITA provides a clear gain in terms of lowering the amount of redundancy in the taxonomy in comparison to all the considered alternative approaches (up to 32% drop) as in terms of domain coverage (up to 14% increase) and provides lower values in terms of label term-length (a reduction up to 6%).

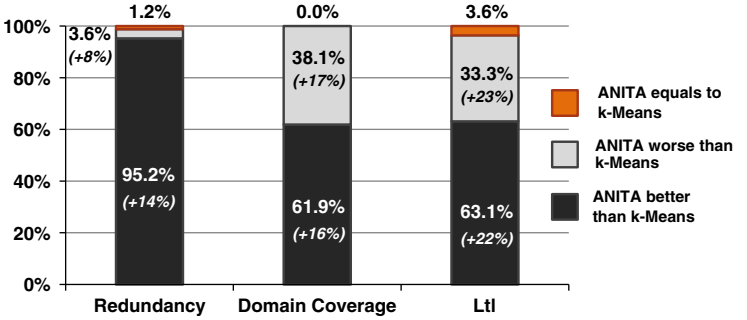


Fig. 10. Comparison in terms of Domain coverage, redundancy, label term-length between ANITA and *k*-Means (both data sets). The values in parentheses are the average gains by the winning scheme.

Table 6
ANITA vs. Hierarchical-EM ($\frac{ANITA}{H-EM}$), EM ($\frac{ANITA}{EM}$) and X-Means ($\frac{ANITA}{X-Means}$).

Context: NSF + NY-Times corpora			
	cover. ratio	redund. ratio	Ltl ratio
ANITA/H-EM	1.140	0.866	0.939
ANITA/EM	1.072	0.688	0.966
ANITA/X-Means	1.089	0.675	0.959

4.6. Execution time and complexity

For all the experiments we used an Intel Core 2CPU @2.16 GHz with 1 GHz Ram. The execution time is dominated by the initial text processing and concept analysis (Section 3.1.1), which for these experiments was around 60 seconds for the scientific input taxonomy of 72 nodes and 50 K NSF articles (and around 140 seconds for the geographical taxonomy of 181 nodes and 64 K NSF articles). The adaptation process itself takes less than 0.1 seconds.

Let n be the length of the narrative (i.e., the number of nodes of the original taxonomy). The adaptation process is dominated by the segmentation step described in Section 3.2, which

1. (a) fixes a starting point (dictated by the ordering process, Section 3.1.2);
2. (b) sequentially scans the narrative until the *drift* value is beyond the *coherence* threshold value;
3. moves the starting point to the first entry beyond the threshold. The segmentation process repeats these three steps until all the elements in the narrative have been considered.

Note that each element in the narrative is accessed only once. Thus independently from the threshold, the cost is O(n). Note that since the text processing is an off-line and one time process, the impact of the distilled taxonomies on the users' navigation times is a more critical factor than the execution time itself. We study this next through user studies.

4.7. User study

In order to analyze the benefits of using an ANITA adapted categorization for text data indexing purposes, we also conducted a user study (similarly to [41]) and evaluated the feedback of 16 users when exploring NSF text articles using different taxonomies. The users represent various range of ages, backgrounds, jobs and education level and they have intermediate web ability (they are not computer scientists or domain experts).

We presented to the users, three different taxonomies that indexed NSF documents: the original portion of DMOZ-extracted taxonomy, with 72 concepts (described in Section 3.3.2), its ANITA-based adaptation with 13 concepts (with k randomly set to 13) and the k-Means based adaptation (with same value of k). In order to avoid bias in the evaluation of the presented taxonomies, we presented the 3 taxonomies to the user in a random order.

4.7.1. Experiment 1: search time and interaction counts

Given one randomly selected concept label extracted from the original taxonomy (different for each participating user), we asked the users, for each presented taxonomy, to retrieve related documents by exploring the presented categorizations. Therefore, we analyze the time and the number of interactions (in terms of expansions/collapses of the presented nodes in the taxonomies) the user needs to reach satisfactory documents. As reported in Table 7, ANITA adapted taxonomy reports gains in terms of time (from an average of 23.5 s to an average of 9.7) and number of interactions (from 5.1 to 2.3) by reducing the number of nodes the user has to navigate through. On the other hand it is important to note that, even if k-Means adapted taxonomy presents the same number of nodes as ANITA, it is not able to guide the user as well as ANITA adapted taxonomies do; the user needs more time to find relevant documents (an average of 11.0 seconds) and also more interactions to retrieve the appropriate contents (an average of 2.9 operations). Therefore, in case the user needs an adaptation of the taxonomy, we can state that ANITA is not only able to reduce the cardinality of the selected taxonomy, but also organizes the concepts in such a way to facilitate the retrieval operations.

Table 7
User study: average time and average number of interactions (clicks on the structure for expanding or collapsing nodes) per taxonomy, when the users explore the structure to retrieve documents related to a randomly selected concept.

Context: NSF corpus		
	avg. time (sec)	avg. num. of interactions
Original (72 concepts)	23.5	5.1
ANITA (13 concepts)	9.7	2.3
k-Means (13 concepts)	11.0	2.9

4.7.2. Experiment 2: classification effectiveness

In this experiment, we aim to measure the effectiveness of different (original and adapted taxonomies) in supporting classification. For this experiment, we presented each user a randomly selected article (different for each user) from the NSF corpus of documents. We then presented the user different taxonomies and asked to select those nodes (if any) that would best represent the article in each taxonomy. Then, we used an automatic classification system (Appendix C) to associate the given article to concepts in each taxonomy. Let d_i be an article and t_j be a taxonomy and the set $U^{t_j}(d_i)$ be the set of concepts selected by the user for d_i relying on t_j. Let also $A^{t_j}(d_i)$ be the set of concepts selected by the automated system for d_i, relying on the information represented in t_j. We then define the *precision-based* and *recall-based* effectiveness of t_j as

$$eff_{precision}\left(t_j\right) = \frac{U^{t_j}(d_i) \cap A^{t_j}(d_i)}{A^{t_j}(d_i)} \quad eff_{recall}\left(t_j\right) = \frac{U^{t_j}(d_i) \cap A^{t_j}(d_i)}{U^{t_j}(d_i)}.$$

respectively.

The experiment results indicate that, for the original taxonomy, 67.7% of the user selected concepts were shared by the system (i.e., recall is 67.7%), while the precision of the process had an average value of 60.7%.

The ANITA-based adapted taxonomy, on the other hand, was more effective both in recall and precision: ANITA-based adapted taxonomies provided an average recall value of 68.7% (indicating that the quality of the taxonomy is as good as the original one despite containing much smaller number of concepts) and, more importantly, increased the average precision value significantly to 76.8%.

In these experiments, the *k*-Means based adapted taxonomy did not prove to be effective: its recall value was only 37.4% and the average precision of the classification process was 52.5%, highlighting the fact that a naive re-structuring process (such as *k*-Means) can cause a significant increase in terms of confusion and disorganization.

4.7.3. Experiment 3: subjective questionnaire measures

After the study, each user also completed a brief questionnaire which included two questions ("Is the taxonomy easy to use?" and "Is the taxonomy sufficiently detailed?"); the users could quantify the responses using a 5-point scale ratings.

As shown in Table 8, the users reported that the ANITA adapted taxonomy was as "easy to use" as the original one (both 4.1) while the *k*-Means adapted taxonomy was significantly harder to use (3.3). Moreover, even if the number of presented nodes was dropped almost 80%, the users commented that, in terms of providing "sufficient details" (i.e., the number of alternatives), ANITA adapted taxonomy provides a good range of details, close to the original one (3.6 vs 3.8). We can summarize these results as follows: as initially supposed, the original taxonomies, developed by domain experts for broad coverage of documents, provide unnecessary details that can be removed without causing a loss in terms of contextual knowledge. On the other hand, a general adaptation method such as *k*-Means, could introduce confusion and disorientation: the *k*-Means adapted taxonomy reduces the "sufficiency" (only 2.6) and results in taxonomies that the users find harder to use (3.3 in terms of "easy to use").

4.7.4. Statistical significance

In order to evaluate the statistical significance of the presented results, we performed the *t*-test, which measures the difference between the means of two or more groups and is generally used to verify that the means of two groups are statistically different from each other. The results are shown in Table 9.

Table 8
Subjective questions in the user study: for each question, each user has quantified her opinion by a 5-point scale rating.

Context: NSF corpus		
	easy to use	sufficiently detailed
Original (72 concepts)	4.1	3.8
ANITA (13 concepts)	4.1	3.6
k-Means (13 concepts)	3.3	2.6

Table 9
P-values for the paired *t*-test for means; comparing user study results for ANITA-based taxonomies against original taxonomies and taxonomies obtained using a *k*-Means based strategy. Results show that ANITA-based taxonomies are statistically significantly better than the original taxonomies in terms of the number of interactions necessary to complete a task and the time taken; otherwise, there is no statistically significant difference in terms of classification accuracy, number of alternatives, ease of use, or user preference between ANITA-based taxonomies and the original ones.

	ANITA vs. original	ANITA vs. *k*-Means
Reduction in num. of interactions	**0.003**	**0.046**
Reduction in time taken for task	**0.00004**	0.065
Classification accuracy	0.89	**0.00003**
Number of alternatives	0.33	**0.0009**
Ease of use	0.718	**0.003**
User preference	0.27	**0.029**

In terms of the number of interactions with the taxonomy to complete the given task (Subsection 4.7.1), ANITA taxonomies are statistically significantly better than the original taxonomy as well as taxonomies created using *k*-Means ($p = 0.003$ and 0.046, respectively). Here *statistically significant* means that the *p*-value is <0.05; i.e., that we are certain with confidence >95% that the difference is not due to chance. In terms of average navigation time (Subsection 4.7.1), the hierarchies created by ANITA are also statistically significantly better than the original hierarchies ($p = 0.00004$). In addition, we have 93.5% certainty that the improvements seen when using ANITA instead of *k*-Means based hierarchies are not due to chance ($p = 0.065$).

As we would expect, in terms of classification accuracy (Subsection 4.7.2), there is no difference between ANITA taxonomies and the original taxonomies($p = 0.896$). However, ANITA taxonomies are statistically significantly better than *k*-Means based taxonomies in terms of classification accuracy ($p = 0.00003$). Also in terms of the provided alternatives (Subsection 4.7.3), there is no difference between ANITA taxonomies and original taxonomies ($p = 0.33$). However, ANITA taxonomies provide, again, better results than *k*-Means based taxonomies ($p = 0.0009$).

In terms of ease of use (Subsection 4.7.3), we did not observe any difference between ANITA taxonomies and original taxonomies ($p = 0.718$). However, ANITA taxonomies are statistically significantly better than *k*-Means taxonomies ($p = 0.003$). Also in terms of users' overall liking of the presented taxonomies (Subsection 4.7.3), there is no statistically significant difference between ANITA taxonomies and the original ones ($p = 0.27$). However, ANITA taxonomies are statistically significantly better than *k*-Means based taxonomies ($p = 0.029$).

In summary, results show that ANITA-based taxonomies are statistically significantly better than the original taxonomies in terms of the number of interactions necessary to complete a task and the time taken; otherwise, there is no difference in terms of classification accuracy, number of alternatives, ease of use, or user preference between ANITA-based taxonomies and the original ones. With respect to the *k*-Means based taxonomies, however, ANITA taxonomies are statistically significantly better in almost all objective and subjective measures. Even in the single case (time to task completion) where the *p*-value is above 0.05, we have 93.5% confidence in that the reductions provided by ANITA taxonomies in terms of the user's navigation time with respect to *k*-Means based taxonomies are not due to chance.

5. Conclusions

In this paper, we introduced *A Narrative Interpretation of Taxonomies for their Adaptation* (ANITA) for re-structuring existing taxonomies to varying application contexts. The experimental results showed that the proposed technique can provide benefits in terms of reducing the redundancies in the taxonomies if they need to be adapted (preserving also the domain coverages), and the user studies also validated the approach from a user point of view.

It is important to notice that many uses of the proposed ANITA adaptation method are possible; for instance, ANITA can be used for organizing documents on a per-query basis (i.e., considering only documents that are relevant for a given query), thus improving the user search experience through large text collections. Our future research will include adaptation of more general ontologies (including directed acyclic graphs) to enable adaptation of many commonly used ontologies, like DMOZ and Wikipedia, to the users' foci of interest and to their navigation devices. We will also investigate the impact of deeper natural language processing [42] of the input text collections to improve the understanding of the keywords considered in the process.

Appendix A. Sentence vector construction

In this Appendix, we describe how to create the sentence-vectors combining information coming from a structural analysis of the relationships formalized in H with the analysis of the most frequent keywords appearing in the corpus of documents D.

Appendix A.1. Taxonomy vectorization

In order to support the creation of sentence-vectors, we map the concepts in the given domain taxonomy, $H(C,E)$, onto a concept-vector space. More specifically, given a taxonomy, $H(C,E)$, with $n = |C|$ concepts, we represent each concept node as a vector $c \rightarrow \upsilon$ with n dimensions such that each vector represents the semantical relationship of the corresponding concept node with the rest of the nodes in the taxonomy. For this analysis step, we rely on the CP/CV mapping process proposed by [43]. Given a taxonomy, CP/CV assigns a *concept-vector*[9] to each concept node in the taxonomy, such that the vector encodes the *structural* relationship between this node and all the other nodes in the hierarchy. In order to create these concept-vectors, each concept-vector of the nodes is simply initialized by setting to 1 the weight corresponding to itself; i.e., considering the node c_i in the given hierarchy, the initial concept-vector of this node is

$$\vec{c\upsilon}_{c_i} = 0, 0, \ldots, 1, \ldots, 0 \tag{A.1}$$

where the only non-zero weight is associated with the i-th dimension related to the node c_i. The total number of dimensions is equal to the number of the nodes in $H(C,E)$.

Then, the process repeatedly enriches the concept-vectors of the nodes by enabling neighboring nodes to exchange concept weights. The propagation of the weights works by adding to each concept-vector the weights of the neighbor ones (parent and

[9] In the rest of the Appendix, we use the terms "*concept-vector*", coined in [43], and "*sentence-vector*" we use in this paper interchangeably.

children), multiplied by a *propagation degree* that sets how much information has to migrate from one node to the neighbors. The propagation degree is computed in a way that reflects the local structure of the taxonomy [43].

This process is iterated until all nodes are informed of all the others. The necessary steps required by the propagation process strictly depend on the depth of the considered hierarchy; for example, in a taxonomy of depth d, it is necessary to perform $2 \cdot d$ iterations to inform all the concept about the entire structure [43].

Appendix A.2. Text document vectorization

In this step, given a data corpus D of text documents, we analyze and extract a representative document-vector for each of them. Thus, each of the $m = |D|$ documents is represented with a *document-vector* in which each component represents a keyword. As usual, the keyword extraction includes a preliminary phase of stop-word elimination and stemming. The weight of each keyword is computed using augmented normalized term frequency [44].

In short, given a corpus document d_i, we define the related document-vector as

$$\vec{d}_i = \left\{ w_{i,1}, w_{i,2}, \ldots, w_{i,v} \right\} \tag{A.2}$$

where v is the size of the considered vocabulary, and $w_{i,j}$ is the normalized term frequency of the jth vocabulary term in the ith document, calculated as

$$w_{i,j} = 0.5 + 0.5 \cdot \frac{tf_{i,j}}{tf_i^{max}}$$

where tf_i^{max} returns the highest term frequency value of the ith document.

Appendix A.3. Analysis of concepts describing a given document

For each document in the corpus, the concepts that best describe the document are those concepts whose similarities (as defined in Appendix A.5) with the document are above an adaptively computed critical point (Fig. A.11). Intuitively, to preserve only the documents with very high similarities, we associate a given document to a hierarchy concept only if their similarity is higher than the average similarity of the documents that best match that concept.

More in detail, the steps of this discovery process are as follows [34]: For each document $d_j \in D$ we

1. consider the document-vector $\vec{dv}_j$
2. compute its similarity wrt. all the concept-vectors describing the given taxonomy.

$$sim\left(\vec{cv}_i, \vec{dv}_j\right) = \Sigma_{k=1}^{u} cv_i[k] \times dv_j[k]. \tag{A.3}$$

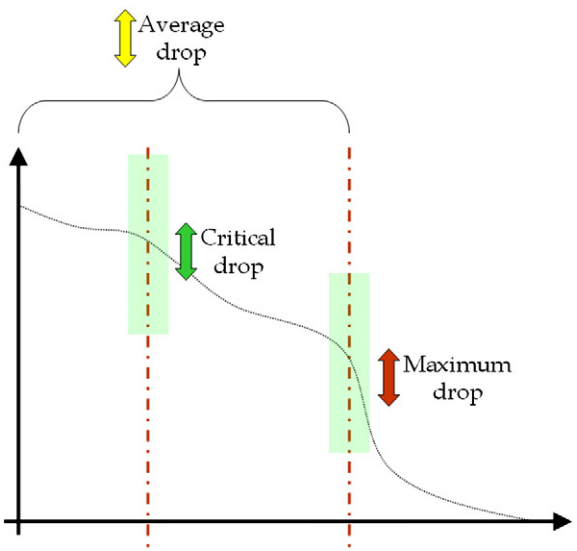


Fig. A.11. The critical-point cut-off [34,45]: the maximum drop is the highest variation in the ordered list of weights (red mark). The average drop (between consecutive entities) is the average difference between those items that are ranked before the identified maximum drop point (yellow mark). The first drop which is higher than the computed average drop is called the critical drop (green mark).

3. sort the concept-vectors in decreasing order of similarity wrt. $\vec{d}\vec{v}_j$;
4. choose the cut-off point to identify the concepts which can be considered *sufficiently similar*; the method adaptively computes this cut-off as follows: it
   (a) first ranks the concepts in descending order of match to $\vec{d}\vec{v}_j$, as previously calculated;
   (b) computes the *maximum drop* in match and identifies the corresponding drop point;
   (c) computes the *average drop* (between consecutive entities) for all those nodes that are ranked before the identified maximum drop point;
   (d) the first drop which is higher than the computed average drop is called the *critical drop*. The concepts ranked better than the point of critical drop are returned as candidate matches.

At the end of this phase, each document in D has a non-empty set of concepts associated to it.

Appendix A.4. Finding keywords that relate strongly to a given concept

The next step toward the sentence-vectors construction process is to discover the concept-keyword mappings using these associations identified in the previous step. In other words, in this phase, we find those keywords that relate strongly to the concepts in the taxonomy.

Let $\vec{cv}_{c_i}$ denote the concept-vector corresponding to concept c_i. We denote the set of documents described by the concept c_i as $D_{\vec{cv}_{c_i}}$. Notice that, in general, the sets of associated documents for different concepts are not disjoint, since the same document can be assigned to multiple (similar) concept-vectors. Note also that, at the end of the process, some of the concept nodes of the taxonomy may not be associated as a descriptive concept to any of the documents in the database. For such concepts, the corresponding sets, D_{desc}, of associated documents are empty.

At this step, given a concept c_i and the set, $D_{\vec{cv}_{c_i}}$, of associated documents, we search for the most contextually informative keywords corresponding to this concept.

More specifically, we compute the degree of matching between the given concept and a keyword which occurs in the associated documents by treating

- the set of documents in $D_{\vec{cv}_{c_i}}$ which contain the keyword as positive evidence of relationship between the concept and the keyword within the given context, and
- the documents in the database containing the keyword but not associated to the concept as negative evidence against the relationship.

Thus, considering a concept c_i and its associated document, we aim to search for the most contextual informative keywords. For this, we treat each document in the related association as a bag of words (containing the keywords extracted from the original texts). Thus, as discussed in [46], we compute the degree of matching between the keyword and the concept by treating each document contained in the association as a positive relevance feedback and each document containing the keyword but not in the concept association as a negative relevance feedback against the relationship. In other words, this phase aims to find those keywords that better characterize the concept in the data corpus. Therefore, given a concept-vector and a corresponding association, this process aims to identify those keywords that are significant for the characterization of the concept in the given context.

Recognizing this, given a concept c_i and a corresponding set of associated documents, $D_{\vec{cv}_{c_i}}$, we identify the weight, $u_{i,j}$, of the keyword k_j relying on a probabilistic feedback mechanism [47]. Intuitively, given a concept in the taxonomy, the corresponding sentence vector is considered as a "*query*" and the document associated to the concept is treated as a "*set of results to this query*". Then, given a keyword from the corpus, we treat each document in the associated set containing the keyword as a positive "*feedback*" for that keyword. On the other hand, to prevent those keywords that are frequent in the corpus, but not related to the given concept, from having high scores, we treat each document in the remainder of the corpus containing the keyword as a negative "*feedback*":

$$u_{i,j} = log\left(\frac{\frac{r_{i,j}}{(R_i - r_{i,j})}}{\frac{(n_j - r_{i,j})}{N - n_j - R_i + r_{i,j}}}\right) \times \left|\frac{r_{i,j}}{R_i} - \frac{n_j - r_{i,j}}{N - R_i}\right|, \tag{A.4}$$

where $r_{i,j}$ is the number of documents in $D_{\vec{cv}_{c_i}}$ containing the keyword k_j, n_j is the number of documents in the corpus containing the keyword k_j, R_i is the number of documents in $D_{\vec{cv}_{c_i}}$; and N is the number of documents in the corpus.

It is important to notice that the first term increases as the number of the associated documents containing the keyword k_j increases, while the second term decreases when the number of the non-associated documents containing the keyword k_j increases. Therefore, keywords that are highly common in a specific association and not much present in others will get higher weights.

For each concept, we consider all keywords contained in at least one document. We apply the adaptive cut-off (as explained in Appendix A.3) to this set in order to select those keywords with the highest weights. Given concept c_i, the selected keywords and their weights are collected in a so-called *extension-vector*, $\vec{l}\vec{v}_{c_i}$.

Appendix A.5. Merging concept- and extension-vectors

At this point, for each concept c_i, we have two vectors: (a) the original concept-vector, $\vec{c}\vec{v}_{c_i}$, representing the concept–concept relationships in the corresponding taxonomy and (b) the extension-vector, $\vec{l}\vec{v}_{c_i}$, consisting of keywords that are significant in the current context defined by the corpus. In order to combine the concept and the collection extension-vectors, into a single sentence-vector,

$$\vec{s}\vec{v}_{c_i} = \alpha_{c_i} \cdot \vec{c}\vec{v}_{c_i} + \beta_{c_i} \cdot \vec{l}\vec{v}_{c_i}, \tag{A.5}$$

we need to first establish the relative impacts (i.e. α_{c_i} and β_{c_i}) of the taxonomical knowledge versus real-world background knowledge.

As defined previously in Appendix A.4, let $D_{\vec{cv}_{c_i}}$ be the set of documents for which the concept c_i is a good descriptive concept. Also, given concept, c_i, let

- $A_{\vec{cv}_{c_i}}$ be the set of documents resulting from querying the database using the concept-vector, $\vec{c}\vec{v}_{c_i}$ (Appendix C); and
- $A_{\vec{lv}_{c_i}}$ be the set of documents obtained by querying the database using the extension-vector, $\vec{l}\vec{v}_{c_i}$ (Appendix C).

We quantify the relative impacts, α_{c_i} and β_{c_i}, of the concept and extension-vectors, $\vec{c}\vec{v}_{c_i}$ and $\vec{l}\vec{v}_{c_i}$, by comparing how well $A_{\vec{cv}_{c_i}}$ and $A_{\vec{lv}_{c_i}}$ approximate $D_{\vec{cv}_{c_i}}$. In other words, if

- $C_{c_i} = D_{\vec{cv}_{c_i}} \cap A_{\vec{cv}_{c_i}}$ and
- $L_{c_i} = D_{\vec{cv}_{c_i}} \cap A_{\vec{lv}_{c_i}}$,

then we expect that

$$\frac{||\alpha_{c_i} \cdot \vec{c}\vec{v}_{c_i}||}{||\beta_{c_i} \cdot \vec{l}\vec{v}_{c_i}||} = \frac{|C_{c_i}|}{|L_{c_i}|}. \tag{A.6}$$

If the concept and extension-vectors are normalized to 1, then we can rewrite this as

$$\frac{\alpha_{c_i}}{\beta_{c_i}} = \frac{|C_{c_i}|}{|L_{c_i}|}. \tag{A.7}$$

Also, if we further constrain that the combined-vector $c{\rightarrow}l\vec{v}_{c_i}$ is also normalized to 1,

$$||\alpha_{c_i} \cdot \vec{c}\vec{v}_{c_i} + \beta_{c_i} \cdot \vec{l}\vec{v}_{c_i}|| = 1, \tag{A.8}$$

then, solving these equations for α_{c_i} and β_{c_i}, we obtain:

$$\alpha_{c_i} = \frac{|C_{c_i}|}{|C_{c_i}| + |L_{c_i}|} \quad \text{and} \quad \beta_{c_i} = \frac{|L_{c_i}|}{|C_{c_i}| + |L_{c_i}|}. \tag{A.9}$$

Thus, given a concept, c_i, we can compute the corresponding sentence-vector as

$$\vec{s}\vec{v}_{c_i} = \frac{|C_{c_i}|}{|C_{c_i}| + |L_{c_i}|} \cdot \vec{c}\vec{v}_{c_i} + \frac{|L_{c_i}|}{|C_{c_i}| + |L_{c_i}|} \cdot \vec{l}\vec{v}_{c_i}. \tag{A.10}$$

Appendix B. Measuring semantic similarities

In our work, we need to measure similarity of a pair of concepts or a concept to a document.

The sentence-vectors associated to concepts provide a convenient representation for this purpose.

Similarity of two concepts

[43] showed that cosine similarity (measuring the angles between the vectors) among concept-vectors leads to highly precise similarity measurement across concepts within the taxonomy; comparisons against other approaches on available human-generated benchmark data [48,49] showed that this provides better concept similarity measurements (in terms of the correlation of the resulting concept similarity judgments to human common sense). Thus, in this paper, without loss of generality, we use this approach to measure the semantic similarity of a pair of concepts using the corresponding sentence-vectors.

Similarity of a concept and a document

We also measure the similarity between a concept and a document similarly by comparing the concept's sentence-vector to the document-vector. However, before computing the cosine similarity of the two vectors, we first need to unify the vector space of the concept and the vector space of the document. The unification of the spaces consists in unioning dimensions in the given ones, and representing every vector in the new extended space by setting to 0 the values corresponding to those dimensions that were not appearing in the original vector space, while keeping all the other components unchanged. (b) Once the process is completed, both vectors are mapped into the same vector space and similarity can be computed by comparing these vectors [50].

Appendix C. Sentence-vector based classification of documents

For classification of documents under concepts in a given taxonomy, we leverage similarities between the corresponding sentence- and document vectors (Appendix A.5).

The classification process is performed by calculating the cosine similarity between each document-vector (containing terms frequency information) and the concept-vectors, in the same terms space. Finally, for each concept, we pick those text documents with similarity higher than a threshold value (calculated adaptively as described in Appendix A.3).

References

[1] L. Tang, H. Liu, J. Zhang, N. Agarwal, J.J. Salerno, Topic taxonomy adaptation for group profiling, ACM Transactions on Knowledge Discovery from Data 1 (2008) 1:1–1:28.
[2] M. Cataldi, K.S. Candan, M.L. Sapino, ANITA: a narrative interpretation of taxonomies for their adaptation to text collections, Proceedings of the 19th ACM International Conference on Information and Knowledge Management, ACM Conference on Information and Knowledge Management (CIKM), ACM, New York, NY, USA, 2010, pp. 1781–1784.
[3] Q. Li, K.S. Candan, Q. Yan, Extracting relevant snippets for web navigation, Proceedings of the 23rd National Conference on Artificial Intelligence, Volume 2, AAAI Press, 2008, pp. 1195–1200.
[4] I. Varlamis, S. Stamou, Semantically driven snippet selection for supporting focused web searches, Data & Knowledge Engineering 68 (2009) 261–277.
[5] D.J. Lawrie, W.B. Croft, Generating hierarchical summaries for web searches, ACM SIGIR Conference on Research and Development in Information Retrieval, ACM Press, 2003, pp. 457–458.
[6] D. Lawrie, W.B. Croft, A. Rosenberg, Finding topic words for hierarchical summarization, Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'01, ACM, New York, NY, USA, 2001, pp. 349–357.
[7] K. Kummamuru, R. Lotlikar, S. Roy, K. Singal, R. Krishnapuram, A hierarchical monothetic document clustering algorithm for summarization and browsing search results, International World Wide Web Conference, ACM, 2004, pp. 658–665.
[8] Y. Li, X. Zhou, P. Bruza, Y. Xu, R.Y. Lau, A two-stage text mining model for information filtering, Proceeding of the 17th ACM Conference on Information and Knowledge Management, ACM Conference on Information and Knowledge Management (CIKM), ACM, New York, NY, USA, 2008, pp. 1023–1032.
[9] P. Buitelaar, P. Cimiano, B. Magnini (Eds.), Ontology Learning from Text: Methods, Evaluation and Applications, volume 123 of *Frontiers in Artificial Intelligence*, IOS Press, 2005.
[10] M. Sanderson, B. Croft, Deriving concept hierarchies from text, ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 1999, pp. 206–213.
[11] P. Cimiano, A. Hotho, S. Staab, Learning concept hierarchies from text corpora using formal concept analysis, Journal of Artificial Intelligence Research (JAIR) 24 (2005) 305–339.
[12] P. Cimiano, A. Pivk, L. Schmidt-Thieme, S. Staab, Ontology learning from text: methods, evaluation and applications, Computational Linguistics 32 (2006) 569–572.
[13] C. Brewster, F. Ciravegna, Y. Wilks, User-centred ontology learning for knowledge management, 7th Intíl Conf. Applications of Natural Language to Information Systems, Springer-Verlag, 2002, pp. 203–207.
[14] C.C. Aggarwal, S.C. Gates, P.S. Yu, On the merits of building categorization systems by supervised clustering, KDD'99, ACM Press, 1999, pp. 352–356.
[15] T. Li, S. Zhu, M. Ogihara, Hierarchical document classification using automatically generated hierarchy, Journal of Intelligent Information Systems (JIIS) 29 (2007) 211–230.
[16] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.
[17] K. Punera, S. Rajan, J. Ghosh, Automatically learning document taxonomies for hierarchical classification, International World Wide Web Conference, ACM, 2005, pp. 1010–1011.
[18] T. Hofmann, The cluster-abstraction model: unsupervised learning of topic hierarchies from text data, IJCAI'99, Morgan Kaufmann Publishers Inc, 1999, pp. 682–687.
[19] E. Segal, D. Koller, D. Ormoneit, Probabilistic abstraction hierarchies, Advances in Neural Information Processing Systems, MIT Press, 2001, pp. 913–920.
[20] S.P. Ponzetto, R. Navigli, Large-scale taxonomy mapping for restructuring and integrating Wikipedia, IJCAI'09, Morgan Kaufmann Publishers Inc, 2009, pp. 2083–2088.
[21] S.P. Ponzetto, M. Strube, Deriving a large scale taxonomy from Wikipedia, AAAI'07, AAAI Press, 2007, pp. 1440–1445.
[22] P. Schmitz, Inducing ontology from flickr tags, in: Proc. of the Collaborative Web Tagging Workshop (WWW'06), 2006 pp. 3–6.
[23] J. Conesa, V.C. Storey, V. Sugumaran, Usability of upper level ontologies: the case of ResearchCyc, Data & Knowledge Engineering 69 (2010) 343–356.
[24] L. Tang, H. Liu, J. Zhang, N. Agarwal, J.J. Salerno, Topic taxonomy adaptation for group profiling, ACM Transactions on Knowledge Discovery from Data 1 (2008) 1–28.
[25] A. Singh, K. Nakata, Hierarchical classification of web search results using personalized ontologies, Proceedings of the 3rd International Conference on Universal Access in Human–Computer Interaction, 2005.
[26] W. Pratt, M.A. Hearst, L.M. Fagan, A knowledge-based approach to organizing retrieved documents, Proceedings of IAAI'99, American Association for Artificial Intelligence, 1999, pp. 80–85.
[27] K.-P. Yee, K. Swearingen, K. Li, M. Hearst, Faceted metadata for image search and browsing, Proceedings of CHI'03, ACM, 2003, pp. 401–408.
[28] O. Ben-Yitzhak, N. Golbandi, N. Har'El, R. Lempel, A. Neumann, S. Ofek Koifman, D. Sheinwald, E. Shekita, B. Sznajder, S. Yogev, Beyond basic faceted search, Proceedings of WSDM'08, ACM, 2008, pp. 33–44.
[29] W. Dakka, P.G. Ipeirotis, K.R. Wood, Automatic construction of multifaceted browsing interfaces, ACM Conference on Information and Knowledge Management (CIKM), ACM, 2005, pp. 768–775.
[30] L. Rosenfeld, P. Morville, Information Architecture for the World Wide Web, 2nd edition O'Reilly Associates, Inc., Sebastopol, CA, USA, 2002.
[31] O. Zamir, O. Etzioni, Web document clustering: a feasibility demonstration, ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 1998, pp. 46–54.
[32] Y. Zhao, G. Karypis, Evaluation of hierarchical clustering algorithms for document datasets, Data Mining and Knowledge Discovery, ACM Press, 2002, pp. 515–524.

[33] M. Sanderson, Word sense disambiguation and information retrieval, ACM SIGIR Conference on Research and Development in Information Retrieval, Springer-Verlag New York, Inc., 1994, pp. 142–151.
[34] M. Cataldi, C. Schifanella, K.S. Candan, M.L. Sapino, L. Di Caro, Cosena: a context-based search and navigation system, Proceedings of the International Conference on Management of Emergent Digital EcoSystems, MEDES'09, ACM, New York, NY, USA, 2009, pp. 218–225.
[35] C. Muller, I. Gurevych, M. Muhlhauser, Integrating semantic knowledge into text similarity and information retrieval, ICSC'07, IEEE Computer Society, 2007, pp. 257–264.
[36] S. Patwardhan, S. Banerjee, T. Pedersen, Umnd1: unsupervised word sense disambiguation using contextual semantic relatedness, SemEval'07, Association for Computational Linguistics, 2007, pp. 390–393.
[37] W.S. Torgerson, Theory and Methods of Scaling, R.E. Krieger Pub. Co, 1958.
[38] M.A. Hearst, Text tiling: segmenting text into multi-paragraph subtopic passages, Computational Linguistics 23 (1997) 33–64.
[39] M.A. Hearst, TextTiling: a quantitative approach to discourse, Technical Report, Computer Science Department, Berkeley, CA, USA, 1993.
[40] S. Kaufmann, Cohesion and collocation: using context vectors in text segmentation, Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics, ACL, 1999, pp. 591–595.
[41] H. Chen, S. Dumais, Bringing Order to the Web: Automatically Categorizing Search Results, CHI'00, ACM, 2000, pp. 145–152.
[42] A. Basden, H.K. Klein, New research directions for data and knowledge engineering: a philosophy of language approach, Data & Knowledge Engineering 67 (2008) 260–285.
[43] J.W.. Kim, K.S. Candan, Cp/cv: concept similarity mining without frequency information from domain describing taxonomies, in: ACM Conference on Information and Knowledge Management (CIKM), 2006, pp. 483–492.
[44] G. Salton, C. Buckley, Term-weighting approaches in automatic text retrieval, in: Information Processing and Management, 1988, pp. 513–523.
[45] M. Cataldi, L. Di Caro, C. Schifanella, Emerging topic detection on twitter based on temporal and social terms evaluation, Proceedings of the Tenth International Workshop on Multimedia Data Mining, ACM, New York, NY, USA, 2010, pp. 1–10.
[46] J.J. Rocchio, Relevance feedback in information retrieval, in: G. Salton (Ed.), The Smart Retrieval System — Experiments in Automatic Document Processing, Prentice-Hall, Englewood Cliffs, NJ, 1971, pp. 313–323.
[47] I. Ruthven, M. Lalmas, A survey on the use of relevance feedback for information access systems, Knowledge Engineering Review 18 (2003) 95–145.
[48] G.A. Miller, W.G. Charles, Contextual correlates of semantic similarity, Language & Cognitive Processes 6 (1991) 1–28.
[49] P. Resnik, Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language, Journal of Artificial Intelligence Research 11 (1999) 95–130.
[50] J.W. Kim, K.S. Candan, Leveraging structural knowledge for hierarchically-informed keyword weight propagation in the web, WebKDD'06, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 72–91.

Mario Cataldi obtained his Ph.D. in Computer Science in 2010 and he is currently a temporary researcher at the University of Torino, Italy. His research interest includes text mining, social network analysis, data representation and summarization.

K. Selcuk Candan is a Professor of Computer Science and Engineering at the School of Computing, Informatics, and Decision Science Engineering at the Arizona State University and is leading the EmitLab research group. He joined the department in August 1997, after receiving his Ph.D. from the Computer Science Department at the University of Maryland at College Park. Prof. Candan's primary research interest is in the area of management of non-traditional, heterogeneous, and imprecise (such as multimedia, web, and scientific) data. His various research projects in this domain are funded by diverse sources, including the National Science Foundation, Department of Defense, Mellon Foundation, and DES/RSA (Rehabilitation Services Administration). He has published over 140 articles and many book chapters. He has also authored 9 patents. Recently, he coauthored a book titled "Data Management for Multimedia Retrieval" for the Cambridge University Press and co-edited "New Frontiers in Information and Software as Services: Service and Application Design Challenges in the Cloud" for Springer.

Maria Luisa Sapino got her Ph.D. degree in Computer Science at the University of Torino, where she's currently Full Professor. Her initial contributions to computer science were in the area of logic programming and artificial intelligence, specifically in the semantics of negation in logic programming, and in the abductive extensions of logic programs. Since mid-90s she has been applying these techniques to the challenges associated with database access control, and with heterogeneous and multimedia data management. In particular, she developed novel techniques and algorithms for similarity based information retrieval, content based image retrieval, and web accessibility for users who are visually impaired. She also focused on temporal and synchronization aspects of distributed multimedia presentations in the presence of resource constraints, and on the modeling and investigation of various aspects of ambient intelligence systems.
Maria Luisa Sapino has been serving as a reviewer for several international conferences and journals in the area.