future internet

Review

SAES: An Introduction to Self-Adapting Exploratory Structures

Giovanni Maria Sacco

Dipartimento di Informatica, Universita' degli Studi di Torino, I-10124 Torino, Italy; giovanni.sacco@unito.it; Tel.: +39-011-6706727

Received: 31 December 2018; Accepted: 19 February 2019; Published: 26 February 2019

Abstract: Self-adapting exploratory structures (SAESs) are the basic components of exploratory search. They are abstract structures which allow searching or querying of an information base and summarizing of results using a uniform representation. A definition and a characterization of SAES is given, as well as a discussion of structures that are SAES or can be modified in order to become SAES. These include dynamic taxonomies (also known as faceted search), tag clouds, continuous sliders, geographic maps, and dynamic clustering methods, such as Scatter-Gather. Finally, the integration of these structures into a single interface is discussed.

Keywords: exploratory search; dynamic taxonomies; faceted search; tag clouds; Scatter-Gather

1. Introduction

Research prior to 2000 on information systems mainly concentrated on efficient ways of querying large structured databases, and, in later years, large text information bases. Data or text queries were assumed to be precise, e.g., the user was expected to know exactly what she wanted. As an example, consider a user who wants to buy a Sony A7RIII camera: the query "Sony A7RIII" will precisely describe her information needs.

However, it has been increasingly recognized that such a type of search (called the locator phase by Sacco [1]) is the exception rather than the norm in practice, and that users very often do not exactly know what they want, or they are unable to precisely frame their information needs. Consider the user above, but this time let her information need be to find the "best" camera for her needs. This information need is practically impossible to satisfy within the boundaries of traditional query systems. What does "best" mean? It depends on the user's priorities, which can be extremely diverse not only among different users, but even for the same user at different times. It also depends on the features cameras have, such as price, resolution, etc. What is required here is that users be able to explore the information base and freely select the features they are more interested in, until they find a camera that satisfies their needs:

1. The user is shown a summary view of the information base, with available features;
2. From these features, the user selects the most important one;
3. The system adjusts the summary view, as to show only the features present in the items with the selected feature;
4. The user iterates between 2 and 3, until the number of items is sufficiently small for manual inspection. Additional selected features are composed in "AND" with previous ones.

It can be contended that this is the natural interaction in e-commerce [2,3], and not surprisingly it is now the de-facto standard in e-commerce. Implementations of this strategy are based on a model first proposed by Sacco in 1987 [4] (and in an extended version in 2000 [1]), called dynamic taxonomies. Later proposals of this model (such as [5], though informal and usually not as powerful as the original model) made the alternative term "faceted search" catch, even though it is a misnomer. In fact, a faceted taxonomy design, after Ranganathan [6], is just a design guideline, and the term faceted search fails to capture the essential characteristics of the model.

Exploratory search based on dynamic taxonomies has an extremely wide application range: from e-commerce to encyclopedias, to medical diagnosis, to e-government, to multimedia databases, etc. The reader is referred to [7], which is the most complete monograph to date and which describes several case studies and applications.

Are dynamic taxonomies the only way to support exploratory search? Sacco [7] proposed an abstraction of the way dynamic taxonomies work (self-adapting exploratory structures or SAES for short), and indicated a series of features that exploratory structures must exhibit to be useful for exploratory search. In Section 2, this abstraction is reviewed and extended. In Section 3, the most important SAES, dynamic taxonomies, are reviewed. Section 4 discusses other SAES, including extended tag clouds, continuous sliders, geographical maps, and clustering strategies, when used in isolation, i.e., as the only way of accessing the information base. Finally, Section 5 discusses the integration of these structures into a single framework provided by dynamic taxonomies.

The focus of this paper is an introduction to and a comparison of structures that afford exploration, and only the features of these structures which are relevant for the discussion are considered here. A review of dynamic taxonomies and exploratory search is outside the scope of the present paper, and areas such as taxonomy design [7], integration with RDF [8], or extensions such as facet ranking [7] are not discussed here.

2. Self-Adapting Exploratory Structures

Self-adapting exploratory structures (SAES) present a uniform view of the information base consisting of multiple choices (selectors or filters) and use the same structure to set interest sets (foci) and summarize results. SAES selectors have three fundamental properties:

- Selectors are used to set interest sets (foci), i.e., to "query" the information base;
- Selectors are used to summarize query results, and consequently they should be able to summarize any subset of the information base. This is a fundamental point, and intuitively, there cannot be any effective exploration capabilities in systems which do not support summaries of foci. Without summaries, users are left in the dark as to what choices are allowed to restrict the information base;
- Summaries must be produced in real time, because a slower response time would discourage users from exploration. Since exploratory systems must be able to produce summaries for any subset of the universe, naive strategies that precompute all the possibilities are obviously not applicable because of exponential growth.

SAES can be further characterized according to seven dimensions:

- *Selector composition*. In the simplest case, only one selector can be chosen. In the most general case, any boolean combination of selectors can be used;
- *Multidimensional* vs. *monodimensional selectors*. Monodimensional selectors are characterized by a 1:N relation with items, i.e., a selector can select many items, but any specific item can be selected by only one selector. The traditional Aristotelian taxonomy is an example of a taxonomy with monodimensional selectors. Multidimensional selectors, on the other hand, have a M:N relation with items. Access through monodimensional vs. multidimensional taxonomies vs. dynamic taxonomies was studied in [9]. In extreme synthesis, it was shown that monodimensional taxonomies are not scalable and do not provide sufficient discrimination for large information bases. Multidimensional taxonomies are even worse from this aspect, unless selector composition (which is irrelevant in monodimensional taxonomies) is allowed. The results carry over to flat organizations;

- *Completeness of iteration*. Iteration is complete if the process of selecting a focus and summarizing it can be iterated at will, or at least until a singleton result set is obtained. Incomplete iteration results in the impossibility of further restricting the information base beyond a certain point;
- *Completeness of reach*. Reach is complete if it can be guaranteed that there are no items in the information base which are not reachable through at least one selector. Incomplete reach cause a loss of potentially relevant items because some items are never considered during exploration. Consequently, reach-incomplete structures should never be used as the only access path to an information base, and their usefulness is, in general, questionable;
- *Self-sufficiency*. A SAES is self-sufficient if the use of selectors is always supported, and it is the only required operation, although other types of queries (e.g., text retrieval) can be supported. Non self-sufficient structures force the user to restrict the information base through other strategies before selectors can be used. As an example, many applications require the user to enter a text retrieval query before exploration can start. This is done primarily for performance reasons, and it results in asymmetric interactions. Most importantly, in this approach the user is unable to have summary information for the entire information base;
- *No-zero-result*. This property is satisfied when selectors which do not describe the result are pruned from summaries, or made non selectable. Consequently, the use of a selector is guaranteed never to produce an empty result;
- *Taxonomic* vs. *flat organization*. In the first case, selectors can be organized according to a taxonomy so that the user can select the appropriate level of abstraction, and user orientation among selectors is simpler. A flat organization is appropriate, if at all, only for very simple applications.

3. Dynamic Taxonomies (Faceted Search)

Dynamic taxonomies are a general knowledge management model based on a multidimensional classification of heterogeneous data items and are used to explore or browse complex information bases in a guided yet unconstrained way through a visual interface. It has been applied to very diverse areas, from electronic commerce to medical diagnosis, among many others.

The schema of a dynamic taxonomy is a taxonomy usually designed by an expert, though automatic approaches have been investigated [7]. It does not require any other relationships in addition to subsumptions (e.g., IS-A and PART-OF relationships).

In the extension, items can be freely classified under several topics at any level of abstraction. This multidimensional classification models common real-life situations because items can very often be classified under different concepts, and usually have different independent features (e.g., Time, Location, etc.), each of which can be described by an independent taxonomy. These features are often called perspectives or facets.

In dynamic taxonomies, a concept C is just a label that identifies all the items classified under C. Because of the subsumption relationship between a concept and its descendants, the items classified under C (items(C)) are all those items in the deep extension of C, i.e., the set of items identified by C includes the shallow extension of C (i.e., all the items directly classified under C) in union with the deep extension of C's sons. The shallow and the deep extension for a terminal concept are the same, by construction. This set-oriented approach implies that logical operations on concepts can be performed by the corresponding set operations on their extension, and therefore the user is able to restrict the information base (and to create derived concepts) by combining concepts through all the standard logical operations (and, or, not).

A fundamental feature of this model is that a formal notion of the relationship between two concepts A and B can be given: two concepts A and B are related (A⇔B) iff there is at least one item d in the knowledge base which is classified at the same time under A or under one of A's descendants and under B or under one of B's descendants, or, more formally,

$$\text{A}\Leftrightarrow\text{B iff items(A)} \cap \text{items(B)} \neq \emptyset.$$

For example, we can infer an unnamed relationship between terrorism and New York, if an item classified under terrorism and New York exists. At the same time, since New York is a descendant of USA, also a relationship between terrorism and USA can be inferred. By construction, $A \Leftrightarrow B \equiv B \Leftrightarrow A$. In addition, subsumptions explicitly defined in the schema are, by construction, a special case of concept relationship.

On the basis of this definition, dynamic taxonomies can find all the concepts related to a given concept C: these concepts represent the conceptual summary of C.

The extensional inference rule can be easily extended to cover the relationship between a given concept C and a concept expressed by an arbitrary subset S of the universe: C is related to S iff there is at least one item d in S which is also in items(C), or, equivalently,

$$A \Leftrightarrow B \text{ iff items(C)} \cap S \neq \emptyset.$$

Consequently, the extensional inference rule can produce conceptual summaries not only for base concepts, but also for any logical combination of concepts. Moreover, since it is immaterial how S is produced, dynamic taxonomies can summarize sets of items produced by other retrieval methods, such as information retrieval, database queries, shape retrieval, etc., and therefore access through dynamic taxonomies can be easily combined with any other retrieval method.

Dynamic taxonomies are defined in terms of conceptual descriptions of items, so that heterogeneous items of any type and format can be managed in a single, coherent framework. Finally, since concept C is just a label that identifies the set of the items classified under C, concepts are language-invariant, and multilingual access can be easily supported by maintaining different language directories holding language-specific labels for each concept in the taxonomy.

3.1. Access through Dynamic Taxonomies

The user is initially presented with a tree representation of the initial taxonomy for the entire information base. The system can associate with each concept label a count of all the items classified under it (i.e., the cardinality of items(C) for all C's). This count is an important user feedback in navigation, because when it is sufficiently small, the user usually terminates exploration and inspects the result items.

The initial user focus F is the universe (i.e., all the items in the information base).

In the simplest case, the user can then select a concept C in the taxonomy and "zoom" over it. The zoom operation changes the current state in two ways. First, concept C is used to refine the current focus F, by intersecting it with items(C); items not in the focus are discarded. Second, the tree representation of the taxonomy is modified in order to summarize the new focus. All, and only the concepts related to F, are retained and the count for each retained concept C′ is updated to reflect the number of items in the focus F that are classified under C′.

The reduced taxonomy is a conceptual summary of the set of documents identified by F, exactly in the same way as the original taxonomy was a conceptual summary of the universe. The term *dynamic taxonomy* is used to indicate that the taxonomy can dynamically adapt to the subset of the universe on which the user is focusing, whereas traditional, static taxonomies can only describe the entire universe.

The exploration process is an iterative thinning of the information base: the user selects a focus, which restricts the information base by discarding all the items not in the current focus. Only the concepts used to classify the items in the focus, and their ancestors, are retained. These concepts, which summarize the current focus, are those and only those concepts that can be used for further refinements. From the human computer interaction point of view, users are effectively guided to reach their goal by a clear and consistent listing of all possible alternatives.

Differently from traditional search methods, the exploration process has the goal of reducing the universe to a set of items sufficiently small that they can be manually inspected by the user.

3.2. Benefits of Dynamic Taxonomies

Dynamic taxonomies are SAES, and additionally, they are taxonomically organized, are multidimensional, self-sufficient, no-zero result, and complete from the perspective of iteration and reach.

The advantages of dynamic taxonomies over traditional methods are dramatic in terms of an extremely fast convergence of exploratory patterns and in terms of human factors. Three zoom operations on terminal concepts are sufficient to reduce a 10,000,000-item information base described by a compact taxonomy with 1000 concepts to an average 10 items [9]. Dynamic taxonomies only require a very light theoretical background: namely, the concept of a taxonomic organization and the zoom operation, which seems to be very quickly understood by end-users.

Dynamic taxonomies cleanly separate the process of classifying documents from the use of the classification information in the browsing system and considerably simplify the design of the conceptual taxonomy. First, the extensional inference rule actually performs concept association mining: concept associations, which are often quite dynamic in time, need not be forecasted and accounted for in schema design. In addition, the user is presented with associations the schema designer might not even be aware of.

Second, since dynamic taxonomies synthesize compound concepts, these need usually not be represented explicitly, so that we avoid the exponential growth due to the description of all the possible concept combinations, and the resulting taxonomy is significantly more compact and easier to understand. Sacco [1,7] developed a number of guidelines for taxonomies that are compact and easily understood by users. Some are superficially similar to the basic faceted classification scheme by Ranganathan [6]: the taxonomy is organized as a set of independent, "orthogonal" subtaxonomies (facets or perspectives). As an example, a compound concept, such as "Musicians from Mobile, Alabama" need not be explicitly accounted for, because it can be synthesized from its component concepts: *Arts>Musicians* and *Location>USA>Alabama>Mobile*, where Arts and Location are facets.

It is worth stressing that a faceted organization is just a design strategy for the intension (the taxonomy), in a way similar to Normal Forms for relational schemata, and it is not a substantial aspect of the model: the model does not require any specific taxonomy organization, apart from a multidimensional classification.

4. Other Self-Adapting Exploratory Structures

4.1. Extended Tag Clouds

Tag clouds [10,11] use the most frequent tags occurring in document metadata to summarize a (usually predefined) set of documents. Tag size is used to indicate the relative importance of each tag, and sophisticated geometric arrangements are used to exploit limited space at best. Although there are many variations, tag clouds are normally a summary device. One can extend their definition by

1. Allowing the user to use one or more tags in the cloud as selectors, and
2. Having the cloud dynamically summarize the result of selectors, and, in fact, any subset of the information base.

At this point, we have a SAES: the selectors are the tags in the cloud, which are used both to query and to summarize. Since the computation is dynamic and must be performed in real time, appropriate and optimized tag cloud creation algorithms must be derived, and even if only metadata are addressed, this could prove difficult for large result sets.

In addition, although this structure is complete from an iteration perspective, it is incomplete from a perspective of reach. In general, only the *k* most frequent tags are displayed because of space limitations, and therefore, there is no guarantee that all the objects in the information base are reachable through a query on the structure. Finally, tag clouds normally use a flat organization rather than a taxonomic one.

Another extension arises by considering the terms contained in the body of the documents in the information base, rather than the tags contained in metadata descriptions. This increases the criticality of real-time operations, can only manage textual documents, and does not guarantee completeness of reach any better than normal tags. In addition, the choice of significant terms becomes more involved than with normal tags. As it is well known in information retrieval, frequency alone is not a good indicator of importance: one would probably use one of the various tf-idf measures [12], which combine frequency with specificity.

Extended tag clouds can support selector composition, are inherently multidimensional (because an item can be associated with different tags), complete from the point of view of iteration, but not complete as far as reach is concerned, are self-sufficient, guarantee no-zero-results, and have a flat organization.

In summary, tag clouds should not be used as the only exploratory device in an application, because one cannot guarantee that all the items in the information base are reachable.

4.2. Continuous Sliders and Geographic Maps

Some data domains, such as weights, prices, dates, etc., which are characterized by a potentially high number of different values, can be represented by a slider. A slider can be used as a summary of the values (e.g., prices) in a subset of items by indicating the lowest and highest values, and a thick area between the two values. The slider can be used as a selector, by simply moving the lowest or highest boundary or both, in or out.

Geographic maps are essentially the same device as continuous sliders, but in two dimensions rather than in one dimension only. The user can select specific areas (by drawing a rectangle, a circle, or a polygon on the map), and query results can be displayed on the map itself.

Both devices are indeed SAES, and are complete for iteration and reach, self-sufficient, and guarantee no-zero results. However, they are inherently monodimensional, and after the first query and relative summary are produced, further iterations can only reduce the range of the slider or the area of the map, and thus provide a trivial interaction. For this reason, use of these devices is recommended only in conjunction with other querying methods.

4.3. Clustering Methods

Document clustering groups documents described by features on the basis of a measure of similarity between documents. The reference model is the vector space model, proposed by Salton [13]. Each feature (e.g., a specific weighted term in the corpus dictionary) of the N considered defines a dimension in an N-dimensional space. A specific document d can then be described by a vector in that space, in which components for features not appearing in the document have weight equal to 0. A popular measure of similarity between two documents d and d′ is the cosine of the angle between the vectors representing the documents.

Various clustering strategies can be applied that are based on the measure of similarity selected. The basic idea is to partition the information base into clusters of similar documents, i.e., the distance (the inverse of similarity) between any two documents in a cluster C should be less than the distance between any document in C and any document in any cluster C′ ($C \neq C'$). Clustering can be applied as to produce a flat organization of clusters, or a hierarchical organization, which is, however, not a taxonomic organization because it is based on feature co-occurrence rather than on IS-A relationships [1]. In order to represent a cluster, the centroid of the cluster is used. The centroid is either the vector that averages all the documents in the cluster (the barycenter) or the document closest to the barycenter.

A dynamic reclustering is required to support SAES requirements. Thus, initially the user will be presented with a number of clusters. She selects one or more clusters among the ones presented, and the system will dynamically produce new clusters considering only and all the documents selected by the user. The process can be iterated. This is basically the first formulation of the Scatter-Gather algorithm by Cutting et al. [14]. It is quite obvious that hierarchical clustering is performed, however, as remarked above, it is not based on a taxonomic organization.

Regardless of the variants, clustering exhibits a number of serious problems. First, it is quite difficult to label a cluster in order to convey its meaning, and very often the "meaning" of a cluster is not useful for the user, i.e., documents are clustered according to a criterion which is not useful or understandable by the user [1]. The common labeling strategy is to list the most common features (e.g., terms) in the centroid, which very often do not convey sufficient information and produce large labels difficult to display on the screen.

Second, clustering is very expensive in terms of time (and space). This is not easily reconciled with the need for a dynamic computation required by a SAES. In order to overcome the performance bottleneck, Scatter-Gather was reformulated in the following way [15]. A hierarchy of clusters is precomputed. The simple expansion of a cluster does not require any clustering since its children were precomputed. Scatter (i.e., dynamic reclustering) is used when the user takes the union of two or more clusters at the same level. In this case, only the centroids of the children clusters, rather than the entire set of documents, are reclustered, in order to achieve a reasonable performance.

There are a number of objections, in addition to the experimental evidence found in [16], showing that the effectiveness of Scatter-Gather for conceptual retrieval is inherently poor. First, as mentioned above, cluster labeling in a way that it is understandable by users is a difficult task. This problem is somewhat mitigated because manual labeling is viable for the clusters of the precomputed hierarchy. This is, however, not true for clusters derived by scatter (i.e., reclustering).

Second, since each document belongs to one and only one terminal cluster by construction, the hierarchical organization is really a monodimensional classification. Monodimensional taxonomies were shown [9] to be extremely poor for retrieval on large information bases because they require a very large number of terminal concepts (i.e., terminal clusters, here). It could be contended that scatter could make the situation better, but this is not the case, since reclustering is performed on the union of clusters (which increases the result set) rather than on the intersection of concepts (which decreases the result sets), as in dynamic taxonomies. In other words, scatter is not useful for refinement, but only, possibly, for improving the selection of interesting items. Even this is questionable, since a clustering based on centroids only is intuitively bound to produce an inaccurate clustering, as recognized by the authors.

In summary, the monodimensional classification (each document can belong to only one terminal cluster) of Scatter-Gather and other hierarchical clustering strategies produce search structures whose refining power is no better than monodimensional taxonomies, which were proven inadequate for large information bases [9]. In addition, computational demands are high, unless shortcuts are taken, and interaction problems regarding cluster labeling are difficult to solve.

5. Integration of Different SAES into a Single Interface

Although different SAES exist, only dynamic taxonomies can fully support exploratory search. In fact, all the SAES reviewed above, with the exception of extended tag clouds, are monodimensional, and consequently offer very limited or trivial exploration capabilities, akin to the capabilities offered by traditional monodimensional taxonomies. The only possible action is refinement on the branch initially chosen. Although extended tag clouds are multidimensional, and thus support exploration, they do not guarantee completeness of reach. Therefore, strategies other than dynamic taxonomies cannot be used by themselves to provide full and effective exploratory search capabilities. However, this does not mean they are useless. An interesting strategy is to combine one or more of these methods with dynamic taxonomies, thus leveraging its efficient exploration capabilities in order to support different views that can be interesting for the user.

One of the most important byproducts of the effort in making dynamic taxonomies a minimal model is the ability to see what the fundamental properties of the model are. The one property that can be used here is the definition of concepts. In dynamic taxonomies, a concept is a label that defines the set of documents classified under it; a concept is a totally abstract entity and the label itself is also abstract and does not need to be textual. From here, an important extension of the model was introduced by Sacco [7]: virtual concepts; that is, concepts that are not represented directly, but that are placeholders that interact with external data by providing abstract functions to query external data sources and to summarize subsets of documents. In short, virtual concepts do not only provide a seamless integration with external data sources, such as relational databases, but also provide a way for integrating any SAES into the framework of dynamic taxonomies. Such integration can be done at any level in the taxonomy, but using a SAES as a top-level facet in the taxonomy gives more flexibility in the definition of the interface, since SAES can be placed at top-level in any position in the screen without any problems of user disorientation.

Integration of continuous sliders and geographic maps [17,18] is especially beneficial. Continuous sliders can more effectively deal with number and date ranges than a hierarchical representation of values, both from the query point of view and the summary point of view. Geographic maps are an invaluable device to provide spatial information in many applications, such as real estate, hotel booking, etc. As an example, a user can select desired hotel features (number of stars, free wifi, etc.) and immediately see selected hotels on the map. Conversely, the user can select an area on the map, and have a taxonomic summary of the features for hotels in that area. So, while continuous sliders and maps are virtually useless when used in isolation, their integration with a dynamic taxonomy proves very beneficial.

Integration of extended tag clouds into the dynamic taxonomy framework is more questionable. Tag clouds are usually based on metadata tags, and consequently on some sort of classification alternative to the one used by dynamic taxonomies. However, it hardly beneficial to have more than one classification, one of which is taxonomic and the other one flat. In addition, a dynamic taxonomy can be constructed on the basis of metadata tags, making the tag classification moot.

A more interesting application derives from extended tag clouds based on textual document terms rather than metadata. In this case, the feedback of popular terms could be effectively used to improve text retrieval queries that are supported, in combination with dynamic taxonomies, for querying textual databases. As remarked above, however, the computation involved might make this approach too expensive for large information bases, and requires further research.

Using a static or a dynamic clustering scheme, such as Scatter-Gather, as a SAES facet would seem to provide little benefits, because of the discussion above. However, this is not true. Embedding predefined (flat or hierarchical) clusters, or a mechanism such as Scatter-Gather as a facet of a dynamic taxonomy, completely eliminates the main objection to using these devices in isolation: namely that they produce, by construction, a monodimensional classification that has an insufficient discrimination for large databases. While a monodimensional classification is still produced, this concerns only a single facet of the dynamic taxonomy: items classified under this facet will be classified under other facets also, so that the entire structure is multidimensional and allows a much better resolution.

It becomes questionable whether the additional reclustering offered by Scatter-Gather is really worthwhile in this context, since a better conceptual manipulation is offered by the hosting dynamic taxonomy. It seems that a precomputed hierarchical clustering would be adequate and this would also solve the problem of labeling clusters, since their number would be sufficiently small for manual labeling.

Although the discussion focused on clustering for text documents, clustering is really an abstract strategy that groups abstract items on the basis of the similarity of their features. As a consequence, clustering can be, and was, applied to other domains, for example image retrieval. The reader is referred to [19], where the "Rosso Tiziano" image exploration system is described. This system was the first one to combine primitive image features, clustering, and metadata into a single exploratory system by embedding them in a dynamic taxonomy.

6. Conclusions

We reviewed and extended the characterization of self-adjusting exploratory structures, which represent the fundamental building block in exploratory search. Different types of SAES were reviewed and classified according the general characterization of SAES. It was concluded that no SAES among those surveyed above, except dynamic taxonomies, are able to efficiently support exploratory search by itself. However, it was shown that not only the shortcomings of other SAES can be overcome by integrating them as a top-level facet in a dynamic taxonomy, but that the result of such integration can be quite advantageous from the point of view of richness and ease of interaction.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sacco, G.M. Dynamic taxonomies: A model for large information bases. *IEEE Trans. Knowl. Data Eng.* **2000**, *12*, 468–479. [CrossRef]
2. Sacco, G.M. The intelligent e-sales clerk: The basic ideas. In Proceedings of the Human-Computer Interaction (INTERACT) '03: IFIP TC 13 International Conference, Zürich, Switzerland, 1–5 September 2003.
3. Sacco, G.M. The intelligent e-store: Easy interactive product selection and comparison. In Proceedings of the 7th IEEE International Conference on E-Commerce Technology (CEC-2005), Munich, Germany, 19–22 July 2005.
4. Sacco, G.M. Navigating the CD-ROM. In Proceedings of the International Conference on Business of CD-ROM, Rome, Italy, 1987.
5. Hearst, M.; Elliott, A.; English, J.; Sinha, R.; Swearingen, K.; Yee, K.-P. Finding the flow in web site search. *Commun. ACM* **2002**, *45*, 42–49. [CrossRef]
6. Ranganathan, S.R. *The Colon Classification*; Rutgers University Press: New Brunswick, NJ, USA, 1965.
7. Sacco, G.M.; Tzitzikas, Y. (Eds.) *Dynamic Taxonomies and Faceted Search—Theory, Practice, and Experience*; The Information Retrieval Series; Springer: New York, NY, USA, 2009; Volume 25.
8. Tzitzikas, Y.; Manolis, N.; Papadakos, P. Faceted exploration of RDF/S datasets: A survey. *J. Intell. Inf. Syst.* **2017**, *48*, 329–364. [CrossRef]
9. Sacco, G.M. Analysis and validation of information access through mono, multidimensional and dynamic taxonomies. In Proceedings of the 7th International Conference on Flexible Query Answering Systems (FQAS 2006), Milan, Italy, 7–10 June 2006; Springer: Berlin, Germany, 2006; Volume 4027.
10. Halvey, M.J.; Keane, M.T. An assessment of tag presentation techniques. In Proceedings of the 16th International Conference on World Wide Web (WWW '07), Banff, AB, Canada, 8–12 May 2007; ACM Press: New York, NY, USA, 2007.
11. Dimitrov, D.; Helic, D.; Strohmaier, M. Tag-Based Navigation and Visualization. In *Social Information Access: Systems and Technologies*; Springer International Publishing: Cham, Switzerland, 2018.
12. Baeza-Yates, R.; Ribeiro-Neto, B. *Modern Information Retrieval*; Addison Wesley: Boston, MA, USA, 2011.
13. Salton, G. *The SMART Retrieval System—Experiments in Automatic Document Processing*; Prentice-Hall: Englewood Cliffs, NJ, USA, 1971.
14. Cutting, D.R.; Karger, D.R.; Pedersen, J.O.; Tukey, J.W. Scatter/gather: A cluster-based approach to browsing large document collections. In Proceedings of the 15th Annual International SIGIR Conference on Research and Development in Information Retrieval (SIGIR'92), Copenhagen, Denmark, 21–24 June 1992; ACM Press: New York, NY, USA, 1992.
15. Cutting, D.R.; Karger, D.R.; Pedersen, J.O. Constant interaction-time scatter/gather browsing of very large document collections. In Proceedings of the 16th Annual International Conference on Research and Development in Information Retrieval (SIGIR'93), Pittsburgh, PA, USA, 27 June–1 July 1993; ACM Press: New York, NY, USA, 1993.
16. Pirolli, P.; Schank, P.; Hearst, M.; Diehl, C. Scatter/Gather Browsing Communicates the Topic Structure of a Very Large Text Collection. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Vancouver, BC, Canada, 13–18 April 1996.
17. Stadler, C.; Martin, M.; Auer, S. Exploring the web of spatial data with facete. In Proceedings of the 23rd International Conference on World Wide Web (WWW '14 Companion), Seoul, Korea, 7–11 April 2014.
18. Lionakis, P.; Tzitzikas, Y. PFSgeo: Preference-Enriched Faceted Search for Geographical Data. In Proceedings of the OTM Confederated International Conferences "On the Move to Meaningful Internet System", Rhodes, Greece, 23–28 October 2017.
19. Sacco, G.M. Rosso Tiziano: A system for user-centered exploration and discovery in large image information bases. In Proceedings of the 19th International Conference on Database and Expert Systems Applications (DEXA 2008), Turin, Italy, 1–5 September 2008; Springer: Berlin, Germany, 2008.