IX

Presentation-Layer Considerations for Browsing and Query Refinement

Human-centered knowledge discovery places great emphasis on the presentation layer of systems used for data mining. All text mining systems built around a human-centric knowledge discovery paradigm must offer a user robust browsing capabilities as well as abilities to display dense and difficult-to-format patterns of textual data in ways that foster interactive exploration.

A robust text mining system should offer a user control over the shaping of queries by making search parameterization available through both high-level, easy-to-use GUI-based controls and direct, low-level, and relatively unrestricted query language access. Moreover, text mining systems need to offer a user administrative tools to create, modify, and maintain concept hierarchies, concept clusters, and entity profile information.

Text mining systems also rely, to an extraordinary degree, on advanced visualization tools. More on the full gamut of visualization approaches – from the relatively mundane to the highly exotic – relevant for text mining can be found in Chapter X.

IX.1 BROWSING

Browsing is a term open to broad interpretation. With respect to text mining systems, however, it usually refers to the general front-end framework through which an enduser searches, queries, displays, and interacts with embedded or middle-tier knowledge-discovery algorithms.

The software that implements this framework is called a *browser*. Beyond their ability to allow a user to (a) manipulate the various knowledge discovery algorithms they may operate and (b) explore the resulting patterns, most browsers also generally support functionality to link to some portion of the full text of documents underlying the patterns that these knowledge discovery algorithms may return.

Usually, browsers in text mining operate as a user interface to specialized query languages that allow parameterized operation of different pattern search algorithms, though this functionality is now almost always commanded through a graphical user interface (GUI) in real-world text mining applications. This means that, practically,

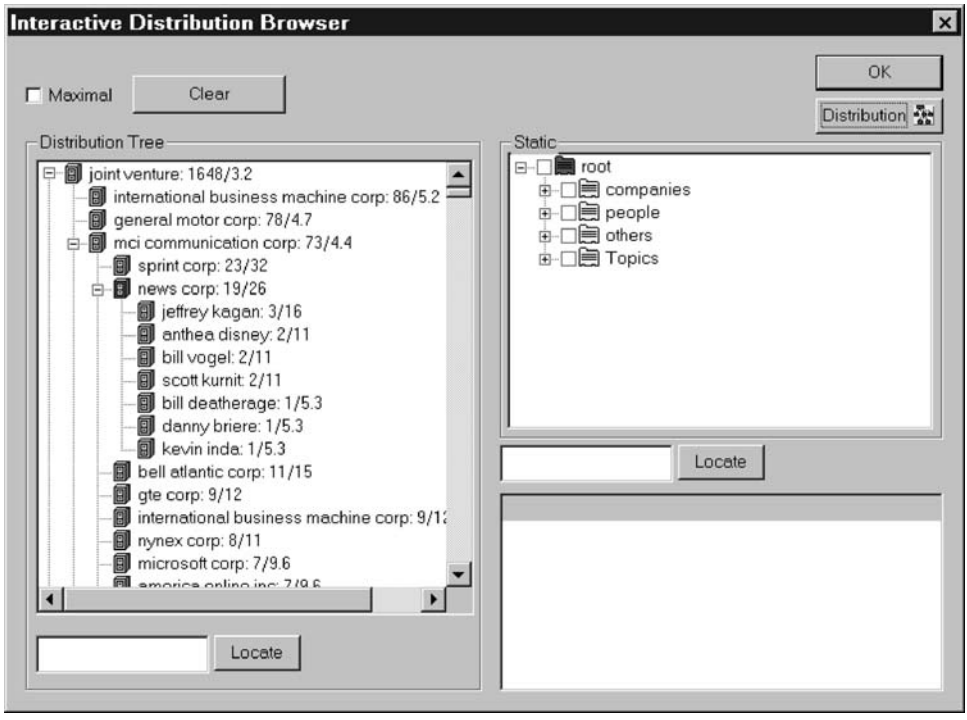

Figure IX.1. Example of an interactive browser for distributions. (From Feldman, Fresko, Hirsh, et al. 1998.)

many discovery operations are "kicked off" by a query for a particular type of pattern through a browser interface, which runs a query argument that executes a search algorithm. Answers are returned via a large number of possible display modalities in the GUI, ranging from simple lists and tables to navigable nodal trees to complex graphs generated by extremely sophisticated data visualization tools.

Once a query is parameterized and run, browsers allow for the exploration of the potentially interesting or relevant patterns generated by search operations. On a basic level, the search algorithms of the core mining operations layer have to process search spaces of instances for a selected pattern type.

This search, however, is structured in relation to certain specified search constraints, and appropriate refinement strategies and pruning techniques are chosen. Such constraints and pruning approaches can be partly or fully specified through a browser interface, though the logic of such refinement techniques may, from a system architecture perspective, reside in as a separate set of services that may be invoked by both presentation-layer and search algorithm components.

All patterns can be studied in the context of a conditioning concept set or context free (i.e., for the general domain of the whole collection). Conditioning a search task therefore means selecting a set of concepts that is used to restrict an analysis task (e.g., a restriction to documents dealing with *USA* and *economic issues* or *IBM* and *hard drive components*). For example, Figure IX.1 shows a simple distribution browser that allows a user to search for specific distributions while looking at a concept hierarchy to provide some order and context to the task.

Many text mining systems provide a heterogeneous set of browsing tools customized to the specific needs of different types of "entities" addressed by the system. Most text mining systems increase the opportunities for user interactivity by offering the user the ability to browse, by means of visual tools, such entities as documents, concept distributions, frequent sets, associations, trends, clusters of documents, and so on. Moreover, it is not uncommon for text mining systems to offer multiple methods for browsing the same entity type (e.g., graphs, lists, and hierarchical trees for documents; maps and hierarchical trees for concept names, etc.).

Although all knowledge discovery operations are susceptible to overabundance problems with respect to patterns, it is typical for text mining systems, in particular, to generate immense numbers of patterns. For almost any document collection of more than a few thousand documents, huge numbers of concept distributions, relations between distributions, frequent concept sets, undirected relations between frequent concept sets, and association rules can be identified.

Therefore, a fundamental requirement for any text mining system's browsing interface is the ability to robustly support the querying of the vast implicit set of patterns available in a given document collection. Practically, however, text mining systems often cope best – and allow users to cope best – with the challenges of pattern overabundance by offering sophisticated refinement tools available while browsing that allow the shaping, constraining, pruning, and filtering of result-set data. Another extremely critical point in managing pattern overabundance is ensuring that the user of a text mining system has an adequate capability for inputting and manipulating what has been referred to as the *measures of interestingness* of patterns in the system.

IX.1.1 Displaying and Browsing Distributions

Traditional document retrieval systems allow a user to ask for all documents containing certain concepts – UK and USA, for example – but then present the entire set of matching documents with little information about the collection's internal structure other than perhaps sorting them by relevance score (which is a shallow measure computed from the frequency and position of concepts in the document) or chronological order.

In contrast, browsing distributions in a text mining system can enable a user to investigate the contents of a document set by sorting it according to the child distribution of any node in a concept hierarchy such as topics, countries, companies, and so on. Once the documents are analyzed in this fashion and the distribution is displayed, a user could, for instance, access the specific documents of each subgroup (see Figure IX.2).

One way to generate a distribution is to provide two Boolean expressions. The first expression could define the selection condition for the documents. The second expression would define the distribution to be computed on the set of chosen documents.

For instance, the user can specify as the selection criteria the expression "*USA* and *UK*" and only documents containing both concepts will be selected for further processing. The distribution expression can be "*topics*," in which case, a set of rules that correlated between *USA* and *UK* and any of the concepts defined under the

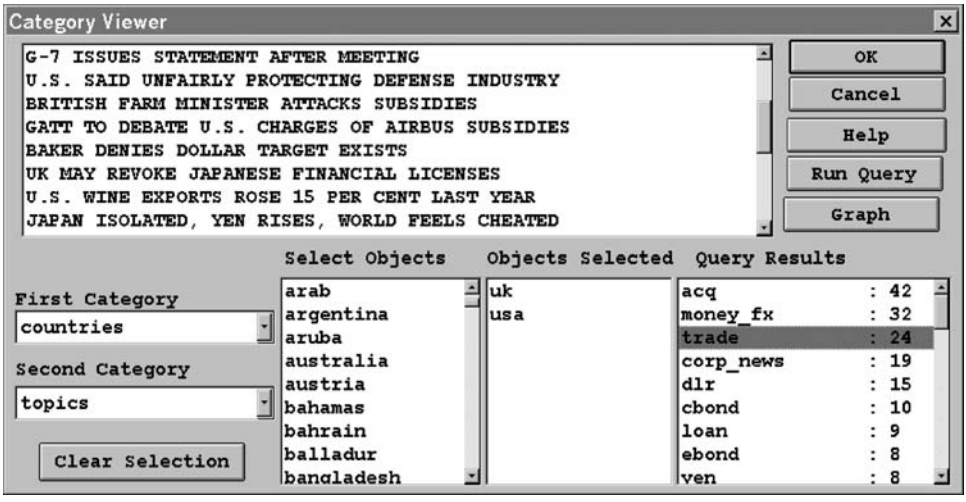

Figure IX.2. Topic (concept) distribution browser from the KDT system selecting for USA and UK. (From Feldman, Dagan, and Hirsh 1998. Reprinted with permission of Springer Science and Business Media.)

node "*topics*" in the taxonomy will be obtained. The results could be shown in a hierarchical way based on the structure of the taxonomy underneath "*topics*."

One can see, for instance, an association rule such as

$$USA, UK \Rightarrow acq\ 42/19.09\%.$$

This rule means that in 19.09 percent of the documents in which both *USA* and *UK* are mentioned, the topic acquisition is mentioned too, which amounts to 42 documents. The user could then click on that rule to obtain the list of 42 documents supporting this rule.

A second association rule could be

$$USA, UK \Rightarrow currency\ 39/17.73\%.$$

In this example, let us assume that *currency* is an internal node and not a concept found in the documents. The meaning of the rule, therefore, is that, in 17.73 percent of the documents in which both *USA* and *UK* are mentioned, at least one of the topics underneath the node "*currency*" in the taxonomy is mentioned too, which amounts to 39 documents.

The user could then expand that rule and get a list of more specialized rules, where the right-hand side (RHS) of each of them would be a child of the node "*currency*." In this case, one would find *UK* and *USA* to be highly associated with *money_fx* (foreign exchange), *dlr* (US Dollar), and *yen*.

IX.1.2 Displaying and Exploring Associations

Even when data from a document collection are moderately sized, association-finding methods will often generate substantial numbers of results. Therefore, association-discovery tools in text mining must assist a user in identifying the useful results out of all those the system generates.

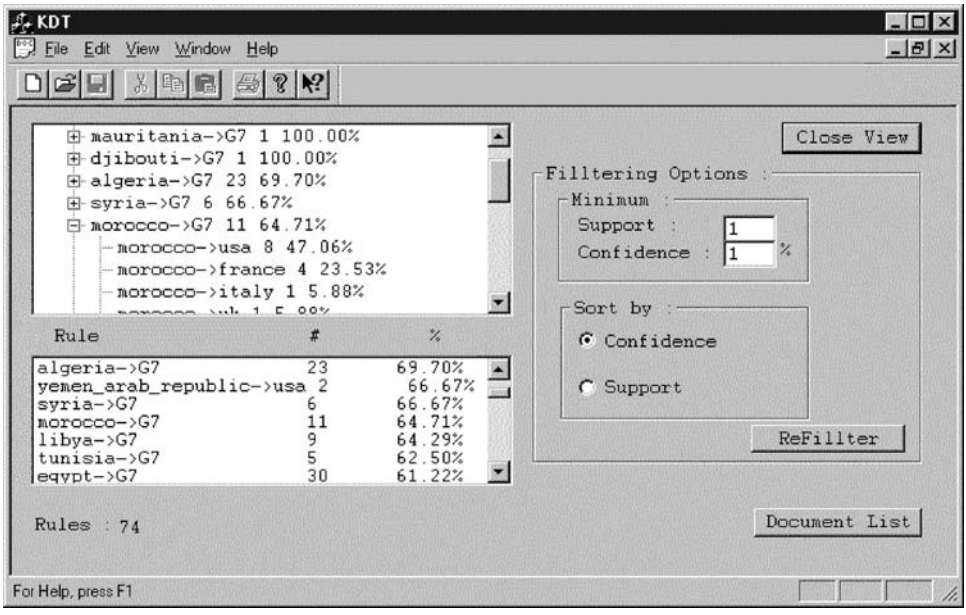

Figure IX.3. An example of an advanced tool for browsing and filtering associations. (From Feldman, Kloesgen, Ben-Yehuda, et al. 1997.)

One method for doing this is to support association browsing by clustering associations with identical left-hand sides (LHSs). Then, these clusters can be displayed in decreasing order of the generality of their LHS.

Associations that have more general LHSs will be listed before more specific associations. The top-level nodes in the hierarchical tree are sorted in decreasing order of the number of documents that support all associations in which they appear.

Some text mining systems include fully featured, association-specific browsing tools (see Figures IX.3 and IX.4) geared toward providing users with an easy way for finding associations and then filtering and sorting them in different orders.

This type of browser tool can support the specification of simple constraints on the presented associations. The user can select a set of concepts from the set of all possible concepts appearing in the associations and then choose the logical test to be performed on the associations.

In even a simple version of this type of tool, the user can see either all associations containing either of these concepts (or), all of these concepts (and), or that the concepts of the association are included in the list of selected concepts (subset). He or she could then also select one of the internal nodes in the taxonomy, and the list of concepts under this node would be used in the filtering.

For instance, if one set the support threshold at 10, and the confidence threshold at 10 percent, an overwhelming number of associations would result. Clearly, no user could digest this amount of information.

An association browser tool, however, would allow the user to choose to view only those associations that might contain, for instance, both the concepts *USA* and *acq* (a shorthand concept label for "company acquisition"). This would allow him or her to see what countries are associated with *USA* with regard to acquisition along with all the statistical parameters related to each association.

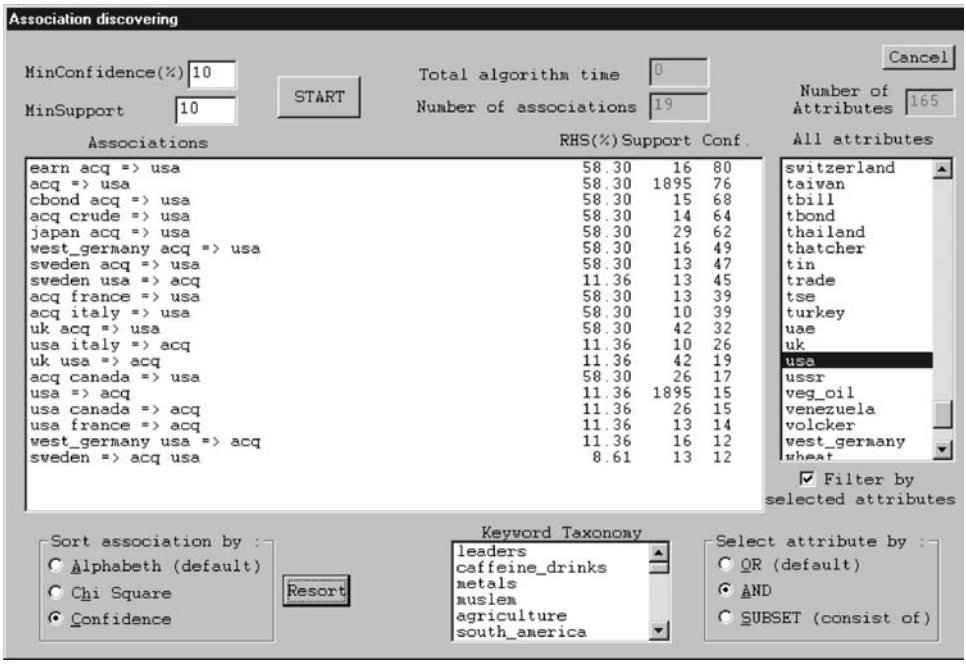

Figure IX.4. An example of another GUI tool for displaying and browsing associations. (From Feldman, Kloesgen, Ben-Yehuda, et al. 1997.)

The utility afforded by even relatively simple techniques, such as sorting, and browsers can provide a user with several sorting options for associations. Two options are rather obvious: sorting the associations in alphabetical order, and sorting the associations in decreased order of their confidence. A third ordering scheme is based on the chi-square value of the association. In a way, this approach attempts to measure how different the probability of seeing the RHS of the association is given that one saw its LHS from the probability of seeing the RHS in the whole population.

IX.1.3 Navigation and Exploration by Means of Concept Hierarchies

Concept hierarchies and taxonomies can play many different roles in text mining systems. However, it is important not to overlook the usefulness of various hierarchical representations in navigation and user exploration.

Often, it is visually easier to traverse a comprehensive tree-structure of nodes relating to all the concepts relevant to an entire document collection or an individual pattern query result set than to scroll down a long, alphabetically sorted list of concept labels. Indeed, sometimes the knowledge inherent in the hierarchical structuring of concepts can serve as an aid to the interactive or free-form exploration of concept relationships, or both – a critical adjunct to uncovering hidden but interesting knowledge.

A concept hierarchy or taxonomy can also enable the user of a text mining system to specify mining tasks concisely. For instance, when beginning the process of generating association rules, the user, rather than looking for all possible rules, can specify interest only in the relationships of companies in the context of business alliances.

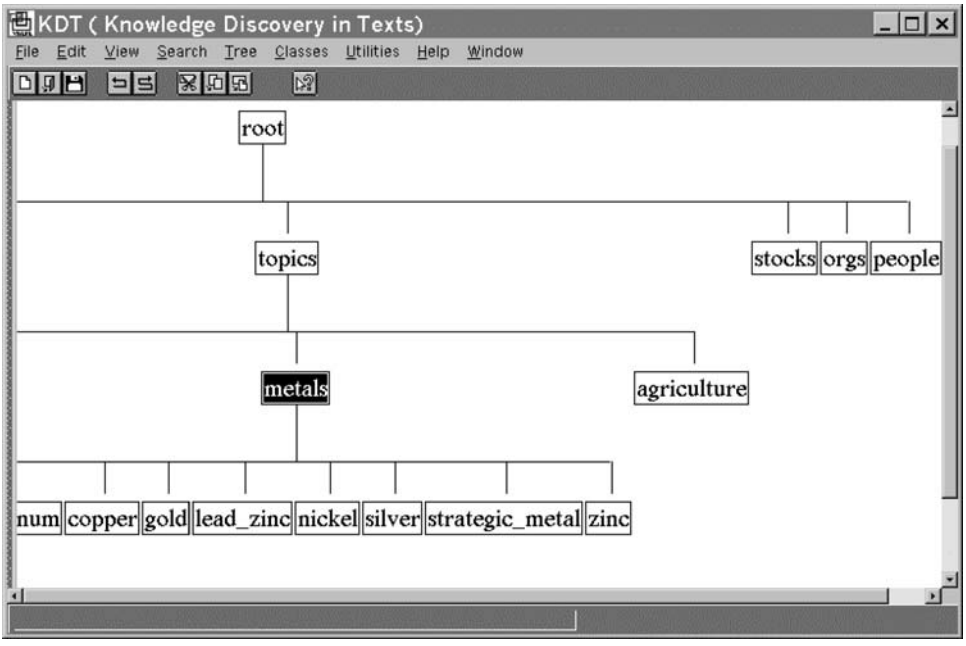

Figure IX.5. A simple graphical interface for creating, exploring, and manipulating a Taxonomy. (From Feldman, Dagan, and Hirsh 1998. Reprinted with permission of Springer Science and Business Media.)

To support this, the text mining system could display a concept hierarchy with two nodes marked "business alliances" and "companies," for instance. The first node would contain terms related to business alliances such as "joint venture," "strategic alliance," "combined initiative," and so on, whereas the second node would be the parent of all company names in the system (which could be the result of human effort specifying such a higher level term, but in many text mining systems a set of rules is employed with knowledge extracted from Internet-based or other commercial directories to generate company names).

In this example, the user could perform a comprehensive search with a few clicks on two nodes of a hierarchical tree. The user would thus avoid the kind of desultory, arbitrary, and incomplete "hunting and pecking" that might occur if he or she had to manually input from memory – or even choose from a pick list – various relevant words relating to business alliances and companies from memory to create his or her query. A very simple graphical display of a concept hierarchy for browsing can be seen in Figure IX.5.

In addition, concept hierarchies can be an important mechanism for supporting the administration and maintenance of user-defined information in a document collection. For instance, entity profile maintenance and user-specified concept or document clustering can often be facilitated by means of the quick navigational opportunities afforded by tree-based hierarchical structures.

IX.1.4 Concept Hierarchy and Taxonomy Editors

Maintaining concept hierarchies and taxonomies is an important but difficult task for users of the text mining systems that leverage them. Therefore, presentation-layer

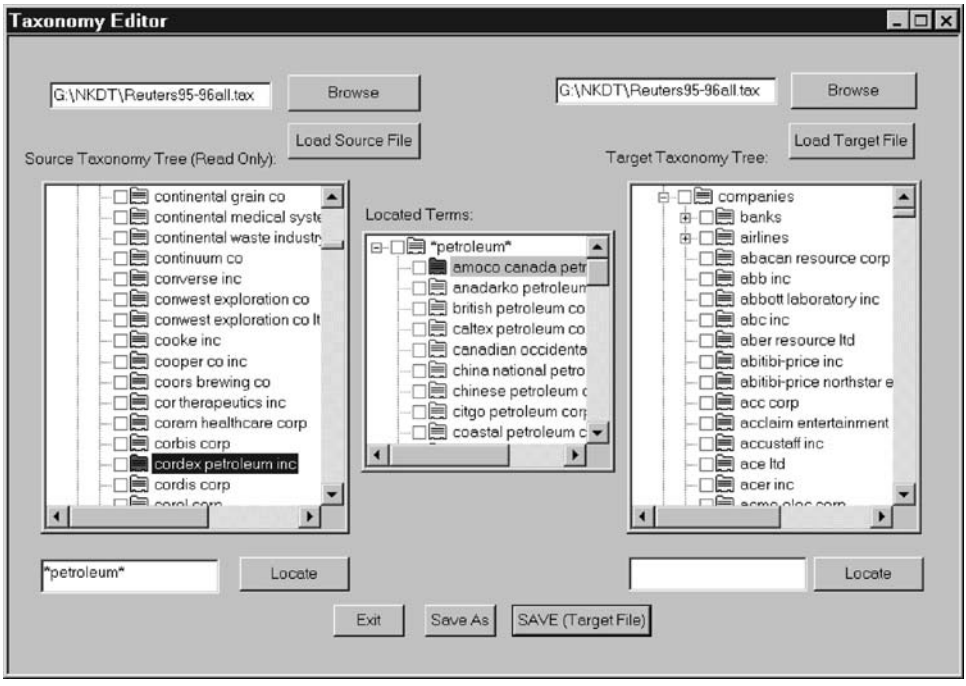

Figure IX.6. User interface for a taxonomy editor showing views of source and target taxonomy trees. (From Feldman, Fresko, Hirsh, et al. 1998.)

tools that allow for easier and more comprehensive administration serve an important role in increasing the usability and effectiveness of the text mining process.

Concept hierarchy editing tools build on many of the same features a user needs to employ a concept hierarchy as a navigational tool. The user must be able to search and locate specific concepts as well as hypernyms and hyponyms; fuzzy search capability is an important adjunct to allowing a user to scrub a hierarchy properly when making major category changes. An example of a graphical hierarchy editing tool appears in Figure IX.6.

Moreover, an important feature in such an editor can be the ability to view the existing *source* concept hierarchy in read-only mode and to edit a *target* concept hierarchy at the same time. This can help a user avoid making time-consuming errors or creating inconsistencies when editing complex tree-structures or making wholesale modifications.

IX.1.5 Clustering Tools to Aid Data Exploration

Although several methods for creating smaller subset-type selections of documents from a text mining system's main document collection have already been discussed, there are numerous situations in which a user may want to organize groups of documents into *clusters* according to more complex, arbitrary, or personal criteria.

For instance, a user of a text mining system aimed at scientific papers on cancer research may want to cluster papers according to the biomedical subdiscipline

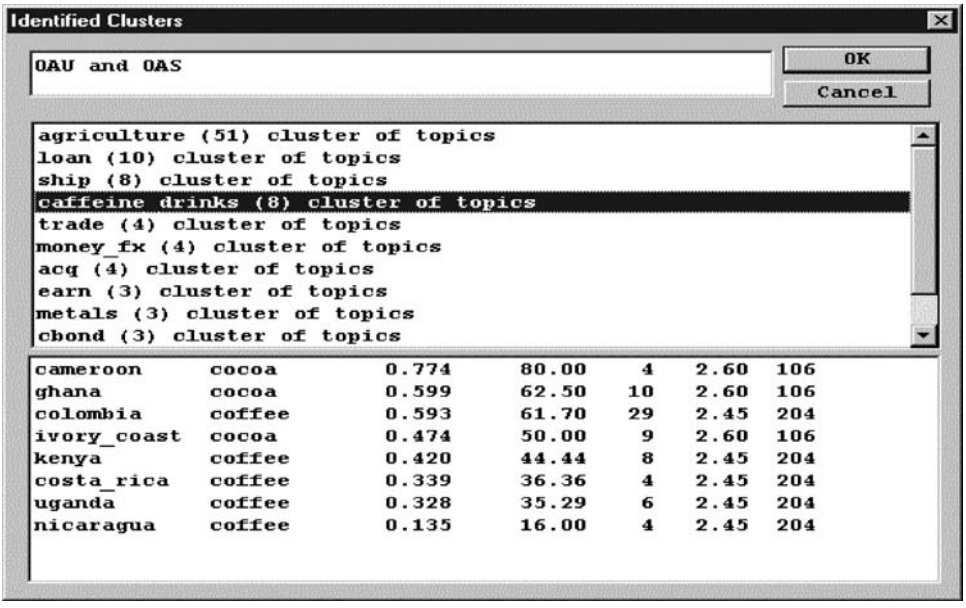

Figure IX.7. Clustering associations using a category hierarchy. (From Feldman, Dagan, and Hirsh 1998. Reprinted with permission of Springer Science and Business Media.)

(e.g., immunology, microbiology, virology, molecular biology, human genetics, etc.) of each paper's lead author. Similarly, a user of a document collection composed of news feeds might want to leverage his or her text mining system's concept hierarchy to cluster patterns involving individual countries under labels representing larger, intercountry groupings (see Figure IX.7).

Clustering operations can involve both automatic and manual processes. Unlike classic taxonomies, groupings of clusters do not need to be strictly hierarchical in structure; individual text mining systems may adopt more or less flexible approaches to such groupings. For this reason, it is generally a requirement that a text mining system offer robust and easy interfaces for a user to view, scrub, and maintain cluster information. Moreover, because both document collections and users' needs can change over time, it is especially important for text mining clustering capabilities to allow flexible reorientation of clusters as a system evolves and matures.

Some text mining systems perform the majority of their manual or unsupervised clustering during preprocessing operations. In these cases, it is still often important to provide users with administrative capability to tweak clusters over the lifetime of a text mining application's use.

IX.2 ACCESSING CONSTRAINTS AND SIMPLE SPECIFICATION FILTERS AT THE PRESENTATION LAYER

Given the immense number of prospective potential patterns that they might identify, text mining systems generally provide support for some level of user-specifiable constraints. These constraints can be employed to restrict the search to returning particular patterns, to limit the number of patterns presented, to offer options for specifying the interestingness of results, or to accomplish all of these objectives.

From a system architecture perspective, the logic of such constraints should be seen more as refinement techniques, and not so much as presentation-layer elements. From a user perspective, however, such constraints and filters are invoked and modulated through the user interface. Therefore, constraint types can be discussed in relation to other elements that can be employed to shape queries through a presentation-layer interface.

Four common types of constraints are typical to text mining browser interfaces:

- ***Background Constraints*** refer to the knowledge of the domain that is given in the form of binary relations between concepts. For example, rules associating persons and countries can be constrained by the condition that an association between a person and a country excludes the nationality of that person. Background constraints typically require a set of predicates to be created relating to certain types of concepts (e.g., entities) in the text mining system's document collection. Binary predicates can allow one input argument and one output argument. Such predicates are usually extracted from some expert or "gold standard" knowledge source.
- ***Syntactical Constraints*** generally relate to selections of concepts or keywords that will be included in a query. More specifically, they can refer to the components of the patterns, for example, to the left- or right-hand side of a rule or the number of items in the components.
- ***Quality Constraints*** most often refer to support and confidence thresholds that can be adjusted by a user before performing a search. However, quality constraints can also include more advanced, customized statistical measures to provide qualities for patterns. An association rule, for instance, can be additionally specified by the significance of a statistical test, or a distribution of a concept group can be evaluated with respect to a reference distribution. These qualities are then used in constraints when searching for significant patterns.
- ***Redundancy Constraints*** have been described as metarules that determine when a pattern is suppressed by another pattern. For example, a redundancy rule could be used to suppress all association rules with a more special left-hand side than another association rule and a confidence score that is not higher than that of the other more general rule.

Constraints are important elements in allowing a user to efficiently browse patterns that are potentially either incrementally or dramatically more relevant to his or her search requirements and exploration inclinations. Moreover, they can be essential to ensuring the basic usability of text mining systems accessing medium or large document collections.

IX.3 ACCESSING THE UNDERLYING QUERY LANGUAGE

Although graphical interfaces make text mining search and browsing operations easier to conduct for users, some search and browsing activities are facilitated if users have direct access to the text mining system's underlying *query language* with well-defined semantics. Many advanced text mining systems, therefore – in addition to

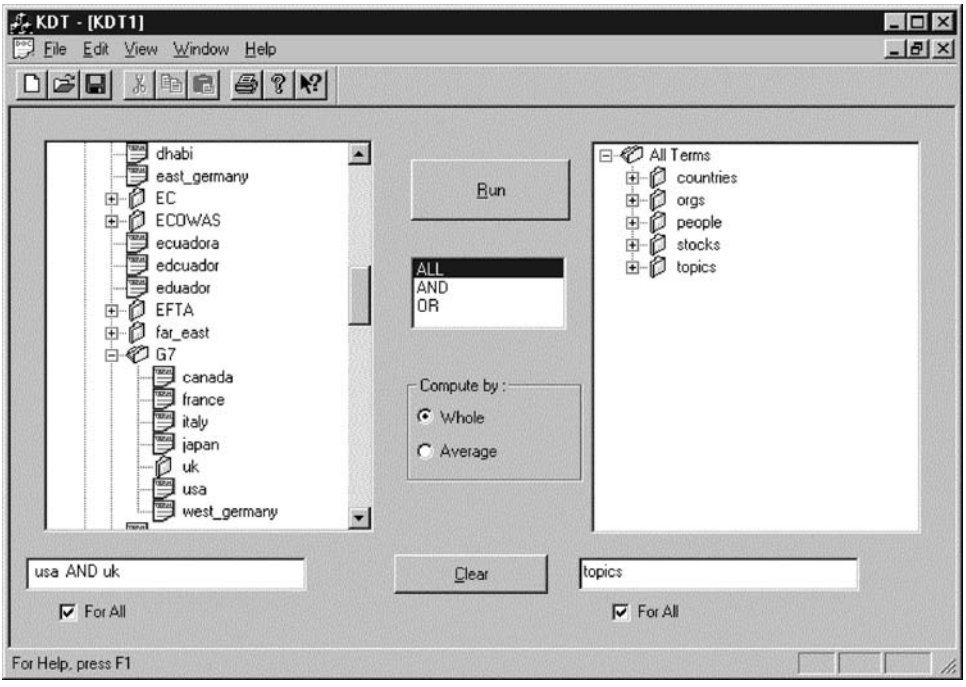

Figure IX.8. Defining a distribution query through a simple GUI in the KDT system. (From Feldman, Kloesgen, Ben-Yehuda, et al. 1997.)

offering pick lists of prespecified query types and common constraint parameters – support direct user access to a query command interpreter for explicit query composition.

Clearly, it is the query language itself that allows a user to search the vast implicit set of patterns available in a given document collection. However, the user environment for displaying, selecting, running, editing, and saving queries should not be given short shrift in the design of a text mining system. Figure IX.8 shows one example of a graphical query construction tool. Regardless of the specific combination of graphical and character-based elements employed, the easier it is for a user to specify his or her query – and understand exactly what that query is meant to return – the more usable and powerful a text mining system becomes.

A more comprehensive discussion of text mining query languages can be found in Section II.3.

IX.4 CITATIONS AND NOTES

Section IX.1

Many of the ideas in Section IX.1 represent an expansion and updating of ideas introduced in Feldman, Kloesgen, Ben-Yehuda, et al. (1997). Methods for display of associations are treated partially in Feldman and Hirsh (1997). Navigation by concept hierarchies is treated in Feldman, Kloesgen, Ben-Yehuda, et al. (1997); and Feldman, Fresko, Hirsh, et al. (1998). Taxonomy editing tools are briefly discussed in Feldman, Fresko, Hirsh, et al. (1998).

Section IX.2

Presentation-level constraints useful in browsing are considered in Feldman, Kloesgen, and Zilberstein (1997a, 1997b).

Section IX.3

Feldman, Kloesgen, and Zilberstein (1997b) discusses the value of providing users of text mining systems with multiple types of functionality to specify a query.