



Topification

A Topic-based Search Methodology

SUMMARY

Users want to input minimal information, search all data sources and instantly receive most relevant search results.

To achieve this goal, Pliant developed **Topification**, a breakthrough auto-classification technology that:

- Requires no resources to implement or maintain classification.
- Simultaneously manages multiple domains.
- Produces a small number of the most relevant results with just a few mouse clicks.
- Requires minimal computer resources.
- Easily handles tens of millions of documents.

This paper describes current search technologies and how they compare with Topification.

BACKGROUND

Current technologies use six common, classification methods to search. They are:

Manual Approach – Each document is manually assigned to a category. Implementation requires creation of categories/taxonomy.

Dynamic Clustering – Statistical or natural language algorithms are used to cluster documents based on word pattern similarities found in each.

Training Sets (Statistical) – Systems analyze training sets and produce metadata that statistical processes use to categorize documents. Implementation requires creation of categories. Training sets are composed of documents representative of each category.

Training Sets (Neural) – This technique is identical to the Statistical approach above except a neural network (developed from training sets) is employed to categorize documents instead of statistical methods.

Training Sets (Metadata) – Implementers provide categories, sample metadata, and categorization rules. Statistical or natural language techniques are used to develop correlations. Correlations are used to place documents in categories.

Combination Techniques – Combinations of techniques described above are used with an additional option to manually override any part of the process. Implementation requires creation of categories.



All of these techniques rely on software programs. Regardless of how elegant the software algorithms used are, programs cannot really “understand” concepts involved in documents as a person would. Therefore, these techniques sometimes inappropriately categorize documents.

Each of these classification approaches also has significant technical drawbacks, including scalability, resource requirements, performance, and usability issues. These issues are discussed below and compared in Figure 1 on the following page.

Classification Problems

Most users lack detailed knowledge of documents they are searching. Classification does two things to help this problem.

First, classification provides user navigation or filtering options. Users may not know how to express the concept being searched, but they can recognize subjects that contain relevant information once presented.

Second, classification categories can be used to organize documents in clusters of related documents. If a system can use search criteria to infer a category, then users can automatically arrive at an appropriate group of documents.

Difficulties start to occur when there is a large number of categories involved and/or if it is unclear exactly what each category represents. This problem is compounded when documents being searched belong in multiple domains.

For instance, content covering bio-mechanical engineering may involve a medical domain, a legal domain, and an engineering domain. The potential for category confusion (by a person or a system) is obvious. Dynamic clustering methodologies can further aggravate this problem by creating (essentially) meaningless floating categories.

Relevancy of Search Results

Classification methodologies create a number of issues related to search result relevancy. For example:

- When searching, is the user looking for exact query matches or matches to the concept implied by the query?
- Is the user looking for hundreds of documents, dozens of documents, a single document, a single page, or a single paragraph or phrase?
- What is the basis for evaluating whether the document is sufficiently focused and, therefore, relevant?

Current training-set methods and automatic clustering methods have only one perspective for evaluating a user search -- the entire document. Some effects of this characteristic are as follows:

- If searched documents are relatively short, like emails or abstracts, this approach is not much of a problem. If the document is multi-page, it will contain many significant words, which is problematic. The ability to mathematically differentiate relevancy disappears.

SEARCH METHODOLOGIES COMPARISON		Categorization Techniques				
Features	pliant Topification	Training Set Based				
		Manual	Auto Cluster	Statistical	Neural	Meta
Relevancy	●	●	○	●	●	●
Result Set Size (Granularity)	●	○	○	○	○	○
Consistency	●	○	○	●	●	●
New Document Processing Time	●	○	●	●	●	●
Meaningful Categories	●	●	○	○	○	○
Taxonomy Depth	●	●	○	○	○	●
Ability to Understand Classification Process	●	●	○	○	○	●
Ability to Customize or Modify	●	●	○	○	○	●
Overall System Flexibility	●	●	○	○	○	●
Setup and Maintenance						
Setup Time	■	□	■	■	■	■
Setup Skill Level Required	■	□	■	□	□	□
Maintenance Skill Level Requirements	■	□	□	□	□	□
Maintenance Resources Requirements	■	□	■	■	■	■

Key:

- Poor
- Marginal
- Adequate
- Good
- Excellent
- None Req'd
- Low
- Moderate
- High

Figure 1: Methodology Comparisons

- Any one domain has a finite set of significant words used in various ways to convey information. A word that is relevant in one domain may not be relevant when found in another domain. Documents in many categories can satisfy the same concept criteria, yet yield many questionable search results.

How Do Manually Entered Taxonomies Help and What are Their Limitations?

Taxonomies were developed by biologists in the 1800's to classify plants and animals. With regard to searching, it

is sometimes argued that users need to navigate and/or understand their search by relating results to hierarchical categories. In this situation, hierarchical categories and taxonomy are synonymous.

In terms of document classification, "subject" taxonomies are developed for documents. Subjects that make up taxonomy become categories that may or may not be hierarchical.

Manual filing systems (like file cabinets) have subject tags or categories. Similar categories are used by training-based methodologies to allow systems to

associate user queries with content. This approach works well if the number of categories is small. In practical applications, however, categories are limited to about 30,000 in number due to maintenance requirements and a user's capacity to cope with category navigation and understanding.

Consider the case of 30,000 documents and 30,000 categories. One might expect that there would be roughly one document in each category. Reviewing the results of one category (one document) would, therefore, be very easy for a user to perform.

If the number of documents was increased to 3,000,000 and there were still 30,000 categories, there would be an average of 100 documents per category. Reviewing 100 results is difficult and time-consuming.

If the number of documents to be searched was increased once more to 30 million, there might be 1,000 documents in each category. Reviewing a thousand results is overwhelming and impractical.

Another factor to consider is the association between categories and documents. If document contents do not parallel the taxonomy's hierarchy, weak associations between categories and documents will result. Alternatively, if a single document is placed in multiple categories, then a system's ability to differentiate categories is reduced.

Any manually created taxonomy and any manual allocation of documents represents very expensive processes.

All methodologies except dynamic clustering require this effort.

Don't Relevancy Methods Handle the Problem of Large Result Sets?

Relevance methodologies aid in resolving the problem of dealing with thousands of results by displaying the most important results first. Relevancy algorithms rate various aspects of some (but usually not all) results. Algorithms return what they consider most relevant.

If there were only one way to judge relevance, there would not be a problem. However, there are many ways to judge relevancy. In the end, the user is the only authority. This means that most, if not all, relevancy methods are arbitrary.

Studies have shown that almost all user searches are under-defined. The user enters only one or two words to define their initial search often within of a pool of potentially millions of documents. An under-defined query can yield tens of thousands of results. As the number of results gets larger, relevance methodologies break down and computer resource requirements increase exponentially. The ability of algorithms to meaningfully discriminate between results disappears, because relevancy criteria are statistically overwhelmed by ranked result similarities (e.g., one result differs very little from the next.)

So What's The Problem? These Six Methods Work – Don't They?

The realities of today's content management systems are that the



number of electronic documents containing information and knowledge is growing at an exponential rate. This implies that assumptions of small document sets, small categories sets, and single content domains are no longer valid. As discussed above, all current classification approaches have fundamental technical issues associated with scale.

Deploying more computer and people resources to solve this problem is an undesirable and/or economically impractical alternative. So, a fully automated system that is scalable and user-friendly is desirable to meet today's and the future's content management needs.

TOPIFICATION SOLUTION

Topification is a patent pending solution developed by Pliant. Topification uses topics to categorize documents and document content. To understand this new approach, a set of definitions, concepts, rules, and tenets (collectively known as an ontology) is described in Figure 2.

After studying Figure 2, it can be inferred that understanding topics (second order concepts) is much easier than understanding categories (third order concepts). Methodologies validate this when they use training sets or example metadata sets to "define" category "meanings".

Defining a third order concept (an abstraction) requires significant knowledge and understanding. Topics, however, only require knowledge of their domain to be understood. Since topics

are concrete (not abstract), knowledge workers easily recognize their meaning.

Figure 2: Topification Method Definitions

Entity	Rules
Word	- First order concept - Associated with a domain
Phrase	- Two or more words - Second order concept - Associated with a domain
Topic	- Two or more words - Second order concept - Associated with a domain. - Types: concrete ⁽¹⁾ and abstract - Only the concrete type is used
Category	- Statistical composite of topics - Third order concept - All abstractions. - Associated with a domain
Tenet 1	Understanding third order concepts requires an understanding of the taxonomy and a known domain. Outside the taxonomy, third order concepts become ambiguous.
Tenet 2	Understanding a second order concept requires only a known domain.
Tenet 3	First order concepts relate the semantic understanding in a given domain.
(1) Concrete topics are those whose component words are found in the document.	

Category Ambiguity Eliminated

Topification eliminates ambiguity by extracting domain-specific topics from content (any size document) through a special process (patent pending). These topics are universal (e.g., they apply to any content that contains that domain). The result is a classification tool that is static for a given domain.



A typical domain will have a topic set containing in the range of 10^5 - 10^6 topics. This is (1-2) orders of magnitude greater than current categorization techniques that are limited to a practical working range of 30,000 categories.

Implementation of Topification

Pliant's Topification solution automatically assigns topic information to all documents based on each document's content as documents are loaded into the system. Pliant automatically translates structured, unstructured, and object-oriented data into neutral form data elements and tags that allow immediate searches of every document's entire content (see Figure 3 on the following page).

System Generates Relevant Topics

When users enter search terms and the result set is larger than the preset number of results requested by the user, Pliant's system automatically generates a (statistically optimal) results sample set and analyzes the topics found within it. Based on this analysis, statistically significant topics (that provide the best hierarchical coverage for the entire result set) are displayed to users. Users can then select one of these topics to further refine their searches.

Using this approach, usability issues and maintenance concerns associated with large taxonomies are no longer present.

Pliant optionally allows users to add any existing taxonomy or topics desired. These can be used with domain topic sets or used independently.

Relevancy is Achieved with Minimum Effort

Once one of the available topics is selected to further refine a search, subsequent searches will reduce results by roughly 1.5 orders of magnitude. As an example, for an initial result set of 100,000 results, selecting and searching on a topic will typically reduce the follow-on results set to less than 5,000. By selecting and searching on a second topic selection, the next results set will typically be less than 300. Selecting and searching on a third topic will bring the results set down to typically less than 15. So in just a few mouse clicks, users can navigate from an unmanageable number of results to a handful of highly relevant documents.

This relevancy is achieved without intense computational load. Each search in the above example yields results in 150 milliseconds or less.

How Do We Know These Search Results are Relevant?

Users direct searches using concrete topics. Original search words and selected topics provide a high degree of confidence that resulting documents are relevant. This also implies that no relevant documents were excluded during the process.

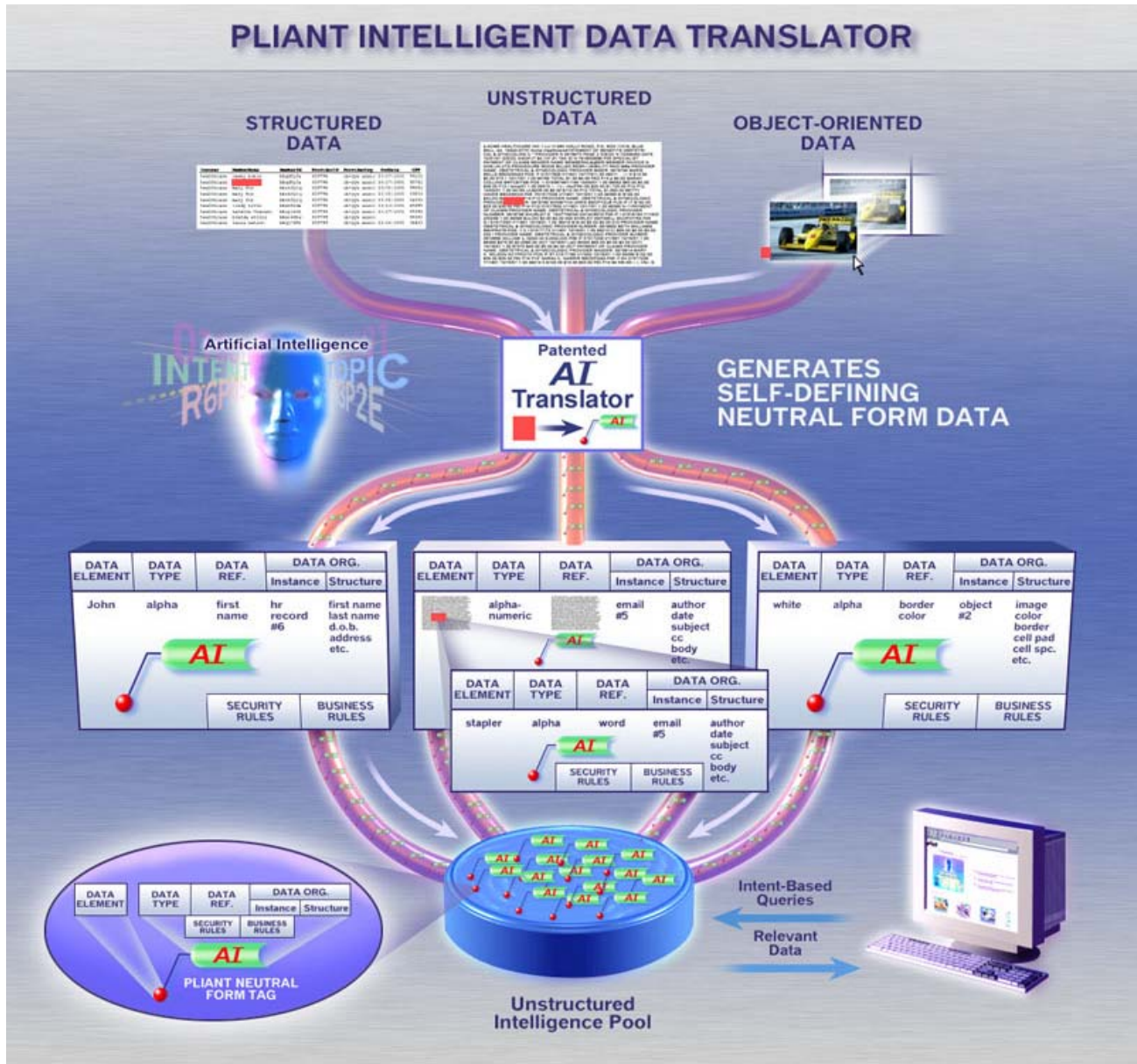


Figure 3: Pliant Data Translator

Classification Errors

Software programs do not classify documents as people do. Pliant's methodology allows users to dynamically classify documents based on their immediate needs and interests.

The previously discussed six current classification methods require categories that users may consider ambiguous. However, with Topification, users work with concrete topics that have far less ambiguity. These topics are always related in some way to information the user has provided.



Hence, the potential for classification error is greatly reduced.

Semantic Issues

Use of only literal forms of search words and topics produces only a subset of the relevant search result set by excluding results with relevant variations in content. Such variations include misspellings (many are usually present in most content), stems, and synonyms. With Topification, as users work to a solution, full semantic expansions of the search are invoked once domain issues have been resolved. This occurs well in advance of producing a final result set and it expands the final result set by about 10% in most cases.

Pliant's Topification Methodology is Unaffected by Scale as Databases Grow

Pliant's Topification methodology is built on a zero-latency architecture. A zero-latency architecture requires no re-indexing when new information is added. As a result, typical limitations, such as maintenance and scale issues, disappear. So managing hundreds of millions of documents with Pliant is not only feasible, but also practical.

Domain Resolution

By definition, topics are domain dependent. With each step in topics selection, users are, therefore, resolving domain issues present in initial search words. Topification is also systematically excluding irrelevant domains that do not overlap with the one of interest. If search terms were, for example, "white house" and a

topic was selected related to "government", results related to the "White House" in Washington, D.C. would be displayed. By contrast, if a topic related to "real estate" was selected, results related to all "white houses for sale" might be displayed.

Granularity Issues

Pliant's algorithm for establishing topics requires that words in a topic must be significant and domain-related. This means Pliant's system does not flag documents just because they contain the right words. The words must be capable of forming the appropriate topic based on the rules outlined in Figure 2.

Topification also reduces large document search sensitivity issues by increasing classification scheme granularity from thousands of categories to a half-million or more topics. More classification options (specific topics vs. relatively broad categories) produce information granularity in a large document. The larger the document, the more likely it is to cover multiple subjects in multiple domains.

Overall, topics are roughly an order of magnitude more specific than a category. Hence, relevant information within a large document can be more easily isolated with topics.

Topic Hierarchies

Topics form a network that has an implied hierarchy. In Figure 4 (on the following page), the relationship between a hypothetical set of topics and documents is displayed. Any given doc-

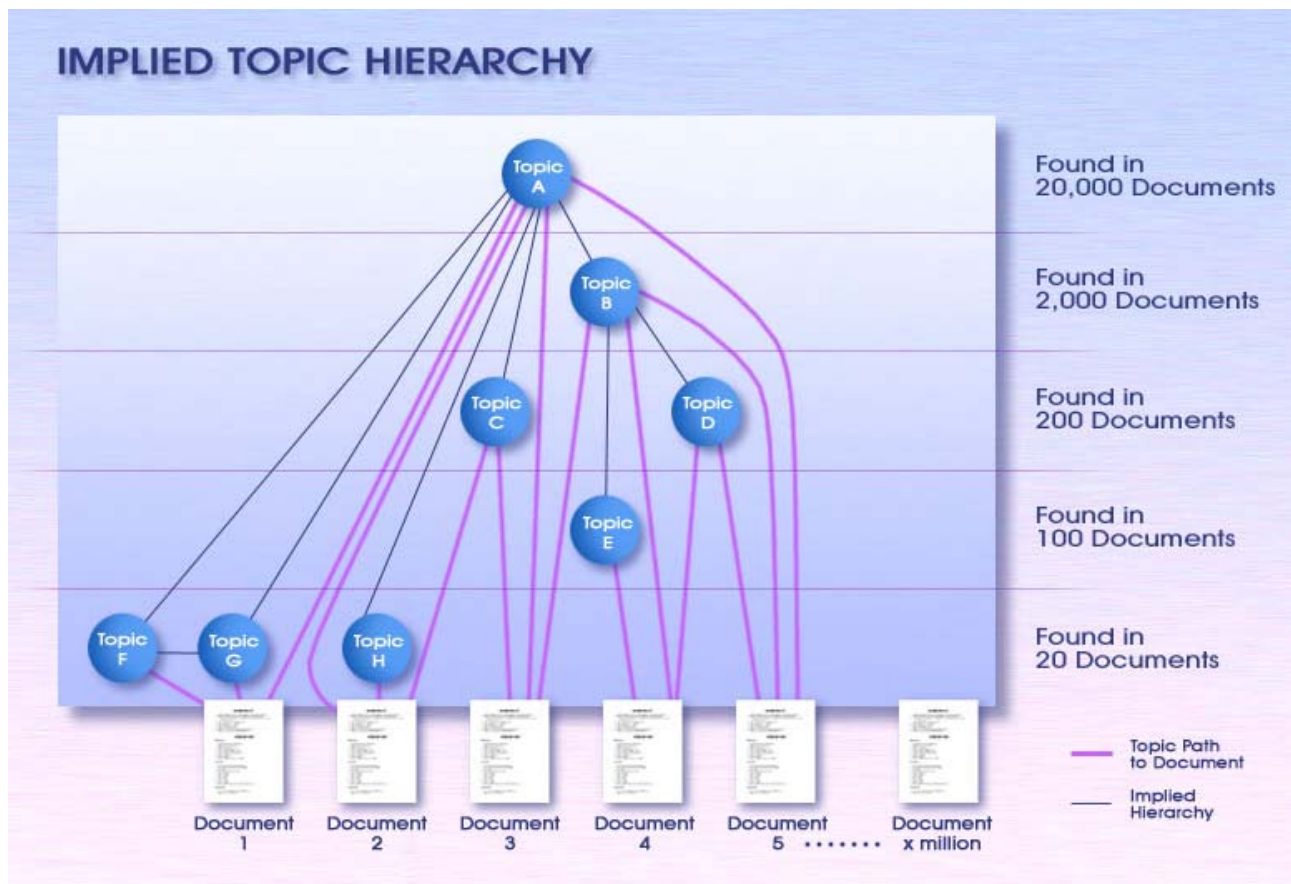


Figure 4: Topification Hierarchies

document contains a set of topics. In the diagram, solid lines represent paths from topics to the documents they are contained in. For example, Topic A is found in Documents 1, 2, 3 and 5 (as well as an approximate 20,000 additional documents). Topic B is found in Documents 3, 4 and 5 (as well as an approximate 2,000 additional documents).

The diagram's bands indicate the (relative) number of documents that contain a given topic. So, Topic A at the top of the diagram is contained in more documents than any other topic. Topic B is found in fewer documents than

Topic A, but in more documents than Topics C, D, E, F, G, or H.

The implied hierarchy is a result of the frequency that a topic occurs in the document set. A topic that appears in many documents is less specific and, therefore, higher in the hierarchy than a topic that appears in just one.

Topic A is related to any topic that occurs with it in a document. For example, Topic A and C both are found in Document 2. Topic A is found in more documents than Topic C, so Topic A is an implied parent of Topic C as expressed by the line connecting both.



Topification uses this topic hierarchy network approach to navigate and display search results. Topic networking characteristics become apparent when studying paths to Document 4. Topic A is not found in Document 4, but both Topic B and Topic D are found in other documents with Topic A. If Topic A is selected as a search constraint, then Topics B, C, D, E, F, G, and H are viable topic results since they are found in common documents along with Topic A. Notice that even though Document 4 does not contain Topic A, it is on a path from Topic B or Topic D. So picking Topic B and then Topic D would lead to the display of Document 4 as a relevant search result.

Topic D has two implied parents: Topics A and B. This means coverage in the topic selection process is extensive because there are multiple paths to relevant results. Taxonomies do not have this networking property. There is only one parent for each child in taxonomy.

Search Coverage

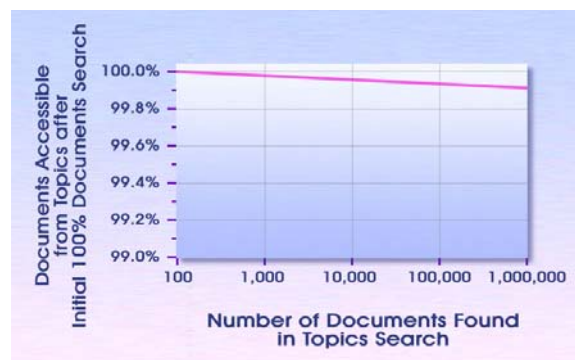
When using categorization methodologies, selecting a category produces a result that contains 100% of all documents assigned to that category. If users understand what is included in that category then they will obtain all relevant documents. However, as previously discussed, categorized searches of large numbers of documents will yield large numbers of results that are impractical to review.

In comparison, Topification dynamically calculates significant topics based on

user search input. If the search is specific (i.e., contains many words or infrequently used words), then the implied hierarchy of topics will reference all documents. However, it is more common that queries are under-defined (yielding 100,000s of results) along with the potential number of topics growing substantially.

Pliant's Topification methodology has two options for dealing with this situation. One method is to give the user the most statistically significant topic list with an option to pick "other" topics not represented in the current list. This provides 100% coverage. A further simplification is to bypass statistically less significant topics. Figure 5 represents the expected outcome using this last option. Typically, less than 0.1% of potential results are excluded by the initial topic set generated by the system. This last option is usually acceptable for grossly under-defined searches.

Figure 5: Topification Follow-On Coverage of (Initially 100% Searched) Documents





CONCLUSION

Topification searches offer considerable advantages over all current classification search techniques. These include:

- Auto-classification of documents.
- No resources needed to implement or maintain classification.
- Simultaneously manages multiple domains.
- Produces a small number of the most relevant results with just a few mouse clicks.
- Requires minimal computer resources.
- Not limited to 30,000 categories and can search tens of millions of documents.
- Focused on topics that are less ambiguous than categories by an order of magnitude.
- Evaluates a document from more than one perspective.
- Superior at evaluating relevancy of large documents.
- Zero-latency architecture.
- Systematically excludes irrelevant domains.

- Focuses on 2nd order, rather than 3rd order concepts.
- Unaffected by scale.
- Allows optional addition of taxonomies and topics.

In conclusion, Topification not only performs far better than classification methodologies, but it is also far less expensive to implement and maintain.

ABOUT PLIANT

Pliant Technologies, Inc. delivers Enterprise Artificial Intelligence (AI) software solutions that solve its client's greatest information challenges. Our patented SourceWare[®] infrastructure and Intent-Based Information Management[™] methods work with unstructured information "as is" and in real time. Our systems rapidly evolve the way companies do.

Copyright © 2002 Pliant Technologies, Inc.

All Rights Reserved.

Pliant, SourceWare and Intent-Based Information Management are Trademarks of Pliant Technologies, Inc.

July 25, 2002

