Relativity®

White Paper

# Document Categorization Using Latent Semantic Indexing

Anthony Zukas and Robert J. Price

**NOTE:** The following white paper was originally published in 2003. The foreword and epilogue are new, while all other content has been updated to remain current.

Content Analyst®

Find | Organize | Discover®

# Contents

# Abstract

The purpose of this research is to develop systems that can reliably categorize documents using latent semantic indexing (LSI) technology.[1] Research shows that LSI technology can effectively construct categorization systems that require minimal setup and training. Categorization systems based on LSI technology do not rely on auxiliary structures (thesauri, dictionaries, etc.) and are independent of the native language being categorized (given the documents can be represented in the Unicode character set).

Three factors led us to undertake an assessment of LSI for categorization applications. First, LSI has been shown to provide superior performance to other information retrieval techniques in a number of controlled tests.[2] Second, a number of experiments have demonstrated a remarkable similarity between LSI and the fundamental aspects of the human processing of language.[3] Third, LSI is immune to the nuances of the language being categorized, thereby facilitating the rapid construction of multilingual categorization systems.

Big Data—the volume, velocity, and variety of complex digital information—has pushed traditional approaches and analytics systems to be revisited, as they can no longer keep up with the flow. The emphasis here is on reducing the overall review burden, or at least organizing the information better, as more systematic and prioritized review is necessary. This has raised the question as to whether or not advanced analytics methods that can filter, cluster, categorize, and retrieve unstructured information that's relevant to the end user can be implemented to address the problem, and if they are easy enough to use within different production workflows.

We will describe the implementation of two successfully deployed systems employing the LSI technology for information filtering (English and Spanish language documents) and document categorization (Arabic language documents). The systems use in-house developed tools for constructing and publishing LSI categorization spaces. Various interfaces (e.g., SOAP-based web service, workflow interfaces, etc.) have been developed that allow the LSI categorization capability to address a variety of customer system configurations. The core LSI technology has been deployed on a variety of platforms and operating systems. In this paper, we will also describe some early results on the accuracy and use of the systems.

*"Big Data—the volume, velocity, and variety of complex digital information—has pushed traditional approaches and analytics systems to be revisited, as they can no longer keep up with the flow."*

1 S. Deerwester et al. Indexing by Latent Semantic Analysis, Journal of the Society for Information Science, 41(6), pp. 391-407, October, 1990.

2 S. Dumais. Using LSI for Information Retrieval, Information Filtering, and Other Things, Cognitive Technology Workshop, April 4-5, 1997.

3 T. Landauer and D. Lanham. Learning Human-like Knowledge by Singular Value Decomposition: a Progress Report, Advances in Neural Information Processing Systems 10, Cambridge: MIT Press, pp. 45-51, 1998.

# Foreword

As many of you know, Content Analyst's latent semantic indexing (LSI) technology is the engine that powers Relativity Analytics. With Analytics, Relativity users can search and organize documents based on concepts, allowing them to find key documents faster or quickly review large sets of similar documents. It's also the foundation for Relativity Assisted Review, in which Analytics leverages the expertise of humans to suggest coding decisions on all documents in a universe, basing its decisions off of a seed set of documents selected and coded by expert reviewers and lawyers.

When we decided to fill Relativity's analytics toolbox, we wanted a powerful engine that could get the job done for our users and that could be integrated fully into our software. There were, and still are, a number of different options out there, many of which were built on stable algorithms that had been around for years.

In the end—January of 2008—we chose Content Analyst's LSI engine, as we saw a theme with it that resonated with us: flexibility. Specifically, here's what stood out:

1. LSI technology is language independent. Users don't need to rely on vendors or installations for specific language packs. LSI only considers the data it's given for categorization, so any text is digestible.
2. Content Analyst's technology has a lot of adjustable knobs. That meant we could incorporate and build out some really fast, impactful Analytics features.
3. Index building for this engine is uniquely customizable. Each Analytics project is different, and only as good as the workflow that moves it. Quality indexes are key for strong results.
4. It's really good. As you'll see in this white paper, this technology has been battle-tested and stable for years—providing some great functionality that's comparable or more powerful than other available engines.

All of that translates into more control and a better experience for our end users. Add in the fact that Content Analyst's people are great to work with, and you have a solid integration.

*"When we decided to fill Relativity's analytics toolbox, we wanted a powerful engine that could get the job done for our users and that could be integrated fully into our software."*

What's new—and what's continually evolving—is how useful and consumable this engine is on the front end. With Analytics, we're pushing ourselves to make this stuff even easier to use. That means fewer clicks, smoother workflows, and more automated processes.

Still, we see a lot of value in giving users the opportunity to take a peek under the hood. This white paper focuses on the strength of the LSI engine itself, how it works, and why it's effective. This emphasis on the technical side of Analytics can give valuable insight into not only the engine's accuracy, but also what makes it different from the rest.

My hope is that, by understanding those details from an end user perspective, the processes and results you see in your own projects will become more transparent.

**Andrew Sieja,** *president and CEO*
kCura

# How LSI Works

LSI is an automated technique for the processing of textual material. It provides state-of-the-art capabilities for:

- automatic document categorization;
- conceptual information retrieval, and;
- cross-lingual information retrieval.

A key feature of LSI is that it is capable of automatically extracting the conceptual content of text items. With knowledge of their content, these items then can be treated in an intelligent manner. For example, documents can be routed to individuals based on their job responsibilities. Similarly, emails can be filtered accurately. Information retrieval operations can be carried out based on the conceptual content of documents, not on the specific words that they contain. This is very useful when dealing with technical documents, particularly cross-disciplinary material.

LSI is not restricted to working with words; it can process arbitrary character strings. For example, tests with MEDLINE data have shown that it deals effectively with chemical names. Points in an LSI space can represent any object that can be expressed in terms of text. LSI has been used with great success in representing user interests and the expertise of individuals. As a result, it has been employed in applications as diverse as capturing customer preferences and assigning reviewers at technical conferences.

In cross-lingual applications, training documents from one language can be used to categorize documents in another language (for languages where a suitable parallel corpus exists).[4]

## Training

Text categorization is the assignment of natural language texts to one or more predefined categories based on their content.[5] Text categorization systems run the gamut from those that employ trained professionals to categorize new items to those that are based on natural language clustering algorithms, which require no human intervention.

Supervised text categorization has a learning (or training) component where pre-defined category labels are manually assigned to a set of documents that will become the basis for subsequent automated categorization. Text categorization systems performing unsupervised training (or learning) automatically detect clusters or other common themes in the data that identify topics or labels without manual labeling of the data.

When used in text categorization applications, LSI requires a labeled training set of documents. Labeled training sets can be as few as 75 to several thousand in number. It is possible to use a small number of labeled documents to bootstrap the supervised learning process. After building an initial index with labeled test documents, additional documents can be submitted as queries, and query documents close in similarity to labeled documents in the index (within some pre-specified threshold value) can then be associated with the same label. In this manner, the labeled test set can be grown over time with a significant reduction in the human effort required to build a large labeled test set.

When using smaller-sized training sets (less than 300 to 600 documents), LSI may require some additional tuning of the dimensionality of the categorization index to capture the higher-ranked latent features in the training set. This is easily accomplished through a graphical user interface and iterations through re-indexing of the training set.

We have also found that adding unlabeled data ("background" text) in the presence of small labeled test sets improves the latent structure of the categorization

4 S. Dumais et al. Automatic Cross-linguistic Information Retrieval using Latent Semantic Indexing, in SIGIR'96 - Workshop on Cross-Linguistic Information Retrieval, pp. 16-23, August 1996.

5 S. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization, Proceedings of ACMCIKM' 98, 1998.

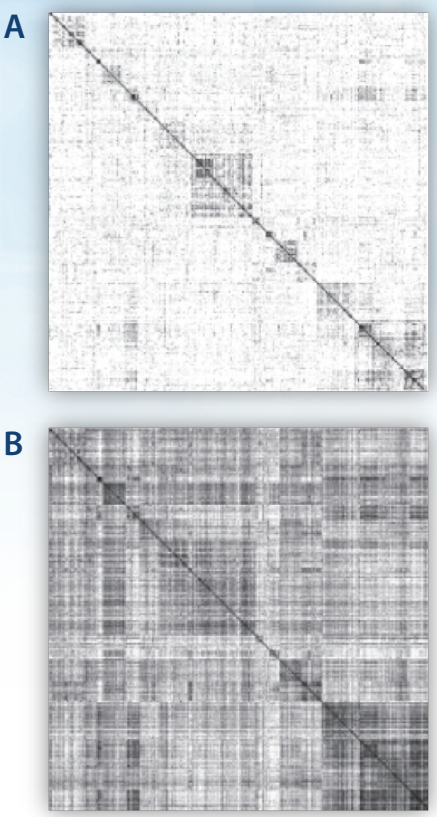


**Figure 1:** Graphics of the Training Space
**Figure 1A** shows the similarity matrix for the training set. The row and column axes represent the documents in the training set; the diagonal shows that every document is related to itself. The stronger outlines surrounding the diagonal represent the labeled classes within the training data.
**Figure 1B** shows how background material strengthens the latent relationships in the training data.

index leading to improved accuracy. Similar results have been reported in the literature.[6,7] **Figure 1** shows the effect of background material on a small labeled test set of 300 documents. Unlabeled examples (e.g., web pages, emails, news stories) are much easier to locate and collect than labeled examples.

A common critique of LSI in the literature is the relatively high computational and memory requirements required by LSI to function.[8] However, with the ever-increasing speeds of modern processors, this former consideration has been overcome. Training LSI with a moderate training set can be accomplished in a matter of minutes on current corporate desktop PCs with less than 1 GB of memory. Larger sets of training documents require less than 10 minutes on equivalent PCs.

## Test Corpus and Performance Measures

LSI as a text categorization engine has been deployed in a number of real-world applications as described later in this paper. To compare its performance to other published results, we used the ModApte version of the Reuters-21578 test set. The ModApte version has been used in a wide number of studies[9] due to the fact that unlabeled documents have been eliminated and categories have at least one document in the training set and the test set. We followed the ModApte split defined in the Reuters-21578 data set in which 71 percent of the articles (6,552 articles) are used as labeled training documents and 29 percent of the articles (2,581 articles) are used to test the accuracy of category assignments.

Many different evaluation criteria have been used for evaluating the performance of categorization systems. For evaluating the effectiveness of category assignments to documents by LSI, we adopted the breakeven point (the arithmetic average of precision and recall) as reported in [5] and [9], and the total (micro-averaged) precision P and recall R.[10] The micro-averaged breakeven point is defined as (P+R)/2.

**6** K. Nigam. Using unlabeled data to improve text classification, PhD. Thesis, Carnegie Mellon University, May 2001.

**7** S. Zelikovitz and H. Hirsh. Using LSI for Text Classification in the Presence of Background Text, Proceeding of CIKM-01, 10th ACM International Conference on Information and Knowledge Management, 2001.

**8** G. Karypis and E. Han. Fast supervised dimensionality reduction algorithm with applications to document categorization and retrieval, Proceedings of CIKM-00, 9th ACM International Conference on Information and Knowledge Management, 2000.

**9** Y. Yang and X. Liu. A re-examination of text categorization methods, Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99), pp. 42-49, 1999.

**10** Y. Yang. An evaluation of statistical approaches to text categorization, Journal of Information Retrieval, Volume 1, No. 1/2, pp. 67-88, 1999.

## Comparison with Other Techniques

Table 1 summarizes the global performance score for LSI along with the best performing classifier from [9]. As can be seen from Table 1, the LSI miF1 value was competitive with the miF1 value for the support vector machine (SVM) in [9].

**Table 1**

| Method | miR | miP | miF1 | maF1 | error |
|---|---|---|---|---|---|
| LSI | .8880 | .8900 | .8890 | .5880 | .0040 |
| SVM | .8120 | .9137 | .8599 | .5251 | .0036 |

miR = micro-avg recall  miP = micro-avg prec
miF1 = micro-avg F1  maF1 = macro-avg F1

While the document counts between the two studies were not exactly the same, the overall ratios of training set to test set were almost exactly the same; in [9] the ratios were 72 and 28 percent for the training and test sets, respectively. Additionally, in [9] there was an assumption that documents could fit into more than one category; unlabeled documents were eliminated and categories had to have at least one document in a training set and the test set. In [9], the number of categories per document was 1.3, on average. The category per document ratio for the ModApte data set used in this paper was 1. This is a more stringent restriction on text categorization classifiers. The LSI results reported in Table 1 reflect this constraint.

In [5], the assumptions concerning what documents made up the ModApte split differ slightly from [9] and the test set used in this study. The mean number of categories per document for [5] was 1.2, but many documents were not assigned to any of the 118 categories, and some documents were assigned to three or more categories. The SVM was the most accurate text categorization method in [5] with an overall miF1 rating of 0.8700, placing it between the LSI and SVM results reported in Table 1.

# Real-World Applications and Lessons

Two real-world applications of LSI follow to demonstrate the use of LSI as a text categorization engine in true cases.

## Information Filtering/Knowledge Discovery

In this application, the customer had a proprietary process for collecting English and Spanish content on a periodic basis. Once collected, the content was indexed with Boolean retrieval technology and made available to analysts for review. Analysts constructed and executed queries to retrieve content specific to their particular interests. Results varied depending on the expertise analysts possessed in constructing queries. An additional drawback was that analysts spent a large amount of their time searching for relevant content rather than analyzing content.

To address the above situation, LSI was integrated into the workflow to replace the Boolean retrieval technology. Rather than construct and execute queries, analysts supplied representative content (i.e., documents) relevant to their areas of interest. This material was tagged and indexed. Content collected on a periodic basis was compared to the index of analyst-relevant content. Content similar in nature (within a specified threshold) to analyst content was routed to the appropriate analyst. Restructuring of the workflow in this manner resulted in a continuous push of relevant content to analysts, resulting in a significant increase in productivity on the part of the analyst. This system has been in production for more than two years.

## Document Categorization/Prioritization

In this application, the customer had a high volume of Arabic language content and an insufficient number of Arabic-qualified analysts to review all the content. In order to ensure that relevant content was not overlooked, all of the material had to be examined—leading to overworked analysts and a situation where, potentially, some item of important material might be overlooked.

To address the above situation, a training set of Arabic content was constructed and labeled according to customer-defined categories. The system was trained with the labeled training set. An additional 20,000 relevant Arabic documents were selected and used as background training material. Integration with the customer's workflow was accomplished using a SOAP-based web service. In the new system, Arabic documents for categorization were passed to the web service. A ranked list of categories and associated similarity scores were sent back to the client process. Based on customer-defined rule sets, the client process made decisions about the importance of the documents and their disposition. Highly ranked documents were immediately forwarded to analysts, less important documents were stored for later examination during periods of analyst workload, and uninteresting documents were discarded. During customer acceptance testing, this system demonstrated 97 percent accurate assignment of Arabic-language documents to individual categories. This result was measured using real-world documents with significant quantities of noise.

## Lessons Learned

The LSI technology has matured to the point where it is a particularly attractive approach for text categorization. Text categorization results with LSI are competitive, on comparable test sets, with the best results reported in the literature. A definite advantage to the LSI text categorization technology is native support for international languages.

LSI categorization can perform well with very limited quantities of training data, generally with only a few examples per category. This is due, in great part, to the exceptional conceptual generalization capabilities of LSI.

User feedback can be incorporated to continually improve performance. The LSI technique has a significant degree of inherent noise immunity with regard to errors in the documents being processed. Documents can be assigned to multiple categories, with reliable indications of the degree of similarity to each category.

# Epilogue

As this study in the engine embedded in Relativity Assisted Review shows, an apples-to-apples comparison yields negligible differences between the strength of LSI and SVM technologies in categorizing documents. In the realm of computer-assisted review, that means there is more to an effective assisted review process than choosing a "black box" to run behind the scenes.

That's because, regardless of the technology, there is still no substitute for the human brain when it comes to making deep, subjective connections between complex topics. Humans need to train the technology and use that deep understanding to validate the results. The best computer-assisted review processes strike a balance between the cognitive strength of domain experts and the raw horsepower of the technology.

An effective computer-assisted review process should consider the unique needs of each case, the composition of the review team, and the desired ease of use for the review team's administrators. Other factors to consider when selecting a computer-assisted review system may include:

1. **Search Capabilities** — Does the system allow for keyword searching to find strong examples to help seed the analytics engine?
2. **Review Workflow** — Does the system make it easy for reviewers to navigate between documents?
3. **Volume** — Can the system handle a substantial amount of documents?
4. **True Time Cost** — What is the combined time to load documents into the system, categorize the documents, train the system, and move to the production phase?

A solid computer-assisted review is a microcosm of the review process, relying on a combination of variables to churn out the results used by the review team. In addition to a strong engine, having a repeatable, defensible, and sound workflow is equally important. That combination—engine, validation workflow, and domain expertise—will ensure that your results are transparent and trustworthy. kCura's white paper on the process behind Relativity Assisted Review provides a strong overview of an example workflow that combines human expertise and statistical validation with the categorization engine. In addition, Dr. Grossman's white paper on validating the Assisted Review workflow demonstrates the effectiveness of joining process with technology.

> *"The best computer-assisted review processes strike a balance between the cognitive strength of domain experts and the raw horsepower of the technology."*

Every case is different, and each project will have different needs, so flexibility in your process is important. The technology should enable the strategy of the review team, not be a limiting factor—picking your technology should be part of the tactics, not the strategy.

**Jay Leib,** *chief strategy officer*
kCura

## About the Authors

**Robert Price** is the principal engineer at Content Analyst. He has more than 24 years of software development experience, with the last 11 years focused on making LSI and other text analytics tools more scalable, accessible, usable, and better performing to address real-world problems with Big Data. In this role, he has been the primary architect and algorithm developer for Content Analyst's CAAT product. Robert received an M.S. in computer science from the University of Illinois at Urbana-Champaign. He is the author of two patents related to LSI.

**Anthony Zukas** is a senior software scientist with Agilex. His research interests include distributed computing, artificial intelligence algorithms, and natural language processing and understanding. He has more than 12 years of experience integrating LSI into solutions for commercial and government clients. Anthony holds M.S. degrees in computer science from George Washington University, as well as in software systems engineering and bioinformatics from George Mason University. He is a member of IEEE, ACM, and AAAS. Agilex is a Content Analyst partner, actively involved in integrating Content Analyst into systems for the intelligence community.

### ACKNOWLEDGEMENTS

The authors wish to thank **Roger Bradford, Janusz Wnek,** and **Rudy Keiser** for their useful comments when reviewing the paper.

kCura®

231 South LaSalle Street, 8th Floor, Chicago, IL 60604
T: 312.263.1177 • F: 312.263.4351
info@kcura.com • www.kcura.com

Content Analyst®
Find | Organize | Discover®

11720 Sunrise Valley Drive, Reston, VA 20191
T: 888.349.9442
rprice@contentanalyst.com • www.contentanalyst.com