# HARVEST: An Intelligent Visual Analytic Tool for the Masses

**David Gotz, Zhen When, Jie Lu, Peter Kissa**
IBM T.J. Watson Research Center
19 Skyline Dr, Hawthorne, NY, USA

**Nan Cao, Wei Hong Qian, Shi Xia Liu, Michelle X. Zhou**
IBM China Research Lab
8 Dongbeiwang West Rd, Beijing, China

**ABSTRACT**
We present an intelligent visual analytic system called HARVEST. It combines three key technologies to support a complex, exploratory visual analytic process for non-experts: (1) a set of smart visual analytic widgets, (2) a visualization recommendation engine, and (3) an insight provenance mechanism. Study results show that HARVEST helped users analyze a corpus of text documents from a corporate wiki.

**Author Keywords**
Visual Analytics, Smart Graphics, Visualization

**ACM Classification Keywords**
Algorithms, Human Factors

**INTRODUCTION AND RELATED WORK**
In recent years, a large number of visualization systems have been developed to help users view, explore, and analyze information. The capabilities supported by these visualization systems vary broadly, ranging from supporting casual visual collaborations (e.g., ManyEyes [11] and Swivel [1]) to commercial-grade visual analytics (e.g., Spotfire [3] and Tableau [2]).

At the same time, businesses have been creating and storing more data than ever before. Recognizing that valuable insights are buried within these mountains of information, companies have begun to push the use of visualization to all their employees to aid their business decision-making processes. However, most of today's visualization tools still target two niche audiences: (1) dedicated information analysts and (2) dashboard consumers.

Tools for information analysts cater to users who have acquired a high degree of visualization and computer skills and often use sophisticated visualization software. However, they are typically too complex for average business users. In contrast, dashboard consumers are typically casual users of visualization. By design, dashboard systems require far less skill and are accessible to a much wider range of users. However, they lack several key capabilities, such as continuous exploration of large data sets, which are often required to support real-world business tasks.

However, there is a third and perhaps largest class of users for whom existing tools are of limited value: everyday business workers. These users typically have extensive domain knowledge but are not visualization or computer experts. Yet as part of their daily responsibilities, they perform situational analysis tasks over massive amounts of data for which visualization can be of great benefit.

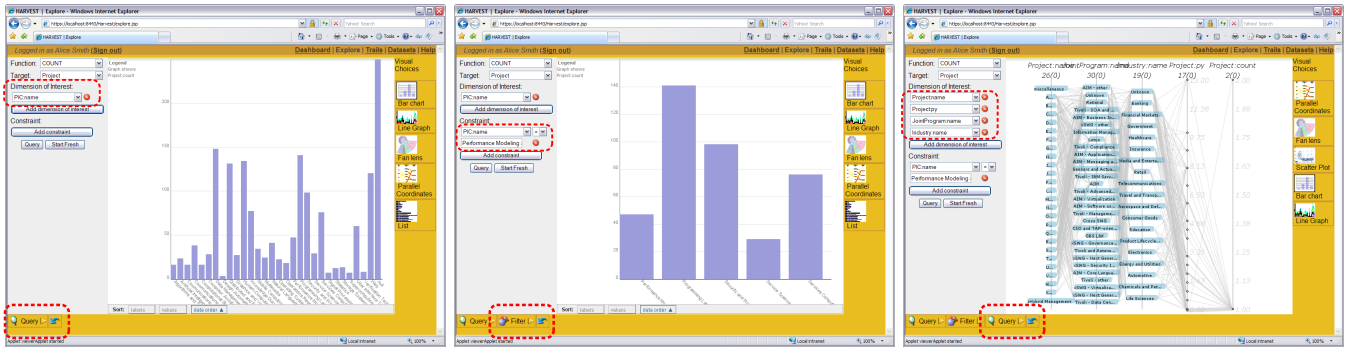
For example, in our own company, employees often examine a large wiki site containing data about numerous projects underway within our organization. While the wiki effectively provides information on individual projects, it is very difficult for users to examine project patterns or trends. Neither can most existing visualization tools make this sort of task any easier for an average person.

To help this user population, we are building HARVEST, an intelligent visual analytic system for everyday business users. HARVEST combines three key technologies to support an exploratory visual analytic process without requiring users to be visualization or computer experts:

- **Smart visual analytic widgets.** A set of visualization widgets that can be easily reused across applications. They support semantics-based user interaction to help identify and capture user intention, and incrementally handle dynamic data sets retrieved during a continuous visual analytic task.
- **Dynamic visualization recommendation.** A context-driven approach that assists users in finding the proper visualizations for use in their context.
- **Semantics-based capture of insight provenance.** A semantics-based approach to modeling and capturing a user's logical analytic process. It supports automatic detection of user action patterns for better visualization recommendation, and enables flexible adaptation of a user's analytic process for reuse in new contexts.

Our work is related to previous systems that use automatic visualization (e.g., [9]) but focuses on situational analytics where role or template based approaches are less effective. Our work is also related to systems that capture user histories (e.g., [4, 8, 10]). However, HARVEST focuses on the extraction of a semantic representation of a user's insight provenance that is independent of application and across a range of visualization tools. It then analyzes that provenance to provide context-relevant visualization recommendations.

**REFERENCE APPLICATION**
Our work on HARVEST is motivated by the common information needs of employees within our own company. Our organization maintains a large wiki site describing all ongoing research projects. Each project page is a semi-structured



(a) (b) (c)

**Figure 1. Screenshots illustrating the typical user workflow in our reference HARVEST application. In addition to the changing visualization canvas, user progress is reflected in both the query panel and history panel as we highlight with in red.**

text document, containing a project description, the people involved, and several other important pieces of information. New projects are added to the wiki regularly, and updates are constantly contributed by project members and managers. While it is relatively easy to look up information about individual projects in the wiki, there is no easy way to obtain an overview of a collection of projects. Yet higher-level summaries of information may often be most valuable.

For example, consider a researcher named Alice who is putting together a new proposal for a computer vision research project. To scope her project properly, Alice must decide how many 'person-years' (PYs) could be realistically funded. To help answer this question, Alice would like to view the distribution of PYs in funded projects, especially in the area of computer vision. Similarly, Alice could better position her proposal if she could discover which funding programs were historically most likely to accept computer vision proposals. In addition, she would like to identify potential collaboration partners by examining related projects and their teams. The information required to answer each of Alice's questions is contained within the project wiki. However, there is no easy way for Alice to extract the needed insights. HARVEST is designed to help people like Alice by providing a set of intelligent visual analysis tools.

Before HARVEST can be used for this application, the wiki data was pre-processed by a text-analysis tool to extract key terms and concepts. The extracted data was then stored together with the documents' structured meta-data within a DB2 database. Then, Alice begins by logging in to HARVEST and initiating a new task. She starts by using the query GUI panel to build a query to summarize the number of projects by discipline. In response, a *Query* action is processed by the three core HARVEST components: (1) the *query manager* interprets the GUI input to formulate a SQL query and executes it, (2) the *visualization recommender* automatically composes a bar chart encoding the retrieved data, and (3) the *action tracker* incorporates the *Query* action into its representation of Alice's insight provenance. The visualization and the newly performed *Query* action are displayed in Figure 1(a).

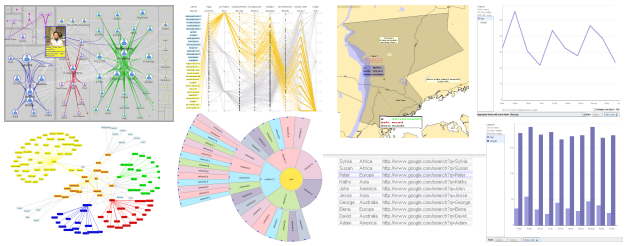


**Figure 2. Samples of visual analytic widgets used in HARVEST.**

Alice then selects a subset of five bars from the visualization that correspond to five disciplines in which she is interested. She issues a *Filter* action using the bar chart's context-sensitive menu. In response, HARVEST updates the visualization to reflect Alice's new data interests. Both the query and history panels also are updated to include the new data constraints and the Filter action, respectively (Figure 1b).

For all projects in the five selected disciplines, Alice now wants to examine the correlations among four variables: the discipline, funding partner, project PY, and related industry. To do so, Alice modifies the current query and submits it. In response to this new *Query* action, HARVEST creates a parallel coordinates visualization to encode the updated data (Figure 1c). After identifying an important trend, Alice switches to a list view of the documents and selects individual items to view the full text. As shown here, Alice's analysis goals and data interests evolve over the course of her task, making it impossible to know ahead of time which data sets Alice would like to analyze or the proper visualizations to use.

## KEY HARVEST TECHNOLOGIES

HARVEST combines three key technologies: (1) smart visual analytic widgets (Figure 2), (2) dynamic visualization recommendation, and (3) semantics-based capture of insight provenance.

### Smart Visual Analytic Widgets

HARVEST's smart visual analytic widgets support incremental visual updates to accommodate users' evolving data interests due to the exploratory nature of their tasks. Few existing visualization tools support incremental updates. In-

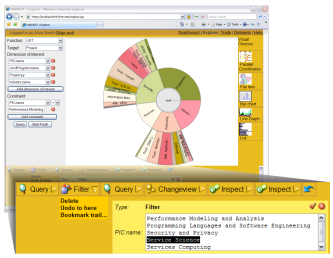


**Figure 3. The history panel displays the unfolding analytic trail.**

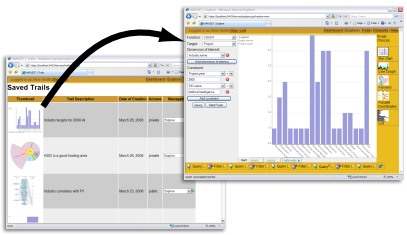


**Figure 4. Users can restore saved trails to reuse past analyses.**

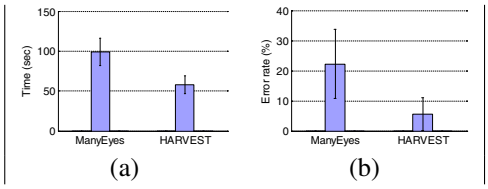



**Figure 5. Mean and 95% confidence interval of (a) task completion time and (b) task error rate.**

stead, a new visualization must be created if the underlying data changes. However, the abrupt change when creating a new visualization disrupts visual continuity and reduces a user's ability to comprehend information across successive displays [12]. To address this issue, a subset of our visual widgets is designed to support incremental visual updates. They include a visual context management module to dynamically decide how to best update the existing visualization to incorporate new data [12]. For example, When a user issues a follow-up query to retrieve additional data, our SmartMap widget dynamically derives a set of visual animation operators to incrementally update the existing visualization, such as *CameraSwitch*, *Add* (adding visual representations of the new data), and *Simplify* (visually simplifying visual representations of old data).

In addition, each widget supports a set of semantics-based user actions to capture the semantics of a user's insight provenance. One of HARVEST's key goals is to capture the semantics of insight provenance, which can be used to help share and re-purpose a user's visual analytic processes. Since a large part of a user's activity is interacting with visual widgets, ideally these widgets should recognize the semantics of user activities as they occur. This is in contrast to most existing visualization tools that support an event-based interaction model (e.g., clicks and drags) and know little about the semantics of a user activity. To achieve our goal, we implement visual analytic widgets to support a set of actions, which are semantics-based interaction primitives (see Section *Semantics-Based Capture of Insight Provenance*).

### Dynamic Visualization Recommendation

As demonstrated in our reference scenario, it is impossible to determine ahead of time which visualization tools should be used. To assist average users in effectively using visualizations in their tasks, we develop a visualization recommendation engine. Given a user's request, our engine automatically recommends the top-N suitable visualizations to the user. This engine extends our previous effort in automated visualization generation [13], which was limited to handling small data sets in a relatively static environment. To support continuous user interaction with large data sets in real-world HARVEST applications, we have extended our work to consider user behavior.

During visual analysis, a user's behavior can often signals implicit analytic needs [7]. For example, assume that Alice is interacting with a FanLens that hierarchically encodes the number of projects by discipline and by sponsor (Figure 3). To compare the number of projects by sponsor in each of the disciplines, she iteratively clicks on each discipline (a slice) to expand it. To better help Alice work more effectively in the above situation, HARVEST provides behavior-based visualization recommendation [5]. First, as a user interacts with HARVEST, the action tracker component examines the user's action history in search of meaningful patterns (see the *Automatic Pattern Detection in User Actions* section). Once a pattern is detected, we use a rule-based approach to map the pattern to an implied visual task. For example, the pattern demonstrated by Alice is mapped to a visual comparison task. Based on the inferred visual task, our visualization recommendation engine would recommend a bar chart visualization to Alice that more effectively encodes the desired information for direct comparison.

### Semantics-Based Capture of Insight Provenance

Visual analytic tasks are often complex and time consuming. To make the process easier for average business users, HARVEST's action tracker component maintains a semantics-based model of a user's visual analytic activity. This model is then used to enable more effective visualization recommendation, and to allow flexible adaptation of a user's analytic process to new situations. We refer to the model of user activity as insight provenance because it contains the history and rationale of how insights are derived during a user's visual analytic process.

#### *Automatic Identification of User Analytic Trails*

Our empirical studies [7] demonstrate that distinct logical sequences of user actions leading to an insight, which we call analytic trails, can be observed in a user's analysis process. Using our reference application scenario, Alice has performed three actions, *Query* $\Rightarrow$ *Filter* $\Rightarrow$ *Query* to reach the state shown in Figure 2(c). As this example illustrates, trails define a user's exploration path and its semantics (e.g., captured by the action types and parameters).

HARVEST actively analyzes the linear sequence of user actions in the order they are performed by an analyst. Based on the type of action being performed, HARVEST builds a graph-based representation of interconnected trails to represent the user's visual exploration behavior. When users save their work via *Bookmark*, HARVEST preserves both the state of the visualization as well as the automatically recorded analytic trail. When a bookmark is later restored, the trail is restored as well. This allows a user to review the exploration

context in which an insight was discovered. This feature is especially useful during collaborative tasks, allowing users to see not only *what* a coworker has found, but also *how* they found it.

#### *Automatic Pattern Detection in User Actions*

In addition to identifying analytic trails, the action tracker performs pattern detection over a user's recently performed actions in search of meaningful activity patterns. We define a pattern as an iterative user behavior that implies a specific analytic goal. Studies show that patterns occur frequently in typical visual analytic behavior and correlate with real or perceived limitations in a tool [7]. Detected patterns are passed as input to the visual recommender to enable user behavior-driven recommendations. HARVEST uses a rule-based approach to pattern detection. Each time a user performs a new action, the user's analytic trail is compared against a library of pattern rules. The library includes one or more rules for each pattern recognized by HARVEST. The system currently detects four user action patterns: *Scan*, *Flip*, *DrillDown*, and *Swap* [5].

#### *Flexible Adaptation of Analytic Trails*

One of the main benefits of HARVEST's automatic capture of analytic trails is that it allows users to adapt their previous analysis processes to new tasks. As a user interacts with the system, HARVEST externalizes a simplified version of the user's exploration path in the history panel (Figure 3). Users can interact with the history panel to directly manipulate their trail. Supported manipulations include: *Undo*, *Revisit*, *Delete*, and *Modify*. This allows users to quickly adapt their previously performed trails to new contexts. User's can modify specific parameters of an action (e.g., change a *Filter* from $Year = 2008$ to $Year = 2006$). Whenever the user's analytic trail is altered, HARVEST automatically requeries for new data and composes an updated visualization. These capabilities are especially powerful when combined with bookmarks. Rather than always starting from scratch, a user can make use of previously saved trails from similar tasks. After restoring a saved trail, a user can back up to any action in the restored trail to use as a starting point for his/her new analysis. Alternatively, s/he can re-use the entire trail and simply modify individual action parameters to meet the new needs.

## EVALUATION

We applied HARVEST to the reference application described earlier and conducted a user study with eight participants, using a modified version of ManyEyes [11] as a baseline. Figure 5 shows that HARVEST performed significantly better by our two objective metrics: task completion time and error rate. Users of HARVEST were able to complete their tasks significantly faster ($p < 0.0001$), with about $40\%$ time reduction at each step of the task on average (Figure 5a). Note that we ignored the time spent retrieving, formatting, and uploading data as required by ManyEyes. Thus, the time accounted for was spent by a user to select a visualization, interact with the visualization to analyze the information, and switch to a new visualization if needed. We attribute this significant reduction in time mainly to HARVEST's visualization recommendation, which quickly led users to proper visualizations for their tasks.

Our results also indicated a significant difference in task error rate between the two systems ($p < 0.01$) (Figure 5b). When we checked user results with facts from the original content, we found that there was a $75\%$ reduction in error rate on average when a task was performed using HARVEST ($5.6\%$) vs. using ManyEyes ($22\%$). We attribute the sharp drop in error rate to HARVEST's ability to let users easily explore data from different angles. Users commented that HARVEST made it "*easy to switch*" to alternative visualizations and different data sets. Moreover, HARVEST's automated analytic trail management facility made operations like "go back" or "undo" trivial. In essence, it was the seamless integration of HARVEST's key technologies that led to more accurate results. As one user commented, "*[there was] coordination among [the] query GUI, analytic trail, and visualization [in HARVEST]...*", where you could "*modify/specify queries from any of the three.*" Finally, from users' subjective feedback, the participants also overwhelmingly favored HARVEST (mean rating of 4 out of 5) over ManyEyes (mean rating of 2.6) for the tasks that they performed.

## CONCLUSION

In this paper, we have presented HARVEST, an intelligent visual analytic system designed to empower everyday business users to derive insight from large amounts of data. We reviewed the key technologies behind the HARVEST system and presented results from a user study. Our study shows that HARVEST technologies were preferred by users, and that they helped users perform significantly better by two objective metrics: task completion time and error rate.

## REFERENCES


1. Swivel. http://www.swivel.com/.
2. Tableau software. http://www.tableausoftware.com/.
3. Tibco spotfire. http://spotfire.tibco.com/.
4. L. Bavoil and et al. Vistrails: Enabling interactive multiple-view visualizations. In *IEEE Vis*, 2005.
5. D. Gotz and Z. Wen. User behavior driven visualization recommendation. In *IUI 2009*.
6. D. Gotz and M. X. Zhou. Characterizing users' visual analytic activity for insight provenance. In *IEEE VAST*, 2008.
7. D. Gotz and M. X. Zhou. An empirical study of user interaction behavior during visual analysis. Technical Report RC24525, IBM Research, 2008.
8. M. Kreuseler, T. Nocke, and H. Schumann. A history mechanism for visual data mining. In *IEEE InfoVis*, 2004.
9. J. D. Mackinlay, P. Hanrahan, and C. Stolte. Show me: Automatic presentation for visual analysis. In *Proc. of IEEE InfoVis*, 2007.
10. Y. B. Shrinivasan and J. J. van Wijk. Supporting the analytical reasoning process in information visualization. In *CHI*, 2008.
11. F. B. Viegas, M. Wattenberg, F. van Ham, J. Kriss, and M. McKeon. Many eyes: A site for visualization at internet scale. In *InfoVis*, 2007.
12. Z. Wen, M. X. Zhou, and V. Aggarwal. An optimization-based approach to dynamic visual context management. In *InfoVis*, 2005.
13. M. X. Zhou and M. Chen. Automated generation of graphical sketches by example. In *Proceedings of IJCAI*, pages 65–74, 2003.