

Research Statement

Zhuowei Bao

My research interests focus on the development of scalable techniques for *big data analytics*. As a first step towards this goal, my dissertation research seeks to provide scalable solutions to track *data provenance* in large-scale *workflow executions*. Workflows have been widely used to model complex scientific experiments as well as data-intensive business processes. Such workflows tend to be repeatedly executed using different parameters or data inputs, generating a large set of intermediate and final data products. In recent years, a number of workflow systems, such as Taverna, VisTrails and Kepler, have been deployed to ease the process of designing, executing and managing workflows. An important capability provided by those systems is to automatically capture the provenance information of produced data. However, tracking data provenance can be prohibitively expensive, due not only to the massive amount of data that workflow executions produce, but also to the complex form of queries that one may ask. The goal of my dissertation is thus to enable efficient evaluation for a rich class of queries over big workflow provenance in a variety of settings.

Another line of my research lies at the intersection of natural language processing and information retrieval. With the growing scale and complexity of intranets, *enterprise search* has attracted increased attention from the information retrieval community. The ultimate goal of my research is to improve the relevance of documents returned by keyword queries in the context of enterprise search. Rather than proposing novel ranking algorithms, the approach that I pursue is to provide *query correction*, which captures the true intention behind the posted search query and therefore better matches the desired document. Logically, the task of query correction is divided into two layers: (1) *spelling correction* for handling various typos in misspelled queries; and (2) *query rewriting* that rewrites the posted query to adapt to the terminology used in relevant document matches. My approach combines the techniques from machine learning and graph theory, and entails big data analytics (e.g., extracting statistics from millions of documents). In this line of work, I collaborated with researchers from IBM Research – Almaden. All the techniques and tools that I have developed are intended for widespread use in the live intranet search engine running at IBM.

1 Query Evaluation for Workflow Provenance

In the early days of my research, I struggled with the question of “What is an appropriate model for workflows and their provenance?” First of all, graphs seem to be a right tool to describe the dependencies between data and modules involved in workflow executions. However, existing graph-based workflow models, such as Open Provenance Model, fail to capitalize on the tight connection between a workflow specification and its executions. Observe that workflow executions originate from a small specification, and can become much larger and structurally more complex than the specification, due to the repeated executions of sub-workflows, e.g., sequentially (*loops*), in parallel (*forks*) or through *recursion*. To formalize this behavior, I borrow the notion of *context-free graph grammar* from graph theory,

and use it to define an elegant workflow model [4]: every workflow specification is represented as a context-free graph grammar whose language denotes exactly the set of all possible executions of this specification. The design of my grammar-based model is also driven by the way people used to build workflows. Since building complex workflows from scratch takes enormous effort, a common practice is to re-use workflows, or portions of workflows, by creating composite modules that encapsulate shareable sub-workflows. This naturally leads to a grammar-based formalization of hierarchical workflows. Based on this model, my dissertation research focuses on the problem of efficiently evaluating provenance queries over workflow executions, derived from a given specification, in a variety of settings.

As a first step, I consider a simple form of provenance queries, called *reachability queries*, which serve as a primitive operation to examine the dependencies between data and modules. However, evaluating reachability queries using two straightforward approaches – traversing the graph or precomputing the transitive closure – can be prohibitively expensive for large provenance graphs. To address the problem, I rely on *reachability labeling*. The idea is to assign each vertex a label such that by comparing only the labels of any two vertices, one can quickly decide if one vertex can reach the other. In contrast to existing labeling techniques, my approach [3] is based on using the labeling for the (small) specification as an effective *skeleton* for designing the labeling for (large) executions. A somewhat surprising result of this work is that knowledge of the specification can be exploited to obtain *compact* and *efficient* labeling for executions, which provides *optimal* theoretical guarantees on the performance of query evaluation (i.e., using only logarithmic-size labels, building all labels in linear time, and answering any reachability query in constant time).

One limitation of the above labeling scheme is that it needs to examine the entire graph before labeling is performed. However, this may not be realistic in a workflow setting, since scientific workflows can take a long time to execute, and users may want to ask provenance queries over partial executions. To overcome this challenge, my follow-up work [5] presents a *dynamic* labeling scheme that allows modules and data to be labeled *on-the-fly* (i.e., as soon as they are produced during the execution). I also show that, in general, for workflows that contain recursion, dynamic labeling of executions requires long (linear-size) labels. Fortunately, most real-life scientific workflows are *linear recursive*, and for this natural class it turns out that dynamic, yet compact (logarithmic-size) labeling is possible.

There is also a need to use workflow *views* to provide useful answers to reachability queries while ensuring that privacy concerns are met [9], which introduces new challenges for query evaluation. Since workflow provenance may contain irrelevant details or private information that should not be revealed to certain users, a common way to secure workflow provenance is to authorize different groups of users to access different workflow views. Such views are defined over a specification and then projected onto its executions, allowing users to access a subset of provenance information, in the desired granularity. To efficiently answer reachability queries over views of an execution, my recent work [6] proposes a novel *view-adaptive* dynamic labeling approach whereby view specifications are labeled *statically* (i.e., as they are created), whereas data (or modules) are labeled *dynamically* as they are produced during a workflow execution. At query time, the labeling of the view over which the reachability query is asked is used to augment the data labels to provide correct reachability results in constant time. Moreover, I identify a large common class of *safe views* over *strictly linear-recursive* workflows for which compact dynamic labeling is possible.

A unifying theme of this line of research is to use the knowledge of the specification to design scalable skeleton-based labeling for large executions. As the final component of my dissertation work, I will explore the feasibility of extending these ideas to tackle more complex forms of queries (e.g., those which extend reachability queries with query constructs like path expression or graph pattern). In particular, I am studying *regular path queries* which ask if there exists a path connecting two given vertices such that the sequence of edge labels on this path matches the given regular expression. Surprisingly, my initial investigation shows that, in spite of the increased complexity of regular path queries, query evaluation can be done as efficiently as that for reachability queries, using skeleton-based labeling.

Throughout my research, I am always interested in understanding the nature of the problem from a theoretical perspective. For example, I study the upper and lower bounds of the label length for different classes of graphs in both static and dynamic settings, and characterize both the structure of workflows and the property of views based on the feasibility of developing compact dynamic labeling. However, I am also interested in their performance in practice. To this end, I implement all the labeling schemes, and examine their effectiveness and limitations through empirical evaluations over both real and synthetic datasets.

Other Work. My earlier research [1] studied the problem of computing the *difference* between two executions of the same specification, where the difference is defined by a notion of *edit distance*. Formally, the problem is to compute the minimum cost of a sequence of path edit operations (e.g., path insertion or deletion) that transform one execution to the other. Although this problem is NP-hard for general specifications, for a natural class of specifications modeled by series-parallel graphs overlaid with well-nested forks and loops, a polynomial-time algorithm for differencing their executions is presented. In collaboration with bioinformaticians, I also implemented a prototype system [2], called **Provenance Difference Viewer** (PDiffView), which allows users to examine differencing results by stepping through the sequence of path edit operations, at any desired granularity. In the other work [10, 11], I designed a query language for integrating streaming data from the Web and sensor devices, and implemented its parser using the Eclipse DataTools open source.

2 Query Correction for Enterprise Search

In the first layer of query correction, I consider spelling correction for keyword queries. While the target application is enterprise search, the proposed technique [7] is also effective for other domain-centric search applications, such as personal email search or desktop search. In contrast to Web search where query logs are abundant, in those applications query logs are too scarce to provide sufficient statistical information for spelling correction. Instead, my approach relies mainly on the raw corpus, which often contains various types of information with different levels of reliability. The key contribution of this work is a novel graph-based algorithm that can handle complex spelling errors like splitting and merging of words, and can incorporate various types of statistical information extracted from the raw corpus in a uniform fashion. The experimental study demonstrates the superiority of this spelling correction approach over existing alternatives in the closed domain of personal email search (Enron email datasets), and indicates its comparable performance to the state-of-the-art Google spelling correction in the open domain of large-scale site search (www.ibm.com).

In the second layer of query correction, I study query writing in the context of enterprise search. To resolve a complaint of a missing relevant document, it is a common practice for search administrators to manually encode query-rewrite rules which can push the desired document up to the top matches. The process of manually detecting and maintaining those rules is extremely tedious and time consuming. In this work [8], I seek to ease the burden of search administrators by providing automatic suggestions for query-rewrite rules. However, this automation faces two major challenges. The first is to select, among many candidates, those rules that are “natural” from a semantic perspective (e.g., corresponding to synonyms or to closely related concepts). This is a typical classification problem that can be addressed by machine learning techniques. The second is to tackle the interplay between a large set of rules maintained by the search engine (e.g., the new rules introduced by one query may eliminate desired results for other queries, or eliminate the desired effect of other rules). This work presents a simple formalization of this challenge as a generic computation problem. It is shown that, albeit its simplicity, this problem is highly intractable in terms of formal complexity theory. Nevertheless, heuristic solutions and optimizations are proposed and proved effective on the real dataset obtained from the IBM intranet search engine.

Even though these two projects involve algorithm design and formal complexity analysis, they are more experimental in nature, and both arise from the problems detected from the deployment of the IBM intranet search engine. Furthermore, all my experiments entail processing big data. For example, in order to extract statistical information (e.g., ngram frequency and word correlation) from 10 million documents, I built a Lucene index with a total size of hundreds of gigabytes, which took several days to construct.

Other Work. To help administrators understand ranking decisions made by the search engine, I implemented a web-based toolkit to visualize and compare *search provenance*, that is, the process of matching the given query against relevant documents. This toolkit has been deployed in the IBM intranet for a few months and found to be extremely useful.

3 Future Work

As an extension of my dissertation research, I plan to study keyword queries in the context of workflows, and investigate efficient labeling approaches for keyword search. I am also interested in exploring other possible techniques to enhance the scalability of query evaluation, such as disk-based approaches, bit vector compression, or parallel computation.

Besides efficiency, I also intend to improve the relevance of query results for searching workflows. Since keyword queries (as well as some structural queries) are inherently ambiguous and can match many results with various levels of relevance, effective ranking is very important. Compared with IR-style ranking, workflow ranking should take into account not only the statistics extracted from the metadata but also the structure of workflows.

Finally, I would also like to explore ways of using ideas from processing provenance data to process big graph-structured data in other application domains such as social networks. Instead of pursuing general solutions to arbitrary graph data, I would like to design scalable approaches which exploit the specific nature of the target domain. For example, the desired techniques for social networks would take advantage of the fact that certain common graph patterns are frequently observed from the network structure.

References

- [1] Zhuowei Bao, Sarah Cohen Boulakia, Susan B. Davidson, Anat Eyal, and Sanjeev Khanna. Differencing provenance in scientific workflows. In *proceedings of the 25th International Conference on Data Engineering (ICDE)*, pages 808–819, 2009.
- [2] Zhuowei Bao, Sarah Cohen Boulakia, Susan B. Davidson, and Pierrick Girard. Pdif-view: Viewing the difference in provenance of workflow results. *proceedings of the 35th International Conference on Very Large Data Bases (VLDB)*, 2(2):1638–1641, 2009.
- [3] Zhuowei Bao, Susan B. Davidson, Sanjeev Khanna, and Sudeepa Roy. An optimal labeling scheme for workflow provenance using skeleton labels. In *proceedings of the 29th ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 711–722, 2010.
- [4] Zhuowei Bao, Susan B. Davidson, and Tova Milo. A fine-grained workflow model with provenance-aware security views. In *proceedings of the 3rd USENIX Workshop on the Theory and Practice of Provenance (TaPP)*, 2011.
- [5] Zhuowei Bao, Susan B. Davidson, and Tova Milo. Labeling recursive workflow executions on-the-fly. In *proceedings of the 30th ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 493–504, 2011.
- [6] Zhuowei Bao, Susan B. Davidson, and Tova Milo. View-adaptive labeling for fine-grained workflows. In *submission to ACM SIGMOD International Conference on Management of Data (SIGMOD)*, 2012.
- [7] Zhuowei Bao, Benny Kimelfeld, and Yunyao Li. A graph approach to spelling correction in domain-centric search. In *proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL)*, pages 905–914, 2011.
- [8] Zhuowei Bao, Benny Kimelfeld, and Yunyao Li. Automatic suggestion of query-rewrite rules for enterprise search. In *preparation for ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2012.
- [9] Susan B. Davidson, Zhuowei Bao, and Sudeepa Roy. Hiding data and structure in workflow provenance. In *proceedings of the 7th International Workshop on Databases in Networked Information Systems (DNIS)*, pages 41–48, 2011.
- [10] Mengmeng Liu, Svilen R. Mihaylov, Zhuowei Bao, Marie Jacob, Zachary G. Ives, Boon Thau Loo, and Sudipto Guha. Smartcis: integrating digital and physical environments. In *proceedings of the 28th ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 1111–1114, 2009.
- [11] Mengmeng Liu, Svilen R. Mihaylov, Zhuowei Bao, Marie Jacob, Zachary G. Ives, Boon Thau Loo, and Sudipto Guha. Smartcis: integrating digital and physical environments. *SIGMOD Record*, 39(1):48–53, 2010.