Research Statement

Dong Deng (dongdeng@csail.mit.edu)

Data integration is the process of combining heterogeneous data from disparate sources and providing the user with a unified representation of that data. It has been a long-standing challenge in the data management community and will become increasingly important in the future due to the increasingly need to combine data both within and outside of organizations. Data integration has many applications in the real world. For example, General Electric (GE) has 75 different procurement systems, through which employees can make purchases. The company can save several hundreds million dollars per year if each of the procurement officers could access the terms and conditions that his 74 counterparts had negotiated and then demand the best pricing at the time of contract renewal. To do so, the 75 supplier databases must be integrated[1]. Another example where data integration has an important role is water quality monitoring in the California delta, where 17 independent studies are being run simultaneously. Integrating the data from these studies can answer questions that cannot be answered by any of the individual programs alone[2]. My research focuses on tackling the theoretical and systems challenges in data integration.

The three key processes in data integration are: (1) data preparation and data exploration, which includes searching and browsing relevant data, (2) entity resolution, and (3) entity consolidation. Data preparation includes data extraction, which is the process of extracting data from unstructured and semi-structured data sources and converting them into tables (*a.k.a.* relations), data cleaning, which detects and possibly repairs anomalies and constraint violations, and schema alignment (i.e., lining up like attributes in the tables). Entity resolution produces clusters of records in the tables thought to represent the same real-world entity while entity consolidation constructs a single record for each cluster that contains the canonical value for each attribute. An oft-cited statistic is that data scientists spend 80% of their time on these three processes[3]. The remaining 20% is spent doing the desired analytic tasks. In order to decrease the 80% "janitor work" needed for data analysis "in the wild", I am working on a project, DataCivilizer, with researchers at MIT and QCRI. DataCivilizer is a data management system designed to facilitate the end-to-end workflow required for a data integration task [3, 19]. I have written two of the components in DataCivilizer, namely entity resolution and entity consolidation. The challenges I have needed to overcome include:

(1) Challenges in Entity Resolution. At the core of entity resolution is the task of deciding if two records represent the same entity. In general, previous approaches have first calculated the pairwise similarity of all records using various similarity measures. Then a classifier (e.g., SVM, decision-tree, or random-forest) is employed to classify each pair of records as matches or non-matches. The matched pairs can then be clustered using clustering algorithms. "In the wild," this activity must be performed at scale, since, for example, GE has a total of 10 million supplier records in the 75 data sources mentioned above. Because pairwise similarity is a quadratic calculation, a naive approach would compare 100 trillion pairs of records. Moreover, some similarity measures are also computationally expensive. Thus, efficient and scalable algorithms are required to calculate the pairwise similarity for different similarity measures. To address these challenges, I have developed the *first algorithm that breaks the quadratic barrier* to find all set pairs with overlap size greater than a given constant threshold [1]. I have also developed an algorithm to efficiently find all similar sequence pairs with edit distance within a given threshold. This algorithm won a competition held by EDBT/ICDT in 2013 and *outperformed the second place competitor by more than 10×* [21].

(2) Challenges in Entity Consolidation. Entity consolidation is usually thought of as producing a "golden record" for each cluster of duplicate records, which contains the canonical value for each attribute. The golden records will be used in downstream analyses, and any error introduced here will be propagated through all of the subsequent steps in the analytics pipeline. Thus, the key challenge in entity consolidation is building a trustworthy solution. To ensure trustworthiness, it is crucial to have a human guide the entity consolidation process. However, a human can only answer a limited number of questions. Therefore I have developed a hybrid human-machine solution to judiciously use human effort. Experiments have shown that by only asking a small number of yes/no questions, my solution was able to produce comparable precision to, and much higher recall than, solutions based on human-written rules.

I now turn to explaining my previous results in more detail. After that I will explore my future research plans.

[1] http://fortune.com/2017/05/17/startup-saved-ge-millions/

[2] https://goo.gl/FF4n2k

[3] https://goo.gl/UH2YGT

Entity Resolution

Similarity join finds all the string pairs with similarity above a given threshold, which is a key step in many entity resolution methods. For example, the first step in the crowdsourced entity resolution framework I developed [24] is conducting a similarity join on all the records. Then, crowd workers are asked to check whether certain record pairs are matches, which will be used to infer answers for other pairs. To efficiently address the similarity join problem under different measures, I have proposed multiple scalable algorithms as described below.

Set Similarity Join. Given a collection of sets and a constant threshold c, the overlap set similarity join finds all the set pairs with an overlap measure at least c. I have designed *the first algorithm that breaks the quadratic barrier*. It has the time complexity of $o(n^2) + O(k)$ where n is the total size of all sets and k is the number of results [1]. The key idea is to categorize the sets as either "small sets" or "large sets" by applying a size boundary to the size of each set. My size-aware algorithm uses different methods to process the large sets and small sets. Experiments on real data confirm the high performance and near-linear scalability of the size-aware algorithm. I have also proposed an efficient algorithm to find all the similar set pairs with Jaccard/cosine similarity above a given threshold [6]. In contrast to existing methods that are typically based on prefix filtering techniques, I proposed to use 1-deletion neighborhoods (subsets by eliminating 1 element) to prune almost all of the dissimilar pairs and to calculate the similarity measure for only the remaining pairs. The algorithm achieves better performance than existing solutions, especially when the threshold is high.

Sequence Similarity Join. Sequence similarity joins finds all sequence pairs with edit distance within a given threshold τ. Compared to set-based similarity measures, calculating the edit distance is expensive. To avoid wasting time on dissimilar sequence pairs, I have developed a filter-and-refine algorithm PassJoin [14]. The key idea is to partition a sequence r into $\tau + 1$ disjoint segments. Based on the pigeon-hole principle, for any other sequence s to be similar to r, s must have a subsequence that is identical to one of the segments of r. To avoid comparing each segment with every subsequence of s, I have designed an optimal method that *selects the minimum number of subsequences to compare with the segments while guaranteeing completeness* (no results will be missed). In the similarity join competition held by EDBT/ICDT 2013, PassJoin came first and *outperformed the second place team by more than 10×* in both the city name (short sequences) and DNA reads (long sequences) datasets [21]. PassJoin can be seamlessly extended to support the edit similarity measure [16].

All of the similarity join algorithms I have developed are easy-to-deploy. The algorithms only require a threshold and the input data, and do not have any parameters to be fine-tuned. To show the scalability of similarity joins, I have extended my algorithms to run on Hadoop [11] and Spark [25]. My techniques have also been applied to similarity joins on records with multiple attributes [22] and to infer the concepts of web tables using knowledge bases [7].

Crowdsourced Entity Resolution. In crowdsourcing, a human is paid to answer questions. I have designed a crowdsourced entity resolution framework, which reduces the monetary cost required to achieve high quality entity resolution [24]. Usually, a pair of records with higher similarity measure is more likely to match. Based on this observation, the framework defines a partial order on the pairs of records based on their similarities: one pair of records with larger similarities on all attributes precedes another pair of records with lower similarities. Then, it selects a pair of records as a question and asks the crowd to check whether the records in the pair refer to the same entity. After receiving the answer to the question, my algorithm infers the answers to other pairs based on the partial order. This framework iteratively selects pairs without answers to ask until it gets the answers for all pairs.

Entity Consolidation

A key idea in building a trustworthy solution for entity consolidation is to use a human in the loop. In real-world scenarios, Master Data Management (MDM) is often employed to perform entity consolidation using a "match-merge" module, which is driven by a collection of human-written match and merge rules. However, it is well understood that such solutions often fail to scale to complex problems, especially those with large numbers of clusters and/or records. To address this issue, I have proposed a new solution [2]. It first automatically mines candidate matching rules (substring pairs that could be replaced by each other, such as `Avenue` ↔ `Ave`) from the clusters, and then groups these rules into sets with common characteristics (such as `9` ↔ `9th` and `6` ↔ `6th`) to allow humans to verify them in bulk. The common characteristics are based on both the syntax of the rules and the programs that describe how the two substrings transform to each other. Next, the human-approved rule groups are applied to merge the duplicate attribute values in the clusters. I have shown that by only asking a small number of yes/no questions, my solution was able to produce comparable precision to, and much higher recall than, solutions based on human-written rules.

Data Preparation

I have also conducted some research in various tasks in data preparation. For schema alignment, I have developed an algorithm to efficiently align a collection of schemas using hybrid similarity measures [4]. The hybrid measures first use the normalized measures (e.g., Jaccard and cosine similarity) to evaluate the similarity between two columns and then align the columns by the maximum matching. For data extraction, I have proposed several algorithms to efficiently extract all the entities in a pre-defined dictionary that appear in an unstructured document using different similarity measures [12, 15, 8, 17]. For data exploration, I have designed several similarity search algorithms to efficiently answer range queries (find all strings in a dataset with edit distance to the query string within a threshold) and top-k queries (find k strings in a dataset with the smallest edit distance to the query string) [11, 9, 26, 23]. I have also developed an error-tolerant autocompletion system to instantly provide users with results as users input queries. It can answer both range queries and top-k queries at interactive speed [6]. For data cleaning, I have conducted an experimental study to compare different data cleaning tools using real world datasets [20] and developed an algorithm to identify abbreviations and acronyms [13]. I have also worked on the database decay problem [18].

Future Research

My general research goal is to improve data usability and enable easier sharing and reuse of data. To accomplish this, I plan to conduct research in the following areas.

Massive Data Lake Management. Data lakes are becoming pervasive. However, they lack effective tools to manage them. Therefore, I intend to build a system to manage massive data lakes. As part of my research methodology, I plan to focus on open data, both because of its availability for scientific research and because of its importance to governments and the society at large. For this purpose, I propose to first design a new information retrieval mechanism for massive data lakes. Database queries are not good at this as they require users to know table schemas in advance. `PageRank` plays an important role in search engines for retrieving web pages. However, `PageRank` is not always a good measure for disparate tables, which are usually far less connected compared to web pages. Moreover, using keyword search makes it difficult to express structural information in the tables in data lakes. Two key components in a new information retrieval mechanism are table ranking and table recommendations. To rank tables, one must use structural information. For example, the information in the first one or two columns in a table may be more important than that in the other columns. Once the user finds a relevant table using table ranking, a table recommendation module will suggest more tables based on both table relatedness and structural information.

Trustworthy Data Cleaning. Real-world data contains various kinds of errors, e.g., typos, incompleteness, and inconsistencies. These data errors have been recognized as an inevitable hurdle in turning raw data into information, according to the New York Times, Forbes, and other major business publications. To make raw data usable, I propose to study the data cleaning problem and build a tool to detect errors in raw data and repair them in a trustworthy way. I envision that this tool can perform data cleaning when the user enters data, at which point it can suggest standardized reformulations based on historical data. Since error detection is often done using human-written data quality rules and error repairing will change the data, it is necessary to involve human experts in the loop. However, in some cases, it may be easier for a human to give some example errors in the data and the corresponding repairs than to specify a rule. I also propose to study how to reverse engineer the error detection and repairing rules from a few examples provided by the end-users using program synthesis techniques.

Incremental Data Cleaning and Integration. In the real world, data is often acquired not all at once but incrementally. The human effort spent on data cleaning and integration should be reused when new data becomes available. For this purpose, I propose to build a system to enable incremental data cleaning. The system should support a pay-as-you-go data cleaning since a human can only answer a limited number of questions. In addition, the historical answers provided by human can ease the human effort needed for new questions. Thus given a new question, the system should be able to find the previous, relevant questions and their answers. Finally, since data cleaning and data integration can reinforce each other (cleaner data can lead to better integration results while matched data can provide additional opportunities to clean the data), it will be interesting to study how to conduct the two tasks iteratively to maximize their performance.

In the next 5-10 years, I will use a broad range of techniques, including machine learning, data management, information retrieval, and programming languages to address the problems raised in the above proposals.

References

[1] Dong Deng, Yufei Tao, Guoliang Li: *Overlap Set Similarity Joins with Theoretical Guarantees.* **SIGMOD Conference 2018**

[2] Dong Deng, Wenbo Tao, Ziawasch Abedjan, Ahmed Elmagarmid, Ihab F. Ilyas, Samuel Madden, Mourad Ouzzani, Michael Stonebraker, Nan Tang: *Entity Consolidation: The Golden Record Problem.* **CoRR abs/1709.10436**: (2017)

[3] Dong Deng, Raul Castro Fernandez, Ziawasch Abedjan, Sibo Wang, Michael Stonebraker, Ahmed K. Elmagarmid, Ihab F. Ilyas, Samuel Madden, Mourad Ouzzani, Nan Tang: *The Data Civilizer System.* **CIDR 2017**

[4] Dong Deng, Albert Kim, Samuel Madden, Michael Stonebraker: *SilkMoth: An Efficient Method for Finding Related Sets with Maximum Matching Constraints.* **PVLDB 10(10)**: 1082-1093 (2017)

[5] Dong Deng, Guoliang Li, He Wen, H. V. Jagadish, Jianhua Feng: *META: An Efficient Matching-Based Method for Error-Tolerant Autocompletion.* **PVLDB 9(10)**: 828-839 (2016)

[6] Dong Deng, Guoliang Li, He Wen, Jianhua Feng: *An Efficient Partition Based Method for Exact Set Similarity Joins.* **PVLDB 9(4)**: 360-371 (2015)

[7] Dong Deng, Yu Jiang, Guoliang Li, Jian Li, Cong Yu: *Scalable Column Concept Determination for Web Tables Using Large Knowledge Bases.* **PVLDB 6(13)**: 1606-1617 (2013)

[8] Dong Deng, Guoliang Li, Jianhua Feng, Yi Duan, Zhiguo Gong: *A Unified Framework for Approximate Dictionary-based Entity Extraction.* **VLDB J. 24(1)**: 143-167 (2015)

[9] Dong Deng, Guoliang Li, Jianhua Feng: *A Pivotal Prefix based Filtering Algorithm for String Similarity Search.* **SIGMOD Conference 2014**: 673-684

[10] Dong Deng, Guoliang Li, Shuang Hao, Jiannan Wang, Jianhua Feng: *MassJoin: A MapReduce-based Method for Scalable String Similarity Joins.* **ICDE 2014**: 340-351

[11] Dong Deng, Guoliang Li, Jianhua Feng, Wen-Syan Li: *Top-k String Similarity Search with Edit-Distance Constraints.* **ICDE 2013**: 925-936

[12] Dong Deng, Guoliang Li, Jianhua Feng: *An Efficient Trie-based Method for Approximate Entity Extraction with Edit-Distance Constraints.* **ICDE 2012**: 762-773

[13] Wenbo Tao, Dong Deng, Michael Stonebraker: *Approximate String Joins with Abbreviations.* **PVLDB 11(1)**: 53-65 (2018)

[14] Guoliang Li, Dong Deng, Jiannan Wang, Jianhua Feng: *PassJoin: A Partition-based Method for Similarity Joins.* **PVLDB 5(3)**: 253-264 (2011)

[15] Guoliang Li, Dong Deng, Jianhua Feng: *Faerie: Efficient Filtering Algorithms for Approximate Dictionary-based Entity Extraction.* **SIGMOD Conference 2011**: 529-540

[16] Guoliang Li, Dong Deng, Jianhua Feng: *A Partition-based Method for String Similarity Joins with Edit-Distance Constraints.* **ACM Trans. Database Syst. 38(2)**: 9:1-9:33 (2013)

[17] Guoliang Li, Dong Deng, Jianhua Feng: *Extending Dictionary-based Entity Extraction to Tolerate Errors.* **CIKM 2010**: 1341-1344

[18] Michael Stonebraker, Dong Deng, Michael L. Brodie: *Database Decay and How to Avoid It.* **IEEE BigData 2016**: 7-16

[19] Raul Castro Fernandez, Dong Deng, Essam Mansour, Abdulhakim Ali Qahtan, Wenbo Tao, Ziawasch Abedjan, Ahmed K. Elmagarmid, Ihab F. Ilyas, Samuel Madden, Mourad Ouzzani, Michael Stonebraker, Nan Tang: *A Demo of the Data Civilizer System.* **SIGMOD Conference 2017**: 1639-1642

[20] Ziawasch Abedjan, Xu Chu, Dong Deng, Raul Castro Fernandez, Ihab F. Ilyas, Mourad Ouzzani, Paolo Papotti, Michael Stonebraker, Nan Tang: *Detecting Data Errors: Where Are We and What Needs to Be Done?.* **PVLDB 9(12)**: 993-1004 (2016)

[21] Sebastian Wandelt, Dong Deng, Stefan Gerdjikov, Shashwat Mishra, Petar Mitankin, Manish Patil, Enrico Siragusa, Alexander Tiskin, Wei Wang, Jiaying Wang, Ulf Leser: *State-of-the-art in String Similarity Search and Join.* **SIGMOD Record 43(1)**: 64-76 (2014)

[22] Guoliang Li, Jian He, Dong Deng, Jian Li: *Efficient Similarity Join and Search on Multi-Attribute Data.* **SIGMOD Conference 2015**: 1137-1151

[23] Jin Wang, Guoliang Li, Dong Deng, Yong Zhang, Jianhua Feng: *Two Birds with One Stone: An Efficient Hierarchical Framework for Top-k and Threshold-based String Similarity Search.* **ICDE 2015**: 519-530

[24] Chengliang Chai, Guoliang Li, Jian Li, Dong Deng, Jianhua Feng: *Cost-Effective Crowdsourced Entity Resolution: A Partial-Order Approach.* **SIGMOD Conference 2016**: 969-984

[25] Ji Sun, Zeyuan Shang, Guoliang Li, Dong Deng, Zhifeng Bao: *Dima: A Distributed In-Memory Similarity-Based Query Processing System.* **PVLDB 10(12)**: 1925-1928 (2017)

[26] Minghe Yu, Jin Wang, Guoliang Li, Yong Zhang, Dong Deng, Jianhua Feng: *A Unified Framework for String Similarity Search with Edit-Distance Constraint.* **VLDB J. 26(2)**: 249-274 (2017)