D-Dupe: An Interactive Tool for Entity Resolution in Social Networks

Mustafa Bilgic *
University of Maryland
College Park, MD

Louis Licamele †
University of Maryland
College Park, MD

Lise Getoor ‡
University of Maryland
College Park, MD

Ben Shneiderman §
University of Maryland
College Park, MD

ABSTRACT

Visualizing and analyzing social networks is a challenging problem that has been receiving growing attention. An important first step, before analysis can begin, is ensuring that the data is accurate. A common data quality problem is that the data may inadvertently contain several distinct references to the same underlying entity; the process of reconciling these references is called *entity-resolution*. D-Dupe is an interactive tool that combines data mining algorithms for entity resolution with a task-specific network visualization. Users cope with complexity of cleaning large networks by focusing on a small subnetwork containing a potential duplicate pair. The subnetwork highlights relationships in the social network, making the common relationships easy to visually identify. D-Dupe users resolve ambiguities either by merging nodes or by marking them distinct. The entity resolution process is iterative: as pairs of nodes are resolved, additional duplicates may be revealed; therefore, resolution decisions are often chained together. We give examples of how users can flexibly apply sequences of actions to produce a high quality entity resolution result. We illustrate and evaluate the benefits of D-Dupe on three bibliographic collections. Two of the datasets had already been cleaned, and therefore should not have contained duplicates; despite this fact, many duplicates were rapidly identified using D-Dupe's unique combination of entity resolution algorithms within a task-specific visual interface.

Keywords: Data cleaning and integration, user interfaces, visual analytics, visual data mining.

Index Terms: H.2.8 [Information Systems]: Database Applications—Data mining; H.5.2 [Information Interfaces and Presentation]: User Interfaces—User-centered design

1 INTRODUCTION

There is a growing interest in tools which support the analysis of social networks. In order for these tools to work effectively and convey accurate visual and analytic information, the underlying data must be clean. Unfortunately this is rarely the case. Often networks are extracted from databases which may contain errors and inconsistencies.

One common problem is that a dataset may contain multiple references to the same underlying entity or actor. In a graph visualization of such a network, a single actor would be represented by multiple nodes. This visual display is clearly misleading; it not only has the incorrect number of nodes but in turn the edges and paths are inaccurate. Furthermore, calculating any of the standard social network measures, such as degree-centrality, betweenness and so on, would give inaccurate results.

Entity resolution is the process of reconciling, from the underlying data references, the "actual" real-world entities [2]. Traditional entity resolution approaches use similarity metrics which compare the attributes of the references. Entity resolution in social networks is more interesting because, in addition to making use of attribute similarities to identify potential duplicates, the social context, or "who's connected to who," can provide useful information to the resolution process. Recently a number of approaches have been developed which make use of relational information to help in the resolution process [3, 27, 4].

Most existing entity resolution methods focus on automated entity resolution. Automated techniques are not perfect, and they face a "precision-recall" trade-off. If they are tuned to have high precision, they rarely merge duplicates, leaving many duplicates in the database. If they are tuned to have a high recall, they mistakenly merge nodes that are in fact distinct. On the other hand, hand-cleaning methods, even with visualization support, can be slow and inefficient in finding duplicates. These approaches tend to be high precision, because there is a human-in-the-loop making the final resolution decision. However, inspecting a large dataset and hunting for duplicates can be like looking for the proverbial needle in a haystack. Thus, while these approaches may have high precision, they tend to have low recall.

Here, we provide an interactive analyst-centric approach to the problem which tightly integrates the data mining techniques with a visualization suited to the task. D-Dupe [6] provides access to sophisticated entity resolution algorithms and enables users to flexibly apply sequences of actions to uncover duplicates. In addition, D-Dupe provides users with a simple network visualization which displays the *collaboration context* for potential duplicates. The collaboration context shows, for any two potential duplicates, their relational neighborhood. The network visualization allows users to quickly identify shared and non-shared relational context and base their exploration and resolution decisions on the context. Emerging principles from information visualization, such as laying out the nodes on a meaningful substrate, are combined with representations for uncertainty, resulting in a tool that is especially well suited to the entity resolution task. Powerful filtering and search techniques are also integrated into the tool.

Two of D-Dupe's novelties are:

1. **Stable Visual Layout Optimized for Entity Resolution:** Instead of visualizing the whole collaboration network, D-Dupe shows only the subnetwork relevant for the entity resolution task. Such a dramatic simplification reduces the users' cognitive load as the networks presented are much simpler, easier to understand, and yet they still contain the information relevant to the task at hand. Furthermore, the simplification allows our visualization to scale to large networks. We also develop a visual layout tuned to the entity resolution task; the nodes are laid out on a stable and meaningful substrate where the potential duplicates and other related entities always appear at the same location, leading to considerable reduction in scanning the network.

2. **User Control for Combining Entity Resolution Algorithms:** Numerous similarity measures can be used to determine potential duplicates; some are good at finding misspellings, others may find abbreviations and so on. Moreover,

*mbilgic@cs.umd.edu
†licamele@cs.umd.edu
‡getoor@cs.umd.edu
§ben@cs.umd.edu

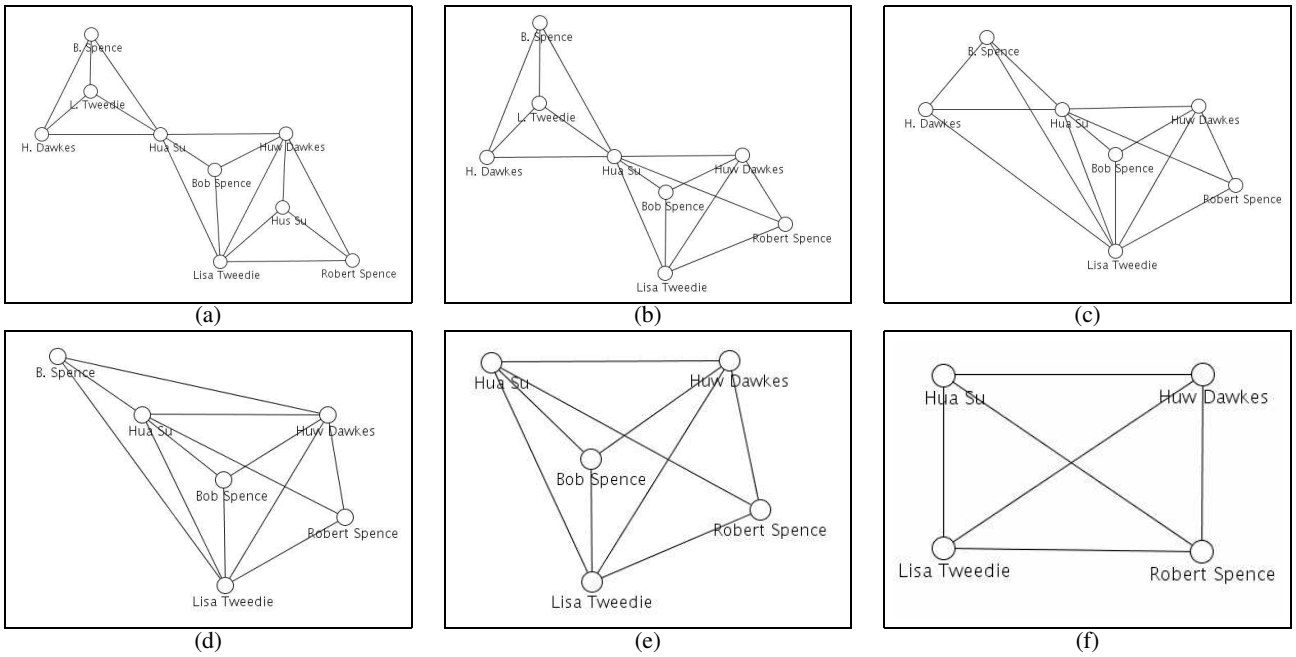

Figure 1: A series of resolutions on a portion of the InfoVis data set. (a) shows the initial network before any resolutions have been performed. It is apparent that there are a number of duplicates. (b) through (f) show the same network, each drawn after each resolution in the process The resolutions are: (a) to (b) "Hua Su" and "Hus Su", (b) to (c) "Lisa Tweedie" and "L. Tweedie", (c) to (d) "Huw Dawkes" and "H. Dawkes", (d) to (e) "Bob Spence" and "B. Spence", and (e) to (f) "Robert Spence" and "Bob Spence". Note the simplicity of the final network in contrast to the original network.

decisions made using one measure might uncover new potential duplicates under another measure. D-Dupe allows users to flexibly apply and interleave different measures. It is hard to get the same benefit from automated combination of the measures. In our case studies, we found this feature, integrated with the visualization of the common social context, to be extremely effective.

Throughout, in both our running examples and our evaluation datasets, we illustrate and evaluate D-Dupe on bibliographic collaboration networks. However D-Dupe's layout and interaction principles are general and can be used in other social networks in which the relational context provides useful information for entity resolution decisions. We show the utility of D-Dupe on three bibliographic datasets; in each we were able to quickly and effectively resolve duplicates. This is particularly impressive, since two of the three datasets had already been extensively cleaned.

2 A Motivating Example

Before going into the specific details of the tool, Figure 1 gives an overview of the deduplication process on a small portion of bibliographic dataset used for the InfoVis 2004 Contest [15]. The dataset describes the papers and authors culled from eight years (1995-2002) of the InfoVis conference. Figure 1(a) shows a co-author network for a portion of the dataset. In this network, a node represents an author, and two authors are linked if they have published a paper (in the dataset) together. It is immediately apparent that the network in Figure 1(a) contains a number of duplicates. Figure 1(b)-(f) shows the transformation the network undergoes, as duplicate authors are found and merged. Figure 1(g) shows the final network, after all of the duplicates have been resolved. As we can see, we have quickly gone from a rather complex network, in the start, to a relatively simple network at the end. More importantly, comparing Figure 1(a) with Figure 1(f) reveals that visualization of datasets with duplicates will lead to incorrect conclusions.

3 Design Components

Our goal with D-Dupe is to help automate the process of bringing potential duplicates to the users' attention, supporting the users in making a resolution decision (deciding whether or not two nodes are in fact duplicates) and allowing the users to flexibly chain together multiple resolutions.

The basic interaction paradigm for D-Dupe is as follows. Users begin by loading a dataset. They can then choose from a number of possible entity resolution algorithms. The entity resolution algorithms use a variety of different similarity metrics to rank pairs of nodes according to how likely they are to be duplicates. The users can scroll through the list of potential duplicate pairs and select a potential duplicate pair for analysis. They can then view the collaboration context network for the pair and apply filtering and highlighting features of D-Dupe to this network. Users can resolve the potential duplicate by deciding that the two nodes are: 1) duplicates, in which case the nodes are merged, or 2) distinct entities, in which case the nodes are marked as distinct. User actions are recorded, and at any point in the process the 'resolved' network can be saved. A video D-Dupe demonstration is available at `http://www.cs.umd.edu/linqs/ddupe/`.

D-Dupe is written in Java and will run on any system with a Java Virtual Machine. D-Dupe makes use of JUNG's [23] visualization support for social networks and uses several string distance measures from SecondString [10], in addition to a Levenstein edit distance algorithm that we implemented.

Figure 2 shows the D-Dupe interface. The tool consists of three coordinated windows [22]: the collaboration context network panel on the left, the entity resolution control panel on the right, and the potential duplicates details panel at the bottom. We describe the capabilities supported in each window in the following subsections.

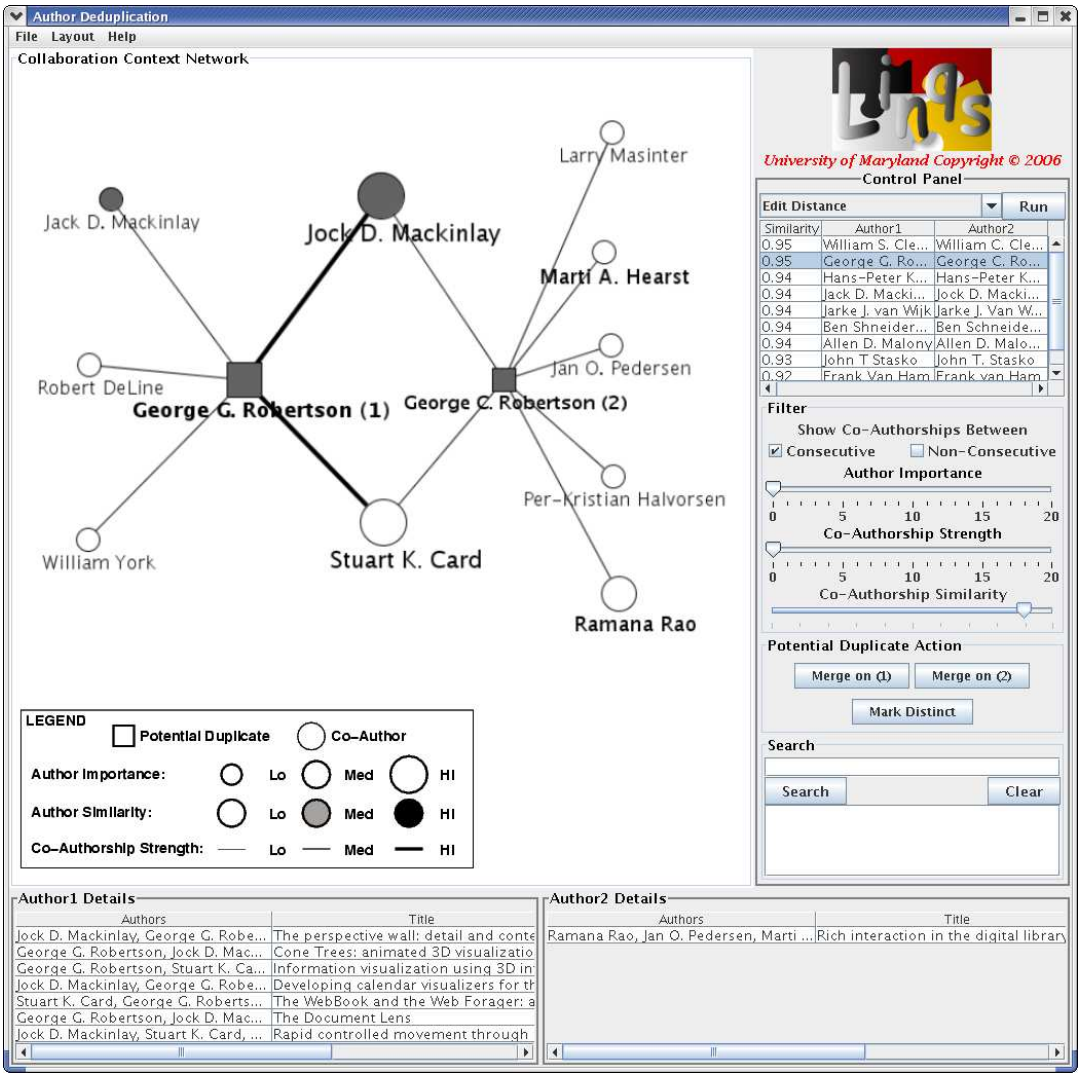

Figure 2: Overview: The layout consists of three coordinated windows. The relevant collaboration context network is shown in the main window. Details on the current candidate duplicate pair are presented in potential duplicates detail panel in the lower window. The entity resolution control panel appears on the right; the user can select an entity similarity measure to use, view a list of candidate duplicate pairs, choose filters for the nodes, edges and collaborators, perform resolutions for a particular pair, and search for a particular author.

3.1 Collaboration Context Network

One of the first challenges in the design of D-Dupe was deciding how to present the users with the collaboration context for potential duplicates. Presenting the full collaboration network is only feasible for extremely small networks. As Figure 3(a) shows, even for moderately sizes datasets, viewing the entire network is ineffective because the network is unreadable. Instead, as mentioned earlier, D-Dupe uses a task-specific network visualization. This visualization is based on the paradigm of interactive visualization where the users inspect each potential duplicate individually. In this paradigm, users first choose a potential duplicate pair to analyze and they are then presented with the relevant subnetwork for that pair. Only the potential duplicates, their neighbors, and relationships among them are shown. Figure 3(b) shows the result for the pair of potential duplicates "George G. Robertson" and "George C. Robertson" (The chosen pair is shown as square nodes in the graph). The collaboration context network shown in Figure 3(b) uses an an off-the-shelf spring embedder method, Fruchterman-Reingold layout [17] algorithm, for laying out the nodes. This simplifies the network sufficiently so that the network is readable, while still containing relevant context information for the entity resolution task. While this layout is a significant improvement over viewing the entire network, its disadvantage is that it is not stable. Each time a potential duplicate pair is analyzed, the nodes will be placed at different locations on the screen, as determined by the spring embedder algorithm. This randomness causes the cognitive overhead of scanning the network to find the potential duplicates which is burdensome in this repetitive task.

This disadvantage led us to to develop meaningful substrates for node placement. These produce a stable layout which reduces unnecessary cognitive overhead for the entity resolution task. The substrates divide the screen into five regions: the first potential duplicate is always at the center of the second region and the second potential duplicate is always at the center of the fourth region. The third region highlights their shared neighbors. Their non-shared neighbors are displayed in the first and fifth regions respectively. Figure 3(c) shows an example of the substrates for the "George Robertson" references. In the center, we see their shared

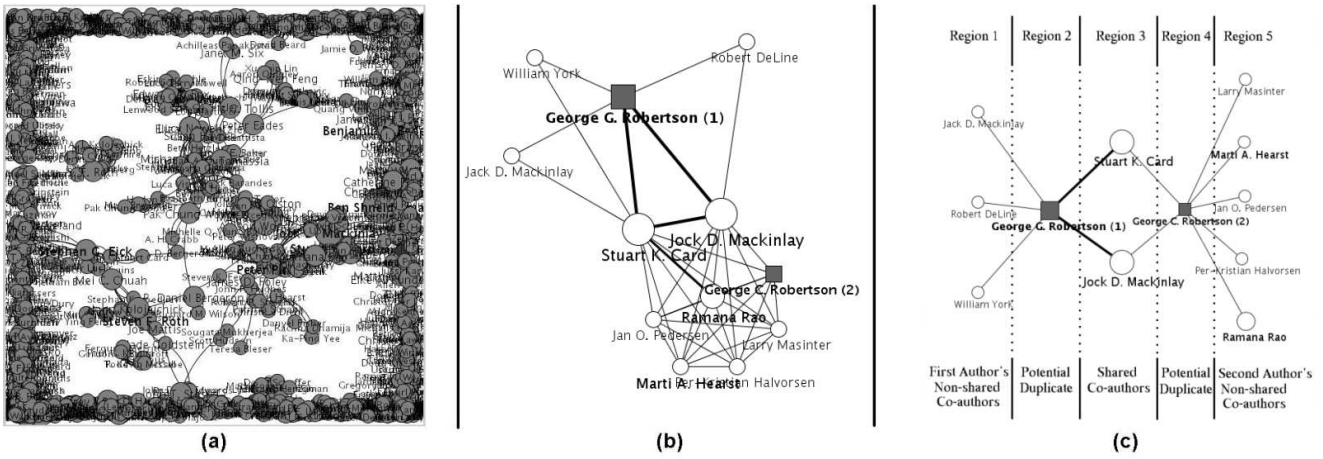

Figure 3: The evolution of the layout. (a) The original collaboration network using the Fruchterman-Reingold layout with no pruning. (b) Showing only the collaboration context network for the potential duplicates with the most commonly used force directed layout (c) The collaboration context subnetwork shown using the stable layout. The potential duplicates are shown in region 2 and 4, the shared neighbors are shown in region 3 and non-overlapping neighbors are shown in regions 1 and 5.

co-authors, in this case "Stuart K. Card" and "Jock D. Mackinley". On the far left, we see the non-shared co-authors of "George G. Robertson" and on the far right, we see the non-shared co-authors of "George C. Robertson".

By default, we show only the links between the potential duplicates and their co-authors, we do not show the co-author links among the co-authors. This results in a simpler graph and eliminates links among nodes in the same substrate and between nodes in non-consecutive substrates. However, the co-author links between potential duplicates' neighbors can be shown, by checking the show non-consecutive edges check-box in the control panel (discussed in more detail in the next section).

This pleasingly simple design has proved to be surprisingly effective. In a preliminary study with five participants to validate the usefulness of this layout, we studied the response time and accuracy for users on 10 collaboration context networks. Users were presented half of the networks using the Fruchterman-Reingold layout and half using the stable layout, with the order appropriately randomized within subjects, and they were asked to determine the number of shared neighbors the potential duplicates shared. For this simple task, there was no difference in accuracy between the two layouts; however there was a 15% reduction in response time when the stable layout was used. The benefit of the layout becomes more apparent as users gain experience with the tool.

Additional information about the nodes is conveyed through shape, shading, and size. The current potential duplicate nodes are squares, and the other nodes are circles. The current potential duplicate pair and other potential duplicates in the neighborhood are shaded according to their similarity based on the current entity resolution metric. Darker nodes indicate a greater degree of similarity. The similarity shading for the nodes in the neighborhood can be controlled using a slider. The node and label size is a function of the "importance" of the node, where "importance" is dictated by the domain semantics. In the bibliographic domain, we define importance as the number of publications attributed to the author. This provides an additional channel to visually show semantic differences between the nodes. The links also convey information; edge thickness indicates the strength of a relationship. In the bibliographic domain, the relationship strength between two nodes can be defined as the number of times authors have co-authored together.

Often, there may be additional potential duplicates among the co-authors of the currently analyzed potential duplicate. Being able to spot these additional potential duplicates is important for making the correct decision about the current potential duplicate. We provide a slider which controls the similarity between the co-authors of the potential duplicate pair. Depending on the currently chosen similarity measure and the threshold, nodes which have a potential duplicate in the neighborhood will be shaded. Nodes in each region are sorted according to their similarity to the nodes in other regions; similar nodes are highlighted and appear together at the top, ranked according to their similarity.

We chose a node-link representation because of its familiarity to social network researchers. Textual lists could be used for a compact representation but additional coding would be needed to show similarity, number of publications, number of co-authorships, etc.

3.2 Entity Resolution Control Panel

The Entity Resolution Control Panel (right side of Figure 2) provides the main functionalities for D-Dupe users. The actions they may perform include:

- Choose an entity similarity measure
- Select the current potential duplicate pair
- Filter the collaboration context network
- Resolve a potential duplicate pair by merging the nodes or marking them distinct
- Search the database for a particular author
- Save the resolved collaboration network

Potential Duplicates List: D-Dupe users can select from a variety of entity similarity measures in the drop down menu at the top of the control panel. By selecting a measure and clicking the **Run** button, users populate the potential duplicates list. This list is sorted with the most similar pairs of nodes at the top.

Filters: D-Dupe allows dynamic filtering of the collaboration context network. As mentioned earlier, users can filter the edges shown by choosing to show only co-author links among consecutive regions, co-author links among the authors in the non-consecutive regions, or both. D-Dupe supports filtering authors based on importance. Users can control the importance of the displayed authors using a slider to set the importance threshold. D-Dupe also supports filtering based on edge weights, set using the co-authorship strength

slider. As mentioned earlier, we also provide a slider which controls the co-authorship similarity.

Potential Duplicate Action: D-Dupe provides an easy way to resolve duplicates. When users are satisfied that the potential duplicate pair that they are inspecting is truly a duplicate they can merge the pair. When merging, users select the author they want to keep by select **Merge on (1)** or **Merge on (2)**.After the merge, the co-authors links are updated to refer to the newly resolved node. On the other hand, if users decide that the potential duplicates correspond to different entities, then they can select **Mark Distinct**. After performing a merge or distinct action, this author pair will no longer be presented to users. Each resolution step is recorded, so that users can examine a history of the resolution decisions. We currently do not support undoing resolutions, because the resolutions are chained together and they result from complex interdependencies. However this is an interesting topic for future research.

Querying: In some cases, users are interested in resolving references for a particular author. D-Dupe users can search the database for specific authors, generating a list of the closest matches according to the currently selected similarity measure. If users select one of these authors, the potential duplicates table will be populated with potential duplicates for that author. Users can also query on a particular author by double-clicking on it in the collaboration context network.

3.3 Potential Duplicates Details Panel

D-Dupe provides two tables, shown at bottom of Figure 2, with details for the current potential duplicate pair. For the bibliographic domain, the window is used to show the publications of each author. By definition, more important authors will have more publications. This extra information allows users to see if the duplicate pairs share additional attributes. In addition, users can double click on a publication, and D-Dupe will search Google Scholar for the paper. For other domains, D-Dupe can go to another online source such as the white pages or company personnel file for additional information.

4 Entity Similarity Measures

D-Dupe users can select from a variety of entity similarity metrics to identify and rank potential duplicates. D-Dupe uses several standard string similarity functions including Levenstein, Jaccard, JaccardChar, Jaro, JaroWinkler and MongeElkan. Users can easily switch between measures to explore the benefits of each measure. Different entity similarity measures are appropriate for finding different kinds of errors. For example, in the bibliographic domain common errors that lead to duplicates in databases are: 1) parsing-errors, such as switching a first name and last name, 2) abbreviations, such as using first initial instead of full first name, and, of course, 3) misspellings. To deal with the first two, the distance measures need to compare terms in the string rather than characters; Jaccard similarity works well for this purpose. To address the misspellings, measures that do comparison at the character level, such as Levenstein, JaccardChar, Jaro, JaroWinkler, and MongeElkan work well.

Since users control resolution decisions, they can chain together resolution decisions based on different similarity metrics. This fine-grained flexibility, while seemingly simple, is not easily achievable in an automatic system. Simply applying the algorithms in some fixed order does not support the complex dependencies that may be discovered in carefully chosen sequences.

5 Case Studies with Three Bibliographic Datasets

We evaluated D-Dupe on three bibliographic datasets. The first two datasets were considered "clean" in that the providers claimed that duplicates had already been removed. Common practices for cleaning data include automated approaches that use a particular entity resolution algorithm and ones that rely on hand cleaning without much automated guidance.

- **InfoVis Contest:** 614 publications from 1974 to 2004 and 1,036 authors, with 1,832 co-authorship links between authors [15]. This dataset was provided as a cleaned dataset for the InfoVis contest in 2004. The contest organizers made a substantial effort to resolve duplicates by asking people within the InfoVis community to point out and resolve duplicates for themselves as well as their co-authors and friends. This intensive process, distributed over many individuals took several months; however, the cleaned dataset still had duplicates.
- **CiteSeer subset:** 1504 publications by 1167 authors, cleaned by its developers [18] and further cleaned by Aron Culotta and Andrew McCallum for use in evaluating entity resolution algorithms within the machine learning community. The method used to initially clean this collection used simple, high-recall methods to "over-merge" entities. Next a domain expert split up the clusters if required—an intensive manual process. Specifically, to resolve the author names, the researchers began by normalizing the author strings by initialing the first name and merging all the author references with the same normalized string. To account for misspellings, approximate string matching algorithms were used. In the manual post-processing step, a web search was performed for resources that could help make an informed decision about when clusters should be split. The researchers spent roughly 8-12 hours resolving this dataset.
- **PubMed subset:** Subset of 56 papers by 161 authors retrieved from a query of PubMed, a carefully built database from the U. S. National Library of Medicine. Unlike the other two datasets, PubMed does not provide identifiers for authors, so our results illustrate how D-Dupe can help label an unprocessed dataset.

We next describe example deduplication task sequences in each of these datasets to highlight D-Dupe's functionalities.

5.1 InfoVis

Figure 4 shows the sequence of resolutions corresponding the example from Section 2. Figure 4(a) shows the potential duplicate pair "Hua Su" and "Hus Su" in the InfoVis dataset. The center of the collaboration context network shows their two shared co-authors, which is a good indication that they are in fact duplicates. Without domain knowledge, however, users may not be sure whether these two references are the same entity, so they can examine the paper reference using Google Scholar. After seeing the original papers, it seems clear that "Hus Su" is in fact a misspelling of "Hua Su." After merging them, the neighbors of the secondary author are transferred to the primary author. In the next step, Figure 4(b), "Hua Su" is shown in green to highlight that it is the result of a recent resolution and thus we are more confident in its identity. Because merging "Hua Su" and "Hus Su" leads to changes in the network structure, it will be wise to inspect Hua Su's co-authors for potential duplicates. Additional resolutions are shown in Figure 4(b), (c), (d), and (e). We stop at Figure 4(f), where there are no more duplicates.

If the collaboration context network consists of two disjoint subnetworks, i.e. the potential duplicates do not have any shared co-authors, it is a strong indication that these potential duplicates represent distinct authors. Figure 5 shows an example where users may be unsure of the identities of "Jian Huang" and "Qian Huang". Based solely on the author names, users might mistakenly merge these authors. But by observing the lack of shared context visually, users will pause to investigate their conclusion more carefully. This

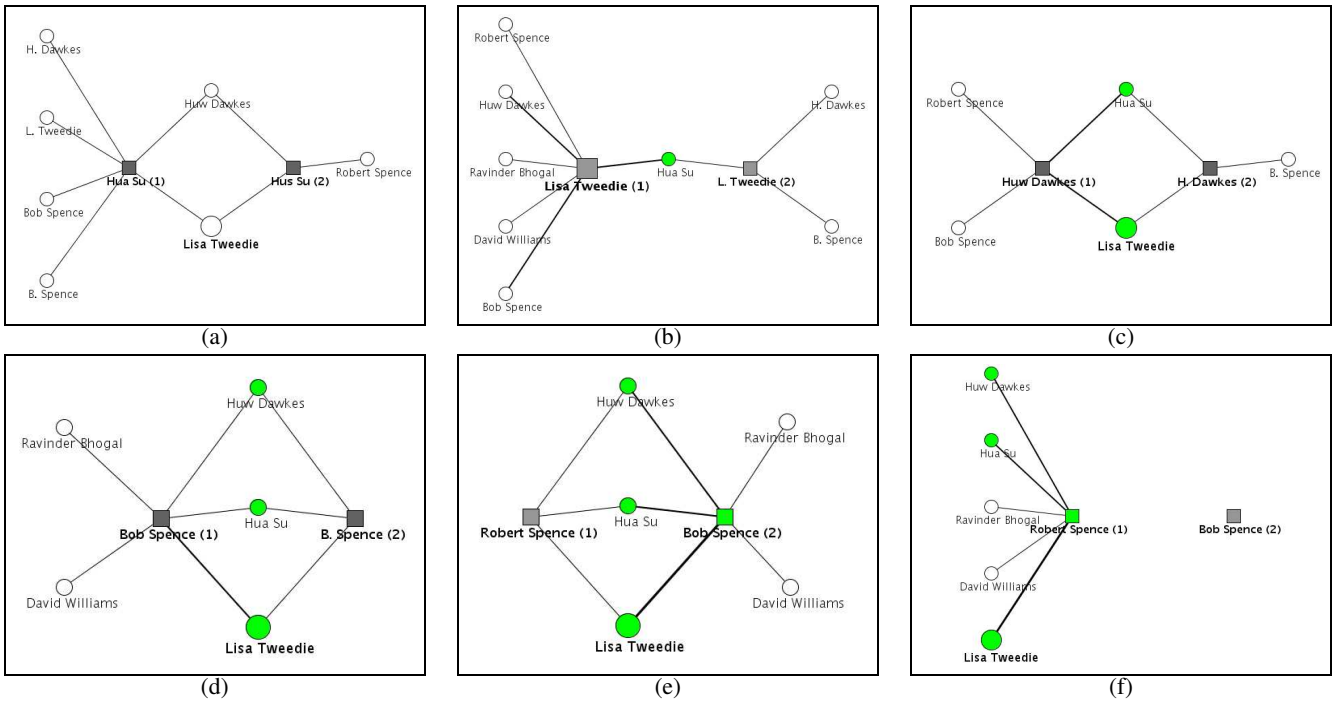

Figure 4: The sequence of collaboration context networks corresponding to the entity resolution steps from the motivating example in Section 2. The resolution process is iterative; an earlier decision effects the next decision. Resolved nodes in the earlier steps are drawn in green, indicating a higher confidence in their identities. The iterative process ends at (f), where we do not have any more duplicates to consider in the network.

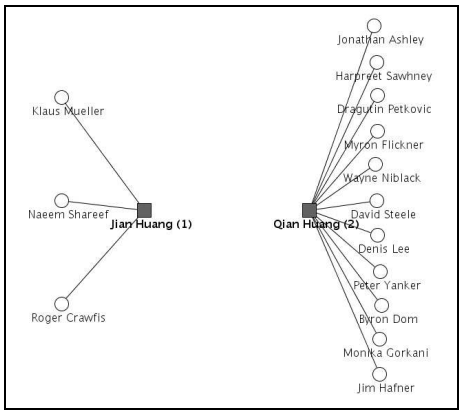

Figure 5: An example where the link structure helps in deciding that two similar nodes are in fact distinct entities.

example demonstrates how the network also highlights distinctions between the potential duplicates.

Figure 6(a) shows another another pair of potential duplicates, "George G. Robertson" and "George C. Robertson", from the InfoVis dataset. Because there are few publications for "George C. Robertson," it is drawn as a small node. This may be an indication that it is a misspelling. Figure 6(b) shows the collaboration context network after the user decreased the threshold for co-authorship similarity. Another potential duplicate pair in the neighborhood, "Jack D. Mackinlay" and "Jock D. Mackinlay", is now apparent. Figure 6(c) shows the result after further filtering using the edge based filtering and node based filtering; this illustrates how users might quickly isolate the "George C. Robertson" node from the rest of the network, revealing additional evidence that it might be a misspelling.

Figure 7 shows another example from the InfoVis dataset, for the potential duplicates "Steven K. Feiner" and "S. K. Feiner". Initially, the potential duplicates do not seem to have any common co-authors. But, after lowering the threshold for co-authorship similarity, "M. X. Zhou" and "Michelle X. Zhou" are highlighted as potential duplicates. This additional evidence may increase users' confidence that "Steven K. Feiner" and "S. K. Feiner" may be duplicates. This example shows another novel way in which D-Dupe can help users, by drawing attention to potentially important nodes in the neighborhood of the authors currently being inspected.

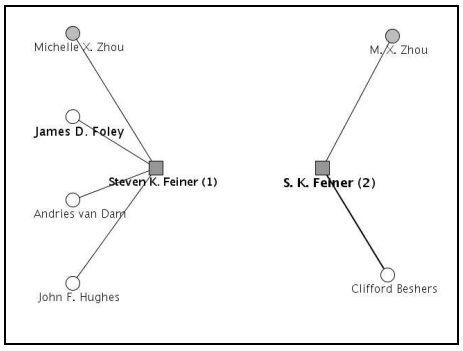

Figure 7: Although these potential duplicates do not share co-authors, we can see that there are potential duplicates in their neighborhoods.

In our experience, D-Dupe enabled us to rapidly find duplicates in the InfoVis dataset. Although this database was carefully prepared from electronic sources, and used by dozens of research groups in a highly visible contest, we were able to detect more than 60 duplicates within our first half hour of use.

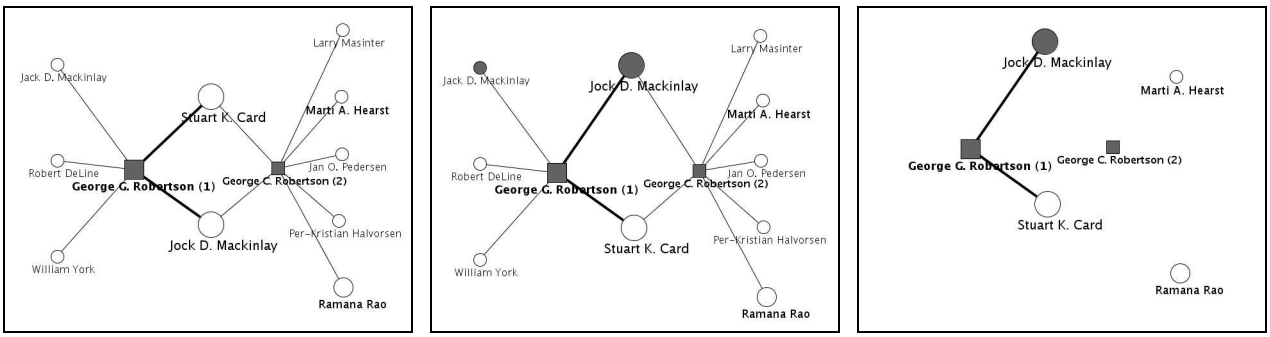

Figure 6: (a) The initial collaboration network for potential duplicates "George G. Robertson" and " George C. Robertson." (b) The use of the Co-Authorship Similarity Slider highlights another potential duplicate among the neighbors: "Jack D. Mackinlay" and "Jock D. Mackinlay." (c) Filtering the collaboration network using the node and edge weights quickly isolates "George C. Robertson" from the rest of the network signaling that it might be a misspelling.

5.2 CiteSeer

The next dataset we consider is the CiteSeer dataset. As an already "cleaned" dataset, used for the evaluation of entity resolution algorithms, we were surprised to see that within only a minute of using D-Dupe we were able to find seven duplicates. These duplicates had quite high similarity measures according to the Jaccard similarity measure. Each of these duplicates were due to parse errors. After these pairs are merged and there are no more pairs that users feel confident in merging under this similarity measure, users can switch to another similarity measure. JaccardChar is similar to Jaccard except that instead of using tokens, it operates at the character level. Using JaccardChar, D-Dupe shows three very similar pairs which require further inspection. Of these, only one is a true duplicate. One potential duplicate pair is "C Codognet" and "P Codognet". The collaboration context network shows that these two authors are co-authors and therefore are likely to be distinct, because an author would not be a co-author of himself/herself. This is one of the constraints that must be met in author networks and is easily detected by visual inspection using D-Dupe.

Using a third entity similarity measure, the MongeElkan measure, D-Dupe is able to detect another parse error. It indicates that "Philips A" and "Philips Andrew B Philips Stevens" are potential duplicates. Users may have high confidence in merging these two entities because they share two co-authors. Finally, D-Dupe detects one more potential duplicate pair using the JaroWinkler distance measure. The pair of "Weiss S" and "Wess S" can be merged after the Google Scholar search shows that the author on the paper for "Weiss" is in fact listed as "Wess".

In our use of D-Dupe for this supposedly clean CiteSeer subset, we were able to detect and resolve 10 duplicates in 20 minutes. As illustrated above, the flexible combination of similarity measures greatly increased our ability to resolve duplicates.

5.3 PubMed

We have shown that D-Dupe can be helpful in finding duplicates in two datasets previously cleaned by others. We next turn to a setting in which D-Dupe works on a database for which no author extraction has been attempted. An example of such a database is the PubMed dataset maintained by NCBI. A paper record has only the authors' names associated with it, but does not have any author IDs. The PubMed records were obtained by querying for "Giardia Translation" producing 56 papers with 161 author references. In loading the data into D-Dupe, we assumed that each author reference was unique and considered all potential duplicate pairs.

In this dataset, we found and resolved the potential duplicates in 20 minutes using the attribute similarity measures and the co-authorship similarity slider. We found seven duplicates by using Jaccard similarity measure, two by JaccardChar, and two more using Jaro. In the end, from the 161 author references, 11 duplicate author entities were identified.

6 Related Work

There has been a surge of recent interest in social network analysis. Not surprisingly, at the same time, there has been extensive work on visualizing social networks [16, 8, 14]. A number of nice graph visualization and social networks packages have been developed in the past several years; a non-exhaustive list of the popular packages includes UCINET [7], Pajek [13], JUNG [23], Prefuse [19], and GUESS [1]. Because our work focuses on cleaning the data, before it is input to these more general social network analysis and visual analytics tools, in some ways this work is orthogonal.

There has also been extensive work on finding and cleaning duplicates in the machine learning and database communities. Most of that work has focused on automatic methods rather than interactive support. Traditional approaches make use of only attribute information, where entities are matched based on the values of their attributes. Much of this work focused on defining approximate string matching algorithms [20, 21, 10] and fuzzy matching [9]. Other attribute-based approaches have been adaptive [26, 25, 5, 11]. More recent work has focused on combining attribute information with the relational structure of the domain [3, 27, 4]. In these works, the relational graph is taken into account for finding possible duplicates. Within the database community, there has been work on interactive data cleaning [24, 12], but unlike D-Dupe which is designed for resolving network data, the interactive data cleaning work typically focuses on a single table.

One of the challenges of visual analytics is data representation [28]. As Thomas and Cook state, "Data are at the heart of the analytic challenge. These data, in their raw form, are rarely appropriate for direct analysis." D-Dupe addresses an element of this analytic challenge and solves it using an interface which effectively combines visual and analytic information for data cleaning in an interactive tool.

7 Discussion

Our evaluation highlights D-Dupe's performance on three bibliographic datasets. We believe that is fairly straightforward to use D-Dupe on other social network data as well, as long as the actors and the relationships are well defined. To best exploit D-Dupe's multiple functionalities, the domain should exhibit the following properties:

- The actors should have properties that can be used by the attribute similarity functions. It is relatively easy to extend the entity resolution algorithms with domain-specific algorithms supplied by the users.
- The collaboration between actors should be informative about the entity resolution task, so that the visualization of the collaboration network helps in the decision process.
- The node and edge importance should be informative for the deduplication task so that the filtering of nodes and edges helps in the decision process.

We do not present results here, but we have investigated applying D-Dupe to other tasks including name resolution in email collections, place resolution in geospatial databases, and name resolution in academic genealogy datasets. We have presented demos to experts in these domains and received very encouraging feedback.

8 Conclusion

D-Dupe integrates data mining algorithms with an interactive information visualization interface to support an important analytic task: entity resolution. The stable and meaningful layout presents small subnetworks from large databases in an task-appropriate, simple, and surprisingly effective design for visually presenting information about potential duplicates. The ability to flexibly apply sequences of similarity measures enables users to be highly effective in entity resolution tasks, because they often exhibit complex interdependencies. This provides a potent environment for decision making and recording of user actions for later review.

By giving users control over the decision making process, they can develop improvements to the data mining algorithms and learn about the distinctive problems in their data. In using D-Dupe, we easily found duplicates in 'gold-standard' entity resolution datasets; in one of our cleaned datasets we found that 6% of the nodes represented duplicate entities. We believe that D-Dupe illustrates the utility of building interactive tools that combine data mining and information visualization to support specific analytic tasks.

Acknowledgments

This work was supported by the National Science Foundation under NSF #0423845 and NSF #0438866.

References

[1] E. Adar. GUESS: a language and interface for graph exploration. In *ACM SIGCHI Conf. on Human Factors in Computing Systems*, pages 791–800, 2006.

[2] O. Benjelloun, H. Garcia-Molina, Q. Su, and J. Widom. Swoosh: A generic approach to entity resolution. Technical report, Stanford University, 2005.

[3] I. Bhattacharya and L. Getoor. Iterative record linkage for cleaning and integration. In *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pages 11–18, 2004.

[4] I. Bhattacharya and L. Getoor. *Entity Resolution in Graphs*, chapter Mining Graph Data (Lawrence B. Holder and Diane J. Cook, eds.). Wiley, 2006.

[5] M. Bilenko and R. J. Mooney. Adaptive duplicate detection using learnable string similarity measures. In *ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 39–48, 2003.

[6] M. Bilgic, L. Licamele, L. Getoor, and B. Shneiderman. D-Dupe: An interactive tool for entity resolution in social networks (poster). In *Int. Symp. on Graph Drawing*, pages 505–507, 2005.

[7] S. Borgatti, M. G. Everett, and L. C. Freeman. UCINET 6, 2006.

[8] U. Brandes, T. Raab, and D. Wagner. Exploratory network visualization: Simultaneous display of actor status and connections. *Journal of Social Structure*, 2(4), 2001.

[9] S. Chaudhuri, K. Ganjam, V. Ganti, and R. Motwani. Robust and efficient fuzzy match for online data cleaning. In *ACM SIGMOD Int. Conf. on Management of Data*, pages 313–324, 2003.

[10] W. W. Cohen, P. Ravikumar, and S. E. Fienberg. A comparison of string distance metrics for name-matching tasks. In *IJCAI Workshop on Information Integration on the Web*, pages 73–78, 2003.

[11] W. W. Cohen and J. Richman. Learning to match and cluster large high-dimensional data sets for data integration. In *ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 475–480, 2002.

[12] T. Dasu and T. Johnson. *Exploratory Data Mining and Data Cleaning*. John Wiley & Sons, Inc., 2003.

[13] W. de Nooy, A. Mrvar, and V. Batageli. *Exploratory Social Network Analysis with Pajek*. Cambridge University Press, 2005.

[14] A. H. Dekker. Visualisation of social networks using CAVALIER. In *Australasian Symp. on Information Visualisation*, pages 49–55, 2001.

[15] J. D. Fekete, G. Grinstein, and C. Plaisant. Contest, the history of InfoVis. In *IEEE Symposium on Information Visualization*, 2004.

[16] L. C. Freeman. Visualizing social networks. *Journal of Social Structure*, 1(1), 2000.

[17] T. M. J. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software Practice and Experience*, 21(11), 1991.

[18] C. L. Giles, K. Bollacker, and S. Lawrence. CiteSeer: An automatic citation indexing system. In *ACM Int. Conf. on Digital Libraries*, pages 89–98, 1998.

[19] J. Heer, S. K. Card, and J. A. Landay. prefuse: a toolkit for interactive information visualization. In *ACM SIGCHI Conf. on Human Factors in Computing Systems*, pages 421–430, 2005.

[20] A. E. Monge and C. Elkan. The field matching problem: Algorithms and applications. In *ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 267–270, 1996.

[21] G. Navarro. A guided tour to approximate string matching. *ACM Computing Surveys*, 33(1):31–88, 2001.

[22] C. North and B. Shneiderman. A taxonomy of multiple window coordinations. Technical Report CS-TR-3854, University of Maryland, 1997.

[23] J. O'Madadhain, D. Fisher, P. Smyth, S. White, and Y. B. Boey. Analysis and visualization of network data using JUNG. *Journal of Statistical Software*, 2005.

[24] V. Raman and J. Hellerstein. Potter's wheel: An interactive data cleaning system. In *Int. Conf. on Very Large Data Bases*, pages 381–390, 2001.

[25] E. S. Ristad and P. N. Yianilos. Learning string-edit distance. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 20(5):522–532, 1998.

[26] S. Sarawagi and A. Bhamidipaty. Interactive deduplication using active learning. In *ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 269–278, 2002.

[27] P. Singla and P. Domingos. Multi-relational record linkage. In *ACM SIGKDD Workshop on Multi-Relational Data Mining*, 2004.

[28] J. J. Thomas and K. A. Cook, editors. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE CS Press, 2005.