An Effective Approach to Entity Resolution Problem Using Quasi-Clique and its Application to Digital Libraries

Byung-Won On, Ergin Elmacioglu, Dongwon Lee*
Penn State University
{on,ergin,dongwon}@psu.edu

Jaewoo Kang
NCSU
kang@csc.ncsu.edu

Jian Pei
Simon Fraser U.
jpei@cs.sfu.ca

ABSTRACT

We study how to resolve entities that contain a group of related elements in them (e.g., an author entity with a list of citations or an intermediate result by GROUP BY SQL query). Such entities, named as *grouped-entities*, frequently occur in many applications. By exploiting contextual information mined from the group of elements per entity in addition to syntactic similarity, we show that our approach, *Quasi-Clique*, improves precision and recall unto 91% when used together with a variety of existing entity resolution solutions, but never worsens them.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Search and Retrieval

General Terms

Algorithms, Performance, Experimentation

Keywords

Entity Resolution, Name Disambiguation, Graph Partition

1. INTRODUCTION

Since the existence of duplicate or variant entities degrades the quality of data collection severely, it is important to de-duplicate them. Such a problem is, in general, known as the **Entity Resolution (ER)** problem. In particular, we focus on the **Grouped-Entity Resolution (GER)** problem, where each *grouped-entity* has "a group of elements" in it. Examples include authors with a paper list or actors with a movie list. Note that this problem cannot be completely avoided since not all entities in data collections carry a unique ID system. By and large, previous approaches to the ER problem (e.g., [1, 3]) work as follows: (1) the information of an entity, e, is captured in a data structure, $D(e)$, such

---

*Contact author. His research is partially supported by Microsoft SciData Award (2004).

as a multi-attribute tuple or an entropy vector; (2) a binary distance function, f, such as the edit distance is prepared; (3) the distance of two entities, e_1 and e_2, is measured as that of the corresponding data structures, $D(e_1)$ and $D(e_2)$, using function f: $dist(e_1, e_2) = f(D(e_1), D(e_2))$; and (4) finally, if the result, $dist(e_1, e_2)$, is less than certain threshold, ϕ, then the two entities are variants: $r < \phi \rightarrow e_1 \sim e_2$. Although working well in many scenarios, this approach often suffers from a large number of *false positives* (i.e., an entity determined to be a variant when it is not). Consequently, the overall recall and precision suffer. If a user asks for top-k answers, such false positives can even override correct variants out of the answer window of $|k|$, degrading the precision substantially. Therefore, in this paper, we aim at devising an algorithm that resolves grouped-entities effectively.

2. THE TWO-STEP ALGORITHM

To exploit a wealth of information hidden in a group of elements per grouped-entity, we capture "contextual information" in context graphs through superimposition (step 1), and measure their contextual similarity in terms of *Quasi-Clique* (step 2).

(Step 1) Suppose we mine a *context graph* out of an author entity A's co-author tokens: B, C, D, and E. First, a vertex is prepared for tokens, A through E, referred to as $V(A)$ through $V(E)$. Then, four co-author vertices, $V(B)$ through $V(E)$, are connected to the main vertex $V(A)$, forming a graph, g_a. Next, g_a is "superimposed" to the *base graph* G, a collaboration graph pre-built using the entire set of co-authors from all entities. For instance, if an author C had co-authored with an author D elsewhere, then now g_a will have an edge connecting $V(C)$ and $V(D)$. The final g_a, then, is the context graph. At the end, if all neighboring co-authors of A have co-authored each other, then g_a becomes a clique. Similarly, for venue information, once we create an initial graph, g_a, we can superimpose it against a base graph. For instance, one may use, as a base graph, a venue relation graph where an edge between two venue vertices represents the "semantic" similarity of two venues (e.g., how many authors have published in both venues). The superimposition works as long as there is a base graph (e.g., collaboration graph, venue relation graph) onto which an entity's graph can be superimposed. For more general cases, a base graph can be also constructed using the co-occurrence relationship among tokens.

(Step 2) Once the contexts of entities are captured and represented as context graphs, their similarity can be properly modeled using *Quasi-Clique* [4]. A connected graph G is a

---

Algorithm 1: distQC

Input: A grouped-entity e, an ER method $\mathcal{M}$, and
three parameters (α, γ and $\mathcal{S}$).
Output: k variant grouped-entities, e_v ($\in E$), such
that $e_v \sim e$.

1 Using $\mathcal{M}$, find top $\alpha \times k$ candidate entities, e_X;
2 $G_c(e) \leftarrow$ context graph of e;
3 forall e_i ($\in e_X$) do
4 $G_c(e_i) \leftarrow$ context graph of e_i;
5 $g_i \leftarrow$ QC($G_c(e)$, $G_c(e_i)$, γ, $\mathcal{S}$);
**6** Sort e_i ($\in e_X$) by $|g_i|$, and return top-k;

γ-*quasi-complete graph* ($0 < \gamma \leq 1$) if every vertex in the
graph has a degree at least $\gamma \cdot (|V(G)| - 1)$. In a graph G, a
subset of vertices $S \subseteq V(G)$ is a γ-*Quasi-Clique* ($0 < \gamma \leq 1$)
if $G(S)$ is a γ-quasi-complete graph, and no proper superset
of S has this property. Clearly, a 1-*Quasi-Clique* is a clique.
In a network (e.g., a social network or a citation network)
scenario, a *Quasi-Clique* is a set of objects that are highly
interactive with each other. Therefore, a *Quasi-Clique* in
a graph may strongly indicate the existence of a potential
community. Since a *Quasi-Clique* contains a group of highly
interacting (and thus likely highly similar in role) objects,
it may be more reliable in representing relationships than
individual objects.

While γ value indicates the compactness of *Quasi-Clique*,
another parameter that is of interest is the number of ver-
tices of *Quasi-Clique*. We denote this parameter as $\mathcal{S}$. For
a graph G, therefore, functions: (1) QC(G, γ, $\mathcal{S}$) returns a
γ-*Quasi-Clique* graph g from G with $|V(g)| \geq \mathcal{S}$ if it exists,
and (2) QC(G_1, G_2, γ, $\mathcal{S}$) returns a *common γ-Quasi-Clique*
graph g of G_1 and G_2 with $|V(g)| \geq \mathcal{S}$. Then, $|g|$ indicates
how strongly two graphs G_1 and G_2 are related, and can
be used as a "distance." Our two-step GER algorithm, dis-
tQC, is shown in Algorithm 1. Given an entity e ($\in E$),
to locate matching k variant entities, the distQC algorithm
first relies on any existing ER solutions, and selects α (e.g.,
$2 \leq \alpha \leq 3$) times more number of candidates than k as an
extra. Since we try to improve precisions by reducing false
positives, once we get more candidates, if we can boost up
those correct variants up into higher ranks in subsequent
steps, then our aim can be met. γ value controls the "quasi-
ness" of the graph, and $\mathcal{S}$ works as the minimum filter.

3. DISCUSSION

For validating our proposal, we used one real data set
(ACM) and four synthetic data sets (ACM , BioMed, Econ-
Papers, and IMDB). Each data set has solution sets to check
correctness of our proposal (due to space constraint, we omit
details here). As evaluation metrics, we used the *Ranked
Precision* (measuring precision at different cut-off points).
Formally, *Ranked Precision* = $\frac{\sum_{i \in C} precision_i}{r}$, where $precision_i$
is the precision at a cut off of i, and C is the window size. Fi-
nally, the *Average Ranked Precision (ARP)* are the averages
of recall and ranked precision over all cases.

Due to space constraint, we only report two ARP re-
sults here (others show similar pattern). We first ran and
measured the performance of three distance metrics – Jac-
card (JC), TFIDF (TI) and IntelliClean (IC) [2] – which
showed good performance in [3]. Then, to each, *Quasi-
Clique* is applied in step 2, and the performance is measured
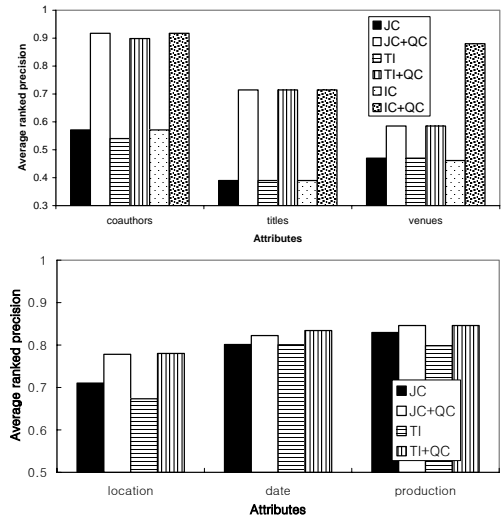
**Figure 1: (Top) Real test case for ACM; (Bottom)
Synthetic test case of IMDB.**

as JC+QC, TI+QC, and IC+QC (i.e., "before" and "after"
Quasi-Clique-based metric is applied). Figure 1 (Top) illus-
trates ARP of these six schemes. Note that *Quasi-Clique*
improved the precision visibly. For instance, the precision
of JC+QC (resp. TI+QC) significantly improves from JC
(resp. TI) on co-authors. On average, precision improves
by 63%, 83%, and 46% for three attributes, respectively. In
general, JC and TI are simple distance metrics, measuring
distances based on the occurrences of common tokens. Since
some authors have name initials and common first names on
co-author data, therefore, these metrics tend to generate a
large number of false positives as shown in Figure 1. Since
Quasi-Clique uses additional information as to how closely
co-authors of the authors are correlated each other on graph
representation, it overcomes the limitations of the simple
metrics.

Figure 1 (Bottom) shows the ARP of IMDB data set.
Compared with citation data sets, distQC performs only
slightly better than string distance metrics because: (1)
records and attribute values of an actor and her variant en-
tities have no strong relationships unlike those of citations;
and (2) attribute values of citations are long while those of
IMDB data set are short, carrying less meaningful informa-
tion. Nevertheless, distQC never worsens the ARP.

Conclusion. Toward the GER problem, we presented a
graph partition based approach using *Quasi-Clique*. Unlike
conventional ER solutions, our approach examined contex-
tual relationships hidden under the grouped-entities, and
showed promising results.

4. REFERENCES

[1] M. A. Hernandez and S. J. Stolfo. "The Merge/Purge
Problem for Large Databases". In *ACM SIGMOD*, 1995.

[2] M. L. Lee, W. Hsu, and V. Kothari. "Cleaning the Spurious
Links in Data". *IEEE Intelligent System*, 19(2):28–33, 2004.

[3] B.-W. On, D. Lee, J. Kang, and P. Mitra. "Comparative
Study of Name Disambiguation Problem using a Scalable
Blocking-based Framework". In *ACM/IEEE JCDL*, Jun.
2005.

[4] J. Pei, D. Jiang, and A. Zhang. "On Mining Cross-Graph
Quasi-Cliques". In *ACM KDD*, Aug. 2005.