Rule Based Method In Entity Resolution For Efficient Web Search

R. Kalpanadevi@soumiya, K. Baskar, A. Kumarasan, K. Vijayakumar

1,2,3,4 Department of computer science, SKP Engineering College Thiruvannamalai.

Received 28 January 2017; Accepted 22 March 2017; Available online 28 April 2017

ABSTRACT

Entity resolution is problem of identifying and linking/grouping different manifestations of the same real word object. ER is one of the most important problems in data cleaning and arises in many applications such as information integration and information retrieval. It contain a new class of rules which could describe the complex matching conditions between records and entities, based on this class of rules were proposed and by presenting the rule-based entity resolution problem, an on-line approach for ER can be developed. The implementation of the process in web search for efficient retrieval of data through the web mining is done through clustering and ranking method which improves the efficiency in retrieving the data and reducing the searching time of the user. Thus, the efficiency can be improved by time based interval by comparing the search process with other search engines.

KEYWORDS: Entity Resolution, Web Search, Ranking

INTRODUCTION

Data mining refers to the mining or discovery of new information in terms of patterns or rules from vast amount of data. To be practically useful, data mining must be carried out efficiently on large files and databases. Web search can be used as mining of WWW to retrieve useful knowledge and data about user behavior, user query, content and construction of the web. Web is a collection of inter-related files on one or more web servers. Web search is a tool that allows user to find the information on the World Wide Web. The search allows for specific criteria, word or phrase and retrieves a list of references that match those criteria. It is used for efficiently and quickly retrieves the data from web. Record linkage (RL) refers to the task of finding records in a data set that refer to the same entity across different data sources (e.g., data files, books, websites, and databases). Record linkage is necessary when joining data sets based on entities that may or may not share a common identifier as may be the case due to differences in record shape, storage location, and/or curator style or preference. A data set that has undergone RL-oriented reconciliation may be referred to as being cross-linked. Record Linkage is called Data Linkage in many jurisdictions, but is the same process.

Ranking technique is the application of machine learning. The main aim of using ranking technique is to produce of a permutation of data items, in unseen list which is similar to training data set. The training data set consist list of items with partial order which is typically induced by giving numerical or binary value. Ranking is the central part of many information retrieval problems such as document retrieval, collaborative filtering and online advertising. Training data set consist of queries and documents; it may be prepared manually by human assessors (raters, called by Google), who checks the result for queries and determine the relevance of each result. The ranking technique that we specified in our project is weighted page rank algorithm. The main purpose of page rank algorithm is, to provide priority to each click through blog. Rule based ER algorithm is...
integrated with page rank algorithm to make the web searching process in efficient manner. It gives priority to each result that is provided so that user is not exploited by the unnecessary advertisement that is displayed. Thus, by combining the rule based ER algorithm with clustering, ranking and classification of data sets, we achieve the accuracy and efficiency in this current project work. Hence, the online searching is implemented efficiently with certainty.

**Related Work:**
To discover high quality Web services, a number of QoS models for Web services and QoS-driven service selection approaches have been proposed in the service computing field. In their study, it is usually assumed that a user explicitly specifies his/her interests (e.g., by using keywords) and QoS requirements, and submits them to the service discovery system. Then the service discovery system matches the user’s interests and QoS requirements with corresponding attributes of Web services, and returns those with the best matching degrees to the user. The scenarios for service selection can be divided into two categories. The first scenario aims to select a set of services for a composite service, which is widely studied by existing work on service selection. The second scenario is to select a single service for a user request, or to select multiple services with the same function for multiple user requests.

The main idea of collaborative filtering is to recommend new items of interest to a user regarding the other users experiences over a set of items. Existing collaborative filtering algorithms can be divided into two categories memory-based and model-based. Memory based methods are more popular in service recommendation, partially because they are more intuitive to interpret the recommendation results. Memory-based collaborative filtering can be further divided into user-based approaches and item based approaches. User-based collaborative filtering methods recommend a user the items preferred by the users with similar interests, while item-based collaborative filtering methods recommend a user the items similar to those he/she preferred in the past.

**Service Recommendation:**
Web services which are highly relevant to the user’s service usage history and own high QoS utility would be recommended to users. An active Web service recommendation approach based on service usage history which incorporates both user interest and QoS preference into Web service recommendation. Web services manipulated by similar users, except the services already used by the active user, are ranked depending on their semantic similarity with services the active end-user used to interact with. Finally, the top k services are recommended to the user.

**Problem Definition:**
Here we study the problem of finding true facts in a assured area. Here, a area refers to a belongings of a certain type of objects, such as authors of books or number of pixels of camcorders. The input of SSEARCH ENGINE is a large number of facts in a domain that are provided by numerous websites. There are typically numerous incompatible particulars from different websites for each object, and the goal of SSEARCH ENGINE is to identify the true fact among them.

**Basic Definitions:**
**Confidence of Facts:**
The confidence of a fact f (denoted by s(f)) is the probability of f being correct.

**Trustworthiness of Websites**
The dependability of a website w (denote by t(w)) is the expected confidence of the facts provided by w.

**Implication of Facts:**
Here implication is defined instead of similarity between facts because such relationship is asymmetric. For example, in a few domain (e.g., book authors), websites be likely to supply imperfect details (e.g., first author of a book). Presume two websites provide author information for the same book. These rules can be discovered from existing high quality data such as master data or manually identified data. Inspired by the swoosh method, each cluster is then merged into a composite record via a merge function. Finally a traditional ER method, denoted by T-ER, can be applied to identify the new data set. Moreover, in order to identify more records, the current ER result can be used as the training data to discover new ER-rules. The training data can also be obtained by using techniques, such as relevant feedback, crowd sourcing and knowledge extraction from the web. Therefore, with the accumulated information, ER-rules for more entities can be discovered.
Invalid rule:

A rule \( r \) is invalid if there exist records that match \( \text{LHS}(r) \) but do not refer to \( \text{RHS}(r) \). Invalid rules might be discovered when the information of entities is not complete. For instance, presume the preparation information place involve the records. The rule \( r: (\text{name} \equiv "\text{wei Wang}")(\text{coa} \equiv "\text{Flipkart}") \) can be generated. For \( \text{e31} \), it matches \( \text{LHS}(r) \) but does not refer to \( \text{e1} \). Therefore, \( r \) is an invalid rule.

Incomplete rule set:

An ER-rule set \( R \) of entity set \( E \) is incomplete if there are records referring to entities in \( E \) that are not covered by \( R \). Both the incomprehensive information of entities and continuous changes of entity features would cause a rule set become incomplete. To solve these problems, we develop some methods to identify candidate invalid rules and candidate useless rules and discover new effective ER-rules.

Fig. 1: Architecture of Proposed System

Existing System:

The work on entity resolution can be broadly divided into three categories. Pair wise ER is the most works on ER focus on record matching, which involves comparing record pairs and identifying whether they match. A major part of work on record matching focuses on similarity functions. To capture string variations, proposed a transformation-based framework for record matching. Some machine-learning based approaches can identify matching strings which are syntactically far apart. Similarity based on record relationships is also proposed to solve the people identification problem. Since in our work, records are not compared with each other, and our work is orthogonal to record matching. However, string similarity functions can be applied to fuzzy match operator (denoted by \( _{-} \)) in ER-rules.

For example, given a string \( s \), we say \( _{-} "\text{weiwang}" \) if the edit distance between \( s \) and \( "\text{weiwang}" \) is smaller than a given threshold. Decision trees are employed to teach record matching rules. However, decision trees cannot be used to discoverer-rules. This is because the domain of the right hand side of record matching rules is \{yes, no\} (two records are mapped or not mapped), while the domain of the right-hand side of ER-rules is an entity set of Non-Pair wise ER. The research on non-pair wise ER includes clustering strategies and classifiers. Most strategies solve ER based on the relationship graph among records, by modeling the records as nodes and the relationships as edges. Machine learning approaches are also proposed by using global information to solve ER effectively. However, these methods are not suitable for massive data because of efficiency issues. We choose a representative work for comparison.

These rules can be discovered from existing high quality data such as master data or manually identified data. Inspired by the swoosh method, each cluster is then merged into a composite record via a merge function. Finally a traditional ER method, denoted by T-ER, can be applied to identify the new data set. Moreover, in order to identify more records, the current ER result can be used as the training data to discover new ER-rules. The training data can also be obtained by using techniques, such as relevant feedback, crowd sourcing and knowledge extraction from the web. Therefore, with the accumulated information, ER-rules for more entities can be discovered.
Proposed System:

Entity resolution rules are used for record matching and identifying record in the table that is stored in a database. These rules can be discovered from existing high quality data such as master data or manually identified data. They only match two entities in the table; but could not match more than two entities. It efficiently matches the entity in the table but is difficult to user to match the entities using rule based method because they contain lot of mathematical logic to match entity. The existing project work is implemented in offline process for searching and retrieving the data. It is the process of retrieving data immediately and easily from database. In our current project, online searching process is implemented efficiently by comparing the live data in each field. The comparison process is undergone by using the rule based ER-Algorithm. The current search process provides relevant data that is based on keywords and queries given by the user for searching. By adopting tm in rule based ER-Algorithm in conjunction with clustering method (k- means clustering) and ranking method (weighted page rank algorithm) to make efficient web searching process.

Modules Description:
Module 1: Collection of unrelated:
DataFirst we have to collect the specific data about an object and it is stored in linked record. Generate counter for exact entity and accumulate the facts about a particular object

![Fig. 2: Collection of unrelated data](image)

Module 2: Data Search:
Searching the related data link according to user input. In this module user retrieve the specific data about an object.

![Fig. 3: Data Search](image)

Module 3: Truth Algorithm:
We design a general framework for the reality difficulty, and create an algorithm called fact Finder, which utilize the dealings linking web sites and their in sequence, i.e., a web site is truthful if it provide many pieces of
accurate in sequence, and a section of in sequence is probable to be factual if it is provided by many trustworthy web sites.

**Fig. 4: Truth Algorithm**

**Module 4: Result Calculation:**
For each response of the query we are calculating the presentation. With the count up intended discover the most excellent connection and show as the output.

**Fig. 5: Result Calculation**

**System Design:**

**Fig. 1: System Architecture**
Ranking Algorithm:

Page Rank is an algorithm used by Google Search to rank websites in their search locomotive outcome. Page Rank was name following Larry Page, one of the founders of Google. Page Rank is a way of measuring the importance of website pages.

The original PageRank algorithm was described by Lawrence Page and Sergey Brin in several publication. It is known by

\[
PR(A) = (1-d) + d \left( \frac{PR(T1)}{C(T1)} + \ldots + \frac{PR(Tn)}{C(Tn)} \right)
\]

where

- \(PR(A)\) is the PageRank of page A,
- \(PR(Ti)\) is the PageRank of pages Ti which link to page A,
- \(C(Ti)\) is the number of outbound links on page Ti and
- \(d\) is a damping issue which can be set between 0 and 1.

So, primary of all, we see that PageRank does not rank web sites as a entire, but is resolute for each page separately. Promote, the PageRank of page A is recursively obvious by the PageRanks of those pages which link to page A.

The PageRank of pages Ti which link to page A does not weight the PageRank of page A regularly. Within the PageRank algorithm, the PageRank of a page T is always weighted by the number of outbound links \(C(T)\) on page T. This means that the additional outbound links a page T has, the less will page A profit from a connect to it on page T.

The weighted PageRank of pages Ti is then extra up. The result of this is that an extra inbound link for page A will always increase page A's PageRank.

Finally, the sum of the weighted PageRanks of all pages Ti is multiplied with a damping factor \(d\) which can be set between 0 and 1. Thereby, the extend of PageRank benefit for a page by another page linking to it is compact.

```java
/**
 * \textcopyright PageRank provides simple API to Google PageRank Technology</b>
 * Original source: http://www.temesoft.com/google-pagerank-api.jsp
 * \textcopyright PageRank queries google toolbar webservice and returns a
 * google page rank.
 */
public class PageRank {

/**
 * List of available google datacenter IPs and addresses
 */
static final public String [] GOOGLE_PR_DATACENTER_IPS = new String[]{
   "64.233.183.91",
   "64.233.189.44",
   "66.249.89.83",
   "toolbarqueries.google.com", // this is useful, and change "search" to "tbr"
};
/**
 * Must receive a domain in form of: "http://www.domain.com"
 * @param domain - (String)
 * @return PR rating (int) or -1 if unavailable or internal error happened.
 */
public static int get(String domain) {
   int dataCenterIdx = new Random().nextInt(GOOGLE_PR_DATACENTER_IPS.length);
   int result = -1;
   JenkinsHash jHash = new JenkinsHash();

   String googlePrResult = "";
   long hash = jHash.hash("info:" + domain.getBytes());

   String url = "http://"+GOOGLE_PR_DATACENTER_IPS[dataCenterIdx]+"/tbr?client=nавclient-auto&hl=en&"+
```
try {
    URLConnection con = new URL(url).openConnection();
    con.setConnectTimeout(10000);
    InputStream is = con.getInputStream();
    byte[] buff = new byte[1024];
    int read = is.read(buff);
    while (read > 0) {
        googlePrResult = new String(buff, 0, read);
        read = is.read(buff);
    }
    if (googlePrResult.equals(""))
        return 0;
    googlePrResult = googlePrResult.split(";")[2].trim();
    result = new Long(googlePrResult).intValue();
} catch (Exception e) {
    e.printStackTrace();
}
return result;

Conclusion:
This project developed a class of ER-rules which are capable of describing the complex matching conditions between records and entities. Based on these rules, we developed an ER algorithm R-ER. We experimentally evaluated our algorithms on real data sets. The experimental results show that our algorithm can achieve a good performance both on efficiency and accuracy. We would like to extend our techniques in web search to retrieve the searching result in efficient manner. It is give high performance in searching by improving the availability of relevant data immediately for user.

REFERENCES