Towards A New Token Based Framework for Record Linkage in Arabic Data Set

Hesham H. Abdel Ghafour†, Ali El-Bastawissy††, Abdelfatah A. Hegazy†

†College of Computing and Information Technology, Arab Academy for Science &Technology, Cairo, Egypt
††Faculty of Computers and Information Technology, Cairo University, Egypt

Summary
Record linkage is the process of identifying if two records represent the same real entity or not. Record Linkage is one of the most important and most investigated issue in data quality literature. Most of the current researches have been applied on English context and these researches didn't mention the required modifications in order to be applicable in other contexts like Arabic context. Applying record linkage algorithms on Arabic context is a challenging task due to the unique characteristics of Arabic language in terms of its morphological and orthographical features. This paper proposed a token based framework for record linkage in Arabic data set. In our framework we use a new technique for Arabic name tokenization and use a new approach for similarity computation.

Keywords:
Arabic Data Cleaning, Data Quality, Duplicate Detection, Data warehouse, Entity Resolution, Record Linkage, Object Identification, String Similarity

1. Introduction

The business today depends on electronic data and cooperative information systems that integrate different systems together, those systems may relay on same or different technologies. Therefore the quality of electronic data became crucial to any organization and affects all levels starting from operational levels and up to the strategic levels. Data quality issues occur due to many reasons such as incorrect data entry, ambiguity during data transformations, erroneous applications, populating databases, faulty database design, and data obsolescence. In data quality literatures, the duplicate data is one of the most important and most investigated issue due to its critical effects on business decision's quality. Duplicate data may exist in a single database or while integrating data from different data sources to build a data warehouse system.
Record linkage is the process that handle duplicate data by aggregating and linking records which represent same real world entities [1], the real world entities could be patient, student, customer, company, bank…..etc. this issue had been investigated under different names like duplicate detection [2], object identification [3], object consolidation [4], entity resolution [5].

Manuscript received June 5, 2011
Manuscript revised June 20, 2011

Many algorithms have been introduced to solve the record linkage problem [2]; some algorithms were designed for a specific domain and some other algorithms could be applied in any domain but in general all algorithms are based on five main approaches: distance based, rule based, machine learning based, active learning based, and token based.
Applying record linkage algorithms on Arabic data is a challenging task due to the unique features and characteristics of Arabic language. According to our research, little work has been done in this area. Ramzi & Ralph [6] proposed an efficient K-way sorting method for duplicate detection and applied it on Arabic data context but it did not focus on handling the special feature of Arabic language. Suleiman et al [7] have used the N-gram technique in Arabic text search but they conclude that the N-gram technique was not an efficient approach and raise a question on the performance. This is due to the lexical structure of Arabic language that most Arabic word variants include a high rate of infix structure; two words may have a very low similarity although they are different only in term of their infixes. In order to improve the results of the N-gram technique, Suleiman et al proposed to combine the N-gram technique with a stemming technique. Moawia et al [8] developed a model for Arabic soundex function for Arabic name using Fuzzy logic. The proposed function shows good result in some patterns of Arabic names. The initial work covered the three letters base stem and then some enhancements have been done to cover the four letters base stem. The Arabic soundex function gives high attention to phonetic similarity and doesn't consider character form similarity or keyboard distance effect and this may give inaccurate results in some cases.
In this paper we propose a token based framework for record linkage in Arabic data set. The framework can be used for databases and data warehouse systems. A lot of token based techniques have been introduced to handle record linkage issue such as [9, 10, 11] the main difference between those techniques and our technique is that our technique uses new methodology for Arabic tokenization based on Arabic character form similarity, also we have used a new approach for similarity computation in order to minimize number of semi matched records.

The paper initially gives an overview about the main features of Arabic language and then in section 2 we discuss the various sources of duplication in Arabic context. In Section 4 describes the framework design and finally section 5 is the last section which concludes our work and states the future directions.

2. Arabic Language Features

There are three main types of Arabic language; classical, modern standard, and colloquial or dialectal Arabic [12]. The classical Arabic is the liturgical language of Islam since its inception in the 7th century. Classical Arabic uses vowel diacritics in the Qur'an, classical poetry, classical books…etc. The modern standard Arabic is derived from classical Arabic and it is widely used in workplaces, government and the formal media such as news channels. The modern standard Arabic rarely use diacritics, it may use it to differentiate between some similar words. Colloquial or dialectal Arabic refers to many national or regional varieties which constitute the everyday spoken language. Colloquial Arabic has many different regional variants; these sometimes differ enough to be mutually unintelligible and some linguists consider them distinct languages. They are often used in informal spoken media, such as talk show as well as occasionally in certain forms of written media, such as poetry and printed advertising. In general, word length in Arabic language is shorter than other Latin based languages. Average length of Arabic words is three or four character and it is very rare to find a word with more than eight characters but this is not the case in Latin based languages, for example the Arabic word "جامعة" is equivalent to "University" in English and "Université" in French and "Universität" in German. We can derive a lot of Arabic words from short roots; for example we can start from a three characters root such as "كتب" to derive a lot of words such as "كتاب", "كاتب", "مكتوب", "مكاتبة", "مكتب", "مكتبة"

3. Sources of Duplicates in Arabic Language

In general, duplicate data may occur due to many reasons [13] such as incorrect data entry, erroneous applications, and data obsolescence. In the following we give more attention to inaccurate reading, phonetic and hearing, keyboard distance, editing, dictation and morphological issues:-

- Inaccurate Reading: Inaccurate reading may occur when data entry operator reads a hand writing forms, Arabic character forms may differ based on writing style (Naskh, Requaa….etc) due to the variety of writing styles in Arabic language this may lead to inaccurate reading especially for some characters which contains dots or hamza. Some examples of this type are: الدهبى Vs الذهبى and اسما Vs أسماء
- Phonetic and Hearing issues: In many cases data entry operator inputs the required data based on dictation from the speaker and due to incorrect pronunciation from speaker's side and maybe due to some hearing problems from operator's side this will lead to duplicate data. Examples of this kind are: زكى Vs ذكى and ثناء Vs سناء
- Keyboard distance Issues: Due to the ergonomic design of Arabic keyboard and experience of data entry operator the type of errors may occur. The data entry operator may press wrong key by shifting his finger to the left or to the right also he may drop some characters due to their difficult location on the keyboard like Al-Zal "ذ" this letter would be replaced by Al-Dal "د". Examples of this kind are: ابوقير Vs الوقير and الخواجة Vs الهواجة
- Editing Issue: The data entry operator may press any character twice, insert additional characters, drop some characters, neglect the spaces between words, and also substitute two characters. Examples of this kind are: مدينةنصر Vs مدينة نصر and صلاح Vs صللاح
- Dictation and Morphological Issues: these kinds of issues occur from low level educated operators and also from non native speaker of Arabic language. Most of these issues are related to position of al hamza, using of alif al made and al alif al maqsoura, using of al taa al marboota, adding or omitting of long vowels (alif, yaa, waw) …etc. Some examples of this category are: مراد Vs موراد and ياسين Vs يسين and فدوا Vs فدوى

4. Framework Design

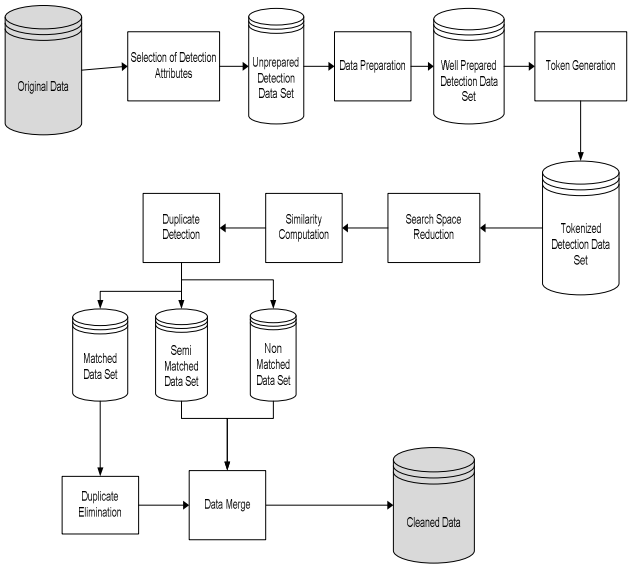


Fig.1 Framework Design

The framework consists of seven phases: selection of detection attributes, data preparation, token generation, search space reduction, similarity computation, duplicate detection and elimination, and finally data merge. The framework illustrated in the below diagram. In the following sections we will discuss each phase in more details.

4.1 Attributes Selection Phase

Record linkage process is usually done on a large data set which may contain hundreds of attributes and millions of records, of course it will be a time consuming task if we consider all attributes when measuring similarity between any pairs of record, for this reason the initial phase in this framework is selection of set of identification attributes. The identification attributes are those attributes which uniquely identify any record in the data set.

Selection of identification attributes require time and effort because data warehouse usually contains hundreds of column also we may find some degree of redundancy and inconsistency because different names may be used to represent same attribute and on the other hand same name may be used to represent different attributes. In this framework we depend on the automatic selection algorithm [14] which consists of four main steps:

Step 1: compute Identification Power $\Omega(x)$ for each attribute which indicates how much the values of a record attribute allow for distinguishing the record itself from others. For example SEX attribute has a very low identification power because it has just two distinctive values either male or female but on the other hand, home phone attribute has a high identification power because it has many distinctive values. $\Omega(x) = \frac{S(x)}{N}$ where S(x) is the number of distinctive values for attribute x and N is number of records.

Step 2: compute an index for data quality Ψ based on data quality dimensions accuracy, consistency, and completeness. Index of Data Quality for Attribute x is

$$\Psi(x) = \frac{\alpha \boldsymbol{Compl}(x)+\beta \boldsymbol{Accur}(x)+\boldsymbol{\gamma Cons}(x)}{\alpha+\beta+\boldsymbol{\gamma}} \quad (1)$$

Where α, β, and γ are the weight of the quality dimensions completeness, accuracy, and consistency respectively.

Step 3: calculate overall identification key IK for attribute x by multiplaying the identification power by the index of data quality dimensions as follow:

$$IK_j = \Psi(x).\ \Omega(x) \quad (2)$$

Step 4: select identification attributes that have max identification key max(IKj).

4.2 Data Preparation Phase

Data preparation process will be applied to eliminate any noise and unnecessary characters or words that may add additional cost on token generation or similarity measurement processes. Our main focus is to remove any special characters such as #) % $ (^ /] > [< { { …..etc., remove Arabic titles that may be added person's name attribute such as الشيخ – المهندس – الاستاذ, remove special words that may add some additional cost while comparing two strings for example while comparing two addresses some words like: منطقة – حى – محافظة make significant different while measuring the similarity between two string while those words should be neglected since many people do not pay too much attention to them while writing the address, Unify all date format to be ddmmyyyy. For example 12 Jun 2008 and 12/06/2008 will be changed to 12062008.

4.3 Token Generation Phase

The objective of token generation phase is to formulate short tokens that represent the detection attributes and use those tokens for measuring the similarity instead of the original attributes because short token will require fewer comparisons [9, 10]. In the following sections we illustrate how to generate tokens from alphabetic and alphanumeric attributes.

4.3.1 Token Generation from Alphabetic Attributes

The previous token based techniques such as [9, 10] generate short tokens by extracting the first character from each word because English names may use just one character instead of using the full name, for example "John

Smith Tom" vs. "J S Tom" but this is not the case in Arabic language because it usually depends on the full name. In our framework the token is generated by extracting the distinct letters from the alphabetic attribute. We will extract the distinct basic form of Arabic letter that does not consider the dots or Hamza because depending on the basic form of Arabic letters will reduce the effects of the typographical errors. The basic forms of Arabic letters are: ا instead of {آ – إ - أ - ء}, ب instead of {ي – ن - ث - ت}, ح instead of {خ – ج}, د instead of {ذ}, ر instead of {ز}, س instead of {ش}, ص instead of {ض}, ط instead of {ظ}, ع instead of {غ}, ف instead of {ق}, ه instead of {ة}, و instead of {ؤ}, ى instead of {ئ – ي if it came at the end of any word} in addition to {ك}, {ل}, {م}.

For example if customer name is "محمد أحمد مصطفى كامل" then there are 10 distinctive characters and the generated token will be "محداصطفىكل". Another example if customer name = "آثار طايل مكرم السحماوي" has 4 non basic forms but 2 of them are in the same group so they will be replaced by BAA, the generated token is "ابرطلمكسحوى".

4.3.2 Token Generation from Alphanumeric Attributes

The most difficult attributes to be handled is the alphanumeric attributes especially if there is no standard method to construct such attribute. There is no standard method to write an alphanumeric attribute like the address attribute. We can find too many combinations for same address like "شارع سعد زغلول – عمارة 18 – شقة 21" and " ش 18 سعد زغلول – شقة 21".

We will generate two tokens from the alphanumeric attributes one for the numeric part and the other one for the alphabetic part. This can be done through four steps. Step 1: extract the alphabetic part from the attribute and arrange it in ascending order. Step 2: construct numeric token. Step 3: extract the numeric part from the attribute and arrange it in ascending order. Step 4: construct alphabetic token as the previous section.

To illustrate the previous procedure let us consider an example of customer address like " عمارات امتداد رمسيس 2 - عمارة 150 - شقة 33 - مدينة نصر -محافظة القاهرة". The address after removing any special words in the preparation phase will be "عمارات امتداد رمسيس 2 150 شقة 33 مدينة نصر القاهرة". The sorted alphabetic part will be " امتداد القاهرة رمسيس شقة عمارات مدينة نصر". The alphabetic token will be "امبدلقهرسعص". The numeric token will be "233150"

4.4 Search Space Reduction Phase

The initial search space for the record linkage problem is R * S where R, and S are the data set to be compared. It is time consuming - especially for large data set – to deal with such space. In order to reduce the search space the data set will be sorted based on the selected fields. Three token tables will be generated each one of them will be sorted using one of the selected tokens. When sorting based on alphabetic tokens we should consider also the length of the original field and then the other two token fields according to their uniqueness power. The moving window will be used on each of the token tables to reduce the number of comparisons from n2 to O(wn) where w is the window size, considering sorting complexity of n log n and hence the total time complexity for the three tables 3* O(n log n + wn).

4.5 Similarity Computation Phase

The input of this phase is the tokenized data set which has been generated from the previous phase. In this phase we use a new technique for similarity computation which measures the similarity on three sequential stages. The first stage is token length comparison, second stage is token similarity measurement and the last stage is attribute similarity measurement which will be applied on alphabetic attribute only if the second stage did not give a clear decision. In the token length comparison stage we calculate token length ratio TLR using the following formula:

$$TLR(s, t) = \frac{|len(s) - len(t)|}{\max(len(s), len(t))} \qquad (3)$$

If TLR is lower than threshold x0 then tokens can't be similar and there is no need to do any more comparisons otherwise we should go for the second stage and compute similarity index based on token type as follow:

For numeric and date tokens, Hamming distance [15] will be used to compute similarity between numeric tokens and date tokens and then compare the output to user defined thresholds x3 and y3 as follow: if TS $\geq$ x3 then the tokens are matched, if y3 > TS then the tokens are not matched and if y3 $\leq$ TS < x3 then the two tokens maybe matched.

For Alphabetic token, Jaccard coefficient [16] will be used to compute token similarity index TS and then compare it to a user defined thresholds x1 and y1. If TS $\geq$ x1 then the tokens are similar, If y1 > TS then the tokens are not matched If y1 $\leq$ TS < x1 then the tokens are semi matched. If the tokens are semi matched then we should go for the third stage which is attribute similarity measurement. In this stage similarity measurement will be applied on the prepared attributes before tokenization using the edit distance algorithm [17]; the output of this step will be compared to user defined thresholds x2 and y2. If TS $\geq$ x2 then the attributes are matched, If y2 > TS then the attributes are not matched and If y2 $\leq$ TS < x2 then the attributes maybe matched.

4.6 Duplicate Detection & Elimination Phase

The first step in this phase is to calculate record similarity RS between two records R1 and R2 in the detection data set. Record similarity RS is the percentage of number of matched attributes n to the total number of attributes m in the detection data set. Record similarity RS will be compared to user defined threshold x4 and y4. If RS $\geq$ x4

then the two records are matched, If y4 > RS then the two records are not matched and finally If y4 ≤ RS < x4 then the records are semi matched.
The next step after the duplicate detection step is duplicate elimination. We should differentiate here between duplicate elimination in databases and duplicate elimination in data warehouse systems. In database systems only one copy of duplicate records are kept as a unique record and then remove other records. In data warehouse systems, we should do fact aggregation first and then duplicate elimination [10] so if duplicate records exist in one of the dimension tables then we will select one of the duplicate record and consider it as a unique record. Fact aggregation step starts by adding fact measures that are related to the duplicate records to the unique record measure. After that we will do duplicate elimination step by removing the duplicate records from the dimension table and their related records in the fact tables.

4.7 Data Merging Phase

In this phase the cleaned data will be merged as a single cluster. The user must maintain the merged records and the prime representation as a separate file in the data warehouse. Data merge step is very important for the incremental data cleaning process[10].When a new data entered into the data warehouse, incremental data cleaning compare the new data with the LOG file, which has been created in the previous cycle to detect the duplicate records. This approach guarantees an easy way for incremental data cleaning and reduces the data cleaning time.

5. Conclusion and Future Work

In this paper we presented a token based frame work for record linkage in Arabic data set. The framework consists of seven phases: selection of detection attributes, data preparation, token generation, search space reduction, similarity computation, duplicate detection and elimination, and finally data merge. In the token generation phase we introduced a new technique for Arabic name tokenization, the proposed technique depends on using the basic forms of Arabic letters (16 letters instead of 28 letters); this has speeded up the processing and reduced the complexity. In order to reduce number of semi matched records in the final results we used an additional stage of similarity computation.
There are a lot of directions that can be investigated in the future. for example: enhance the framework to be applicable on classical Arabic language, apply the token based technique on unstructured and semi structured Arabic data, use our framework to build an Arabic data cleaning tool that can be used for large Arabic data set such as census data in Arabic world.

References

[1] Liang Jin, Chen Li, and Sharad Mehrotra. Efficient Record Linkage in Large Data Sets. Proceedings of the Eighth International Conference on Database Systems for Advanced Applications (DASFAA'03). 2003.

[2] Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, and Vassilios S. Verykios. Duplicate Record Detection: A Survey. IEEE Transactions knowledge and Data Engineering, Vol. 19, NO. 1, Jan 2007.

[3] M. Neiling, S. Jurk, H.-J. Lenz, and F. Naumann. Object identification quality. In Intl. workshop on Data Quality in Cooperative Information Systems (DQCIS2003), Siena, Italy, 2003.

[4] Michalowski, M., Thakkar, S. & Knoblock, C. A. Exploiting Secondary Sources for Automatic Object Consolidation. In the Proceedings of the VLDB Workshop on Information Integration on the Web. 2004.

[5] David Guy Brizan & Abdullah Uz Tansel. A Survey of Entity Resolution and Record Linkage Methodologies. Communications of IIMA, Volume 6 Issue 3, 2006.

[6] Ramzi A. Haraty and Ralph Varjabedian. ADD: Arabic Duplicate Detector. A Duplicate Detection Data Cleansing Tool. Proceedings of the ACS/IEEE AICCSA Conference, Tunis July 2003.

[7] Suleiman H. Mustafa and Qasem A. Al-Radaideh. Using N-Grams for Arabic Text Searching. Journal of the American Society for Information Science and Technology— September 2004.

[8] Moawia E. Yahia, Soundex Function for Arabic Names Using Fuzzy Logic (in arabic), in proceedings of 5th international conference of computer science practice in Arabic, Morocco, 144-153, May/2009.

[9] C.I. Ezeife and Timothy E. Ohanekwu, Use of Smart Tokens in Cleaning Integrated Warehouse Data, the International Journal of Data Warehousing and Mining (IJDW), Vol. 1, No. 2, pp. 1-22, Ideas Group Publishers, April-June 2005.

[10] J. Jebamalar Tamilselvi. and V. Saravanan. A Unified Framework and Sequential Data Cleaning Approach for a Data Warehouse. IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.5, May 2008.

[11] Amira Elzeiny, Ali El-Bastawissy, Ahmed S. Tolba, Osman H. Mohamed "An Enhanced Smart Tokens Algorithm for Data Cleaning in Data Warehouses", Cairo University, Faculty of Computers and Information, Egypt, Proceedings of the 5th International Conference on Informatics and Systems, Mar, 2007.

[12] Information on http://en.wikipedia.org/wiki/Arabic_language

[13] Khaled Shaalan, Amin Allam, and AbdAllah Gomah. Towards Automatic Spell Checking for Arabic. Conference on Language Engineering, ELSE. Cairo. 2003.

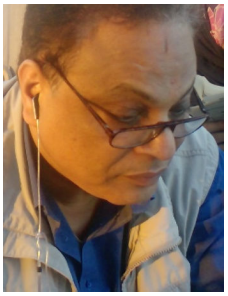
[14] Bertolazzi, P., De Santis, L., and Scannapieco, M. Automatic Record Matching in Cooperative Information Systems. In Proc. DQCIS ICDT Workshop Siena, Italy, 2003.

[15] Hamming, R. W, "Error detecting and error correcting codes". Bell System Tech. J, vol. 29, no. 2, pp.147–160, 1950.

[16] Paul Jaccard. Etude comparative de la distribution orale dans une portion des Alpes et des Jura. In Bulletin del la Socit Vaudoise des Sciences Naturelles, volume 37, pages 547-579, 1901.

[17] V.I. Levenshtein, "Binary Codes Capable of Correcting Deletions, Insertions and Reversals," Doklady Akademii Nauk SSSR, vol. 163, no. 4, pp. 845-848, 1965, original in Russian—translation in Soviet Physics Doklady, vol. 10, no. 8, pp. 707-710, 1966.

Hesham H. Abdel Ghafour received the B.Sc. degree in 1999 from Production Engineering Department, Alexandria University, Egypt. In 2002, Hesham obtained a diploma degree in Software System Development from ITI Information Technology Institute, Cairo, Egypt. Currently, he is a master student at Colleague of Computing and Information Technology at AAST Arab Academy of Science and Technology, Cairo, Egypt. His research interests include: Data Quality, Master Data Management, Data Cleansing, Data Integration, Data Warehouse, Data Modelling, Business Analysis, and IT Service Management.



Ali H. El-Bastawissy Professor of Information Systems - Cairo University – Egypt has about 30 published articles in the area of Data Engineering and Management, Analytics, Data Integration and Virtualization. A senior IT Consultant in Egyptian Cabinet, Ministries, and other Arab Countries Director of the Center of Studies and Development of Computers and Information Systems – Cairo University.

Abdelfatah A. Hegazy received the B.E. degrees, from the Military Technical Collage, Cairo, Egypt, 1978. In 1982 he received the M.Sc. in Computer Sciences from George Washington University, USA. Dr. Hegazy received the Ph.D. Degree Computer Sciences from George Washington University, USA, in 1985. After working as an assistant professor (from 1985) in the Dept. of Computer Engineering and Operations Research, the Military Technical Collage, and an associate professor (from 1990), he has been a professor at College of Engineering at the Arab Academy for Science and Technology, since 1998. His research interest includes: Information Systems Planning; E-Commerce, E-Government, Information Systems Security, Network Security, Knowledge Management, Web Intelligent Systems and Enterprise Resource Planning Systems. He is a member of IEEE, ACM, AIS, AANIS, and CSS Computer Scientific Society Egypt.