A hybrid Technique for Cleaning Missing and Misspelling Arabic Data in Data Warehouse

Mohammed Abdullah Al-Hagery
Department of Computer Science, College of Computer, Qassim University, KSA
E-mail: hajry@qu.edu.sa, drmalhagery@gmail.com

Latifah Abdullah Alreshoodi, Maram Abdullah Almutairi, Suha Ibrahim Alsharekh, Emtenan Saad Alkhowaiter
Master Students, Department of Computer Science, College of Computer, Qassim University, KSA
E-mail: La.Alreshoodi@qu.edu.sa, maram.a.almutairi@gmail.com, soha.e.sh@gmail.com, e.alkhowaiter@qu.edu.sa

Received: 26 June 2018; Accepted: 19 May 2019; Published: 08 July 2019

Abstract—Real-World datasets accumulated over a number of years tend to be incomplete, inconsistent and contain noisy data, this, in turn, will cause an inconsistency of data warehouses. Data owners are having hundred-millions to billions of records written in different languages, hence continuously increases the need for comprehensive, efficient techniques to maintain data consistency and increase its quality. It is known that the data cleaning is a very complex and difficult task, especially for the data written in Arabic as a complex language, where various types of unclean data can occur to the contents. For example, missing values, dummy values, redundant, inconsistent values, misspelling, and noisy data. The ultimate goal of this paper is to improve the data quality by cleaning the contents of Arabic datasets from various types of errors, to produce data for better analysis and highly accurate results. This, in turn, leads to discover correct patterns of knowledge and get an accurate Decision-Making. This approach established based on the merging of different algorithms. It ensures that reliable methods are used for data cleansing. This approach cleans the Arabic datasets based on the multi-level cleaning using Arabic Misspelling Detection, Correction Model (AMDCM), and Decision Tree Induction (DTI). This approach can solve the problems of Arabic language misspelling, cryptic values, dummy values, and unification of naming styles. A sample of data before and after cleaning errors presented.

Index Terms—Data Cleaning, Missing Data, Arabic Misspelling, Data Quality, Data Consistency.

I. INTRODUCTION

Quality improvement in Data has become a critical issue for several companies and organizations because low data quality degrades organizational performance whereas enhanced data quality results in cost saving, customer satisfaction, and better analysis that will lead to a good Decision-Making. There are a lot of researchers developed various data cleaning methods of variant types of errors to improve data quality.

Data cleaning starts with identifying the expected problem when integrating various sources of data. A number of algorithms and techniques have been proposed and examined to improve the quality and consistency of the data warehouse, which is a central repository for integrated data. A real data tends to be incomplete, inconsistency, and noisy. These become problems of the data warehouse. Therefore, a comprehensive method is required to solve such problems.

There are many solutions to keep the data warehouse consistency, but all that solutions are limited to specific problems. In data warehouses, data cleaning is the main part of Extraction, Transformation, and Loading (ETL). Data cleaning helps data specialists to determine and detect all inconsistent and incomplete data items [1]. The automate data cleaning provides a solution to find and solve some of these problems and make datasets partially on a consistency level. Hence, there are many techniques such as Decision Tree algorithm that is handling missing values, it can be merged with other techniques to solve the most of data errors.

The proposed approach is an extension to improve the smart technique presented in [2]. Which focused on the cleaning of some errors of datasets, such as redundant values, error formats, numeric errors of values, and Arabic mistakes in sentences. Many techniques that have applied to solve common problems and specific kinds of errors still limited. Consequently, we proposed a hybrid method that can deal with many issues, such as Arabic language misspelling, missing data, cryptic values, dummy values, and unification of naming style problem, where one attribute like department name can hold different naming styles.

The paper is organized as follows: Section 2 describes the various methods and techniques used for data cleaning and corrections. Section 3 describes the research methodology, it explains the different steps of
the proposed approach and ended with a proposed algorithm. In section 4, the results are discussed, while section 5 presents the conclusions. Finally, section 6 describes the future work.

II. RELATED WORKS

Data into the data warehouse usually collected from different sources that contain errors, missing values, contradicting data, cryptic data, noisy data, misspelling errors, etc. Currently, many types of research works focus on data cleaning from many types of errors to maintain data consistency and enhance its quality. There are many techniques to deal with missing values. Some techniques deal with missing values by deleting the records that had these values. However, deleting these records of a small dataset generally reduces the sample size and its usability for several statistical analyses. Moreover, the statistical analysis results getting from a data set having an insufficient number of records are not strong, because of incomplete and noisy data deletion. In the following, we will discuss common methods of this regard. These methods are employed to fill in missing data using the median, mode or mean.

In terms of knowledge discovery from data, the application of the mean replacement method can often lead to more misleading results than the method of simple record deletion [3]. The transitive closure algorithm employed in data cleaning to find the duplicates in datasets explained in detail [4]. A transitive closure infilling method of missing values, removing redundancies of data items and the combination of similar data items together discussed [5]. Wang et al. analyzed a set of problems relevant to data cleaning illustrated by the results [6]. As well, a hybrid method developed to clean data using enhanced versions of two basic techniques namely PNRS and Transitive Closure explained in [7]. On the other hand, an educational data mining field a special system had developed and examined by the Decision Tree to solve the educational problems of datasets [8]. The missing value is one of the most problems found, so researchers use some common algorithms to solve this problem such that ID3, CART, and C4.5.

Likewise, a simple technique was proposed to handle and copy of missing values by using the Decision Tree technique. This technique called messiness in the MIA and the EMMI algorithm used for the same purpose for range data type then copying in missing values [9]. Yang et al. discuss the test costs method to query missing values, by applying the Test Sensitive Classification Learning (TSCL) algorithm to minimize the test and classification cost [10]. Also, some imputation techniques such as the K-Nearest Neighbor, Mean-Mode, Hot_Deck, Expectation Maximization, and C5.0 used to impute artificially created missing data from various datasets. The techniques performance was compared based on the accuracy of the classification of the original data and the imputed data. Identifying useful imputation technique provided accurate results of classification as presented [11]. A number of solutions were provided to clean cryptic, dummy values, and contradicting data, which represent the most values, in this regard. For instance, Swapna et al. assume some dummy values before comparing with the database values. Whenever Data mining query finds this assumed value, it directly knows that it is a dummy value [12]. In the same direction, there has been a remarkable series of tasks for capturing the general errors of data as violations of data integrity restrictions as accomplished [13-17]. Many data repairing methodologies have been proposed, developed, and applied [18-25]. A set of heuristic algorithms were developed for data preprocessing [18,20], based on functional dependencies [17,26], involved in inclusion dependencies [18], Conditional Functional Dependencies (CFDs) [27], the CFDs, Matching dependencies [22], and denial constraints [19].

Some research works employ confidence values placed by users to guide a repairing process [18,20,22] or use master data [28]. The statistical inference process is proposed to derive missing values [24]. In order to ensure the quality and the accuracy of data repairs required consulting professionals or users [24,25,28].

In more specific, the spelling errors detection and correction in the English Language mostly investigated, many types of research are dealing with the English language errors. On the other hand, there is a limited set of approaches were applied to solve this problem the Arabic language because it is very complicated, where [29] established an error detection and candidate generation model to detect the misspellings, and generate their possible candidates then design a misspelling correction system. Furthermore, a hybrid system based on the confusion matrix and Noisy Channel Spelling Correction Model developed to detect and correct Arabic spelling errors [30]. A spelling error detection and correction method targeted at non-native Arabic language learners evaluated by a team of specialists. They applied edit distance algorithm and rule-based transformation approach. Rules are formulated according to their study on common spelling errors made by Arabic learners to handle the noisy data [31,32]. However, these approaches are partially different from our proposed approach, where, the most of them were concentrated on limited problems and used different mechanisms.

The Arabic edited distance algorithm proposed to measure the similarity between two Arabic strings. This type of algorithm is working based on a Levenshtein algorithm [32], likewise, a system developed for ‘Arabic character’ recognition using guessing the Fast Fourier Transform descriptors for counting the basic part of characters presented by [33]. The system used ten features to classify the character models. Also, Higazy et al. had developed an English/Arabic enabled web-based framework, considering the wide range of variations in the Arabic language to allow the fast processing, improved indexing/blocking techniques are
used [34]. There is a set of cleaning tools available for data cleaning. It concentrates on specific fields, such as cleaning address and name or a specific attribute to remove the duplication. Also, it is restricted to a specific domain, these tools usually execute very well but must be integrated with other tools to fix most of the cleaning problems that affecting the data integration.

In addition, Hamad and Jihad developed an enhanced method works on the identification of error corrections in data. Such as domain format error, lexical mistakes, constraint violation, irregularities, integrity, and redundant values [36].

Other tools developed for Extract, Transform and Load (ETL), which have comprehensive transformation and workflow capabilities help to transform and load then clean raw data. These tools developed to cover the most of the data transformation and cleaning tasks. There is a common problem with these tools that is the difficulty to merge the functionality of various tools together [3,37].

A set of Key elements discussed in a survey for cyberspace [6]. An automated technique provided for proof negative and positive during data cleaning, based on reference tables and Sherlock Rules to update them when enough information is available. The Sherlock Rules methods are able to repair and annotate un-cleaned instances in a deterministic fashion with high precision [38]. Data cleaning also improved and it puts forward a method named an association mining data cleaning [39].

In addition, Al-Hagery et al. had developed a software tool to clean, integrate, and transform the content of a big dataset of the Hepatitis Disease written in the English language [40]. Furthermore, a smart algorithm proposed to identify the most data errors including errors in data formats, redundant data, data duplicity, limited Arabic language mistakes, numeric errors, and symbol errors. However, it did not consider some type of errors in the data, that is why we propose our method [41].

III. RESEARCH METHODOLOGY

The proposed technique is similar to the smart technique presented by [41]. It is focusing on the Arabic language misspelling because there are no more perfect techniques to deal with this subject. This Technique planned to work on the data gathered from multiple sources before the loading to the data warehouse. This technique tested and validated using a set of real data instances. Fig.1 illustrates the headlines of the research methodology. The methodology includes; Arabic misspelling detection, misspelling correction, fill missing data, cryptic values, dummy values, unification of naming styles, and development of a hybrid algorithm to clean missing and misspelling arabic data.

The data set used in this paper consists of seven attributes as shown in Table 1. The data sample used before processing is presented in Table 2 while Table 3 will contain the data after processing. The Students’ Names placed in Arabic and in English (STD_Name_AR and STD_Name_EN), were included in these tables.

A. Arabic Misspelling Detection and Correction

The structure of AMDCM used to solve the Arabic Misspelling problem is presented in Fig.2, the steps of this approach will be discussed in the following sections:

1. Misspelling Detection: The spelling error detection is a process of detecting incorrect word/words (non-word) using different detection methods. The most common technique is the N-grams algorithm. This algorithm is an N letter subsequence of a string, where N is usually mono, bi or tries to determine if each gram in an
input word is expected to be valid or invalid in the language. Some techniques use two-level morphological analysis to check whether an input value follows those language morphological rules. In addition, some machine-learning algorithms as hybrid techniques are using for Misspelling Detection. However, the most widely used technique in many applications for Misspelling Detection is the dictionary lookup technique, where the input word is compared with the words in the dictionary, if the input word is not found in the dictionary, the word is considered as an incorrect word. This technique is unlike the morphological analysis, it guarantees that the most common words used are covered, but it will not provide a complete coverage of the language. In our approach we used the dictionary lookup techniques to detect Arabic Misspelling errors since this technique can handle the most common words used, faster than others, it is a language independent. In this model, the size of our proposed dictionary is about 427 K unique words, where these words organized in the dictionary in an ascending order to speed-up the comparison using a search algorithm. All words that begin with the same letter will be placed in a separate table, to improve the comparison.

2. Misspelling Correction: Techniques used for spelling errors correction, it can be classified into context-dependent and context-independent error corrections. In this paper, our focus is on context independent error correction techniques, which perform in an isolated word or words. In order to propose an efficient misspelling correction model for the Arabic language, we significantly need to study and categorize common error patterns. Fig.2 shows the steps and the chart of AMDCM.

The Misspelling Correction includes the following tasks:

- **Single-word Errors:** The misspellings appear because of omitting, adding, replacing or duplicating a letter within a given word by mistake. For example, spelling the word 'سلام - Saleem', in this case, misspelling happened by duplicating the letter 'ي'. Also, may omit a letter identical to 'ب', cause the misspelling such as: 'مروتة' instead of 'مروتة' - muratab' and adding an extra letter 'ن' in a word as 'خريف' instead of 'خريف' - khareef', or by replacing a letter in a word as 'كدير' in place of 'كيبر - kabeer'.

For these types of errors, several techniques and algorithms can be used together to perform a context-independent error correction such as rule-based, N-gram, probabilistic, and neural network techniques. According to [31] a rule-based technique represents common spelling error patterns in the form of rules, which are used to transform misspelt words with the correct word. A neural network technique performs associative recall based on the incomplete input. A probabilistic technique works based on N-gram technique, which is used either as a standalone or with other techniques to perform error correction. Also, the similarity key technique can employ to transform words with keys as applied [29]. Therefore, computing a key to misspelling word will point to similarly spell words of the dictionary.

The most common technique used for misspelling correction is the edit distance, which use the dynamic programming to find out the lowest cost of editing operations (insert, delete, and replace) in order to transform string \( Str_1 \) into new string \( Str_2 \) by giving a cost value of edit operations [32]. As mentioned in the example given by [41], to correct the word 'كدير' to 'كيبر kabeer' based on the Arabic dictionaries and unifying the text into standard word. The edit distance is a more efficient technique for calculating the distance between
the incorrect, the correct word in the dictionary and select the minimum cost word. In that example, we measure the distance between the previous two Arabic words, as shown in Fig.3, the edit distance technique deletes the second letter in that word, which is 'ي-yaa'. The deletion operation of one letter in this technique is costing one.

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Fig.3. Edit Distance

- Space Deletion Errors: When the user/editor forgets to put spaces between words that caused the space deletion errors. A sequence of Arabic letters was selected with the maximum marginal probability of A* lattice search, N-gram probability estimation, and using 15-grams language model of Arabic letters. This technique is more suitable for long Arabic texts because it will reduce the complexity of processing [29]. However, our exhaustive search model is more appropriate because the field of the search will not contain more than 3-4 words so, merging these words will not be much complicated.

Correction of the deleted spaces or adding spaces in the word and try to divide it into S_i, correctly spelt words. For example, the STD_Department column contains the value 'علوم الحاسب' as a single entry, the dictionary will classify it as an incorrect word. In this case, an exhaustive search process will find all possible correct words S_i, where S_i<1 from the dictionary, then the sequence of letters of 'علوم الحاسب'. It will be spelt into two words 'علوم' and 'الحاسب', by insertion of the deleted space.

- Space Insertion Errors: When the user adds one or more spaces in a single word that causes the space insertion errors. The proposed algorithm [29] can be used to find the possible merging suggestions for an incorrect phrase. In our model, we will use the exhaustive search as in the correction of space deletion errors (discussed above) by trying to concatenate S_i incorrect words sequence to find all possible correct words of the dictionary. For example, if the user enters in the STD_Department name column, in Table 2, this string consists of two different names of a student; the first name and the second name: 'سليمان' and 'محمد'.

B. DTI Algorithm to Fill Missing Data

The DTI algorithm applied [12] to fill the missing data for each data attribute, there is one binary tree that will be generated. Each Decision Tree involves a set of nodes and each node is a test on that attribute. Two paths will come out from each value/node except for the end leaf nodes. The first path is “No” and the second is “Yes”. The missed values are placed in the Leaf Nodes.

The missing data is replaced by the data placed in the leaf nodes of the tree. Initially, data in the form of rows are collected from different educational data sources. Each row is a combination of several attributes.

Fig.4 illustrates a Decision Tree for a set of students, it presents how Decision Trees is created for STD_Department attribute values, it checks the value of each row until it reaches to the missing value then replaces it with the leaf Node value. Likewise, a Decision Tree is used to fill in the missing data by the value found in the leaf node. The Decision Tree was chosen because it is simple, fast, and it does not require any domain knowledge.
C. Cryptic Values

A simple example of Cryptic Values is the Cumulative Grade Points Average (CGPA). If the student fills his/her CGPA as a 4.6, the cleaning process should convert this value to the percentage, because most of the universities prefer percentages than CGPA [12]. In our example, if the user enters the CGPA such as 4.50 out of five, it will be converted to percentages by the division of five then multiplied the result by 100.

D. Dummy Values

Usually, the dummy values are those values generated in a huge range on the Internet. For example, Email account registrations, other accounts in Tweeters, Facebook, WhatsApp or other applications. Currently, there is no existing algorithm to find the dummy values. Each field follows a special method to identify and solve the dummy values. The following are a set of examples given as dummy values.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
<th>Field Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>STD_Name_AR</td>
<td>Student Name in the Arabic language</td>
<td>String</td>
</tr>
<tr>
<td>STD_Name_EN</td>
<td>Student Name in the English language</td>
<td>String</td>
</tr>
<tr>
<td>STD_Age</td>
<td>Student Age</td>
<td>Number</td>
</tr>
<tr>
<td>STD_Department</td>
<td>Student Department</td>
<td>String</td>
</tr>
<tr>
<td>STD_GPA</td>
<td>Student Grade Point Average</td>
<td>Number</td>
</tr>
<tr>
<td>Nationality</td>
<td>The nationality of Student</td>
<td>String</td>
</tr>
<tr>
<td>Gender</td>
<td>Male or Female</td>
<td>String</td>
</tr>
</tbody>
</table>

Table 2, in the column of STD_CGPA.

4) Process the Arabic Language Misspelling, check all words based on lookup dictionary technique, in case of misspelling errors, and the AMDCM model will correct the misspelling through several techniques based on the following error (Single Word Errors, Space Deletion Errors, Space Insertion Errors).

5) For each Missing Data, use the DTI to fill the required value.

6) For Cryptic Data values, check the field of this attribute then convert the values to the preferable

Table 1. Field Description of Student Data Set

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
<th>Field type</th>
</tr>
</thead>
<tbody>
<tr>
<td>STD_Name_AR</td>
<td>Student Name in the Arabic language</td>
<td>String</td>
</tr>
<tr>
<td>STD_Name_EN</td>
<td>Student Name in the English language</td>
<td>String</td>
</tr>
<tr>
<td>STD_Age</td>
<td>Student Age</td>
<td>Number</td>
</tr>
<tr>
<td>STD_Department</td>
<td>Student Department</td>
<td>String</td>
</tr>
<tr>
<td>STD_GPA</td>
<td>Student Grade Point Average</td>
<td>Number</td>
</tr>
<tr>
<td>Nationality</td>
<td>The nationality of Student</td>
<td>String</td>
</tr>
<tr>
<td>Gender</td>
<td>Male or Female</td>
<td>String</td>
</tr>
</tbody>
</table>

Table 2. A sample of Dataset for Students in Computer College Before Cleaning

<table>
<thead>
<tr>
<th>ID</th>
<th>STD_Name_AR</th>
<th>STD_Name_EN</th>
<th>STD_Age</th>
<th>STD_Department</th>
<th>STD_CGPA</th>
<th>Gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ahmed</td>
<td>Ahmed</td>
<td>24</td>
<td>22</td>
<td>83.0</td>
<td>دكر</td>
</tr>
<tr>
<td>2</td>
<td>Bader</td>
<td>Bader</td>
<td>22</td>
<td>20</td>
<td>4.45</td>
<td>دكر</td>
</tr>
<tr>
<td>3</td>
<td>Youssef</td>
<td>Dalal</td>
<td>21</td>
<td>23</td>
<td>60.5</td>
<td>أنثى</td>
</tr>
<tr>
<td>4</td>
<td>Dalal</td>
<td>20</td>
<td>24</td>
<td>23</td>
<td>3.15</td>
<td>أنثى</td>
</tr>
<tr>
<td>5</td>
<td>Saad</td>
<td>Omar</td>
<td>21</td>
<td>22</td>
<td>5.00</td>
<td>null</td>
</tr>
<tr>
<td>6</td>
<td>Omar</td>
<td>Dalal</td>
<td>23</td>
<td>21</td>
<td>82.0</td>
<td>دكر</td>
</tr>
<tr>
<td>7</td>
<td>Faten</td>
<td>Dalal</td>
<td>22</td>
<td>22</td>
<td>72.5</td>
<td>أنثى</td>
</tr>
<tr>
<td>8</td>
<td>Suliman</td>
<td>Saad</td>
<td>24</td>
<td>24</td>
<td>84.8</td>
<td>دكر</td>
</tr>
<tr>
<td>98</td>
<td>Rahf</td>
<td>Saad</td>
<td>24</td>
<td>24</td>
<td>4.66</td>
<td>أنثى</td>
</tr>
</tbody>
</table>

F. The proposed Algorithm

The following algorithm is summarizing the mechanism of the hybrid method for data cleaning depend on various techniques.

1) Start
2) Use the dataset (table/tables) collected from the Original Sources $S_1$, $S_2$, ..., $S_n$
3) Chick the dataset field by field, identify the error types
4) Process the Arabic Language Misspelling, check all words based on lookup dictionary technique, in case of misspelling errors, and the AMDCM model will correct the misspelling through several techniques based on the following error (Single Word Errors, Space Deletion Errors, Space Insertion Errors).
5) For each Missing Data, use the DTI to fill the required value.
6) For Cryptic Data values, check the field of this attribute then convert the values to the preferable
range

7) For Dummy Values, check the data in this field if
it’s dummy value by comparing it with the assumed
dummy values such as ‘...’ then replace
it with the “Dummy” word.
8) For all incompatible naming styles, use the most
common name in that field to unify all names.
9) Save the cleaned data and exit.
10) End.

IV. RESULTS DISCUSSION

When integrating data gathered from multiple sources, the
noisy data appears and the needs for data cleaning are
increased. In this paper, the proposed approach applied to
a sample of data contains 98 data records (presented in
Table 2) to test the validation of the proposed approach.
this sample contains data before the cleaning process.
It contains various types of errors occurred during data
entry, such as missing letters, missing some fields, extra
space ‘_’ (column 2, row 1 and column 2, row 4), these items converted into Table 3 to
related to adding or deleting space respectively. Moreover, there is
another type of errors was discovered and corrected, it is
the doubling of one letter, as in the content of row 5 and
row 98 (field of STD_Name_AR).

In addition, some words there is Arabic Misspelling
related to adding or deleting spaces in a single words or
may omit a letter in a word or words, these problems were solved by adding the omitted letter or by inserting
of a space in the required position, as shown in the final
result (Table 3, STD_Name_AR row 6 and 7).

Additionally, various naming styles found in the
dataset sample, as in column "STD_Department", which contain a set of variable names with different naming
styles in the following list:

<table>
<thead>
<tr>
<th>ID</th>
<th>STD_Name_AR</th>
<th>STD_Name_EN</th>
<th>STD_Age</th>
<th>STD_Department</th>
<th>STD(CGPA)</th>
<th>Gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ahmed Khaled</td>
<td>'Dummy'</td>
<td>24</td>
<td>علم حاسب</td>
<td>83.0%</td>
<td>ذكر</td>
</tr>
<tr>
<td>2</td>
<td>Bader Ahmed</td>
<td>علم الحاسب</td>
<td>89.0%</td>
<td>ذكر</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Yousef Sayed</td>
<td>علم الحاسب</td>
<td>60.5%</td>
<td>تثنى</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Dalal Ibrahim</td>
<td>علم الحاسب</td>
<td>63.0%</td>
<td>تثنى</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Saud Khalil</td>
<td>تكنولوجيا المعلومات</td>
<td>100%</td>
<td>تثنى</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Omar Qamar</td>
<td>علم الحاسب</td>
<td>82.0%</td>
<td>ذكر</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Faten Sami</td>
<td>هندسة البرمجيات</td>
<td>72.5%</td>
<td>ذكر</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Suliman Mohammed</td>
<td>علم حاسب</td>
<td>84.8</td>
<td>ذكر</td>
<td></td>
<td></td>
</tr>
<tr>
<td>98</td>
<td>Rahf Nasser</td>
<td>هندسة البرمجيات</td>
<td>93.2%</td>
<td>ذكر</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In Table 3, the results of cleaning data have shown
in this list:

Table 3. A sample of Data for Student’s Information in Computer College, After Cleaning

V. CONCLUSION

One of the most important steps of data processing is
the verification of data values are correct. Data cleaning
is an important task for data warehousing development.
In this paper, we proposed a comprehensive approach
includes most of the data cleaning methods, this
approach is able to check and repair the Arabic words
that have misspelling, through the DIT model to fill in
missing data and AMDCM model to correct the
misspelling based on error type: Single Word Errors,
Space Deletion Errors and Space Insertion Errors, etc.
Moreover, this approach can solve the problems of
cryptic values, dummy values, and unification of
various naming styles. A sample of data has presented
before and after errors cleaning to show the validation

Copyright © 2019 MECS
The proposed approach can be modified according to the following perspectives:

1. Increase the size of the Lookup Techniques Dictionary to cover all possible cases.
2. Develop this approach as a software application, as an implementation of the algorithm of the proposed approach to be applied on big datasets.
3. Improve this approach to cover multilingual, exceeded the Arabic and English languages.

REFERENCES

A hybrid Technique for Cleaning Missing and Misspelling Arabic Data in Data Warehouse

Mohammed Abdullah Al-Hagery: received his B.Sc in Computer Science from the University of Technology in Baghdad Iraq-1994. He got his MSc. in Computer Science from the University of Science and Technology Yemen-1998. Al-Hagery finished his Ph.D. in Computer Science and Information Technology, (Software Engineering) from the Faculty of Computer Science and IT, University of Putra Malaysia (UPM), November 2004. He was a head of the Computer Science Department at the College of Science and Engineering, USTY, Sana'a from 2004 to 2007. From 2007 to this date, he is a staff member at the College of Computer Science, from the College of Computer Science and Technology, Qassim University, KSA. He published more than 20 papers in various international journals. Dr. Al-Hagery was appointed a head of the Research Centre at the Computer College, and a council member of the Scientific Research Deanship Qassim University, KSA from September 2012 to October 2018. Currently, he is teaching the master degree students and a supervisor of four master thesis. He is a jury member of a number of PhD and master thesis, as an internal and external examiner in his field of his specialist.

Latifah Abdullah Saleh Al-Reshoodi: received her B.Sc. in Information Technology from the College of Computer Science, Qassim University, Saudi Arabia in 2014. Currently, she is studying MSc in Computer Science, at Qassim University. In addition, from 2016 to this date, she is working as a Teaching Assistant in the Computer College, Qassim University, Saudi Arabia.


Authors’ Profiles

Mohammed Abdullah Al-Hagery: received his B.Sc in Computer Science from the University of Technology in Baghdad Iraq-1994. He got his MSc. in Computer Science from the University of Science and Technology Yemen-1998. Al-Hagery finished his Ph.D. in Computer Science and Information Technology, (Software Engineering) from the Faculty of Computer Science and IT, University of Putra Malaysia (UPM), November 2004. He was a head of the Computer Science Department at the College of Science and Engineering, USTY, Sana'a from 2004 to 2007. From 2007 to this date, he is a staff member at the College of Computer Science, from the College of Computer Science and Technology, Qassim University, KSA. He published more than 20 papers in various international journals. Dr. Al-Hagery was appointed a head of the Research Centre at the Computer College, and a council member of the Scientific Research Deanship Qassim University, KSA from September 2012 to October 2018. Currently, he is teaching the master degree students and a supervisor of four master thesis. He is a jury member of a number of PhD and master thesis, as an internal and external examiner in his field of his specialist.

Latifah Abdullah Saleh Al-Reshoodi: received her B.Sc. in Information Technology from the College of Computer Science, Qassim University, Saudi Arabia in 2014. Currently, she is studying MSc in Computer Science, at Qassim University. In addition, from 2016 to this date, she is working as a Teaching Assistant in the Computer College, Qassim University, Saudi Arabia.

Copyright © 2019 MECS

I.J. Information Technology and Computer Science, 2019, 7, 17-25