Пропонується гібридний метод виявлення неповних дублікатів в таблицях. Метод базується на моделі знаходження неповних дублікатів в текстових даних на основі локально-чутливого хешування та моделі найближчого сусіда для ідентифікації подібностей в числових даних. Цей метод може бути використаний для встановлення запозичень в наукових публікаціях та дисертаційних роботах

Ключові слова: неповний дублікат, подібність, локально-чутливе хешування, метод найближчого сусіда

Предлагается гибридный метод выявления неполных дубликатов в таблицах. Метод основан на модели нахождения неполных дубликатов в текстовых данных на основе локально-чувствительного хеширования и модели ближайшего соседа для идентификации подобий в числовых данных. Этот метод может быть использован для установления заимствований в научных публикациях и диссертационных работах

Ключевые слова: неполный дубликат, подобие, локально-чувствительное хеширования, метод ближайшего соседа

UDC 005:37
DOI: 10.15587/1729-4061.2016.86243

DETECTION OF NEAR DUPLICATES IN TABLES BASED ON THE LOCALITY-SENSITIVE HASHING METHOD AND THE NEAREST NEIGHBOR METHOD

P. Lizunov
Doctor of Technical Sciences, Professor
Department of Computer Science*
E-mail: lizunov@knuba.edu.ua
A. Biloshchytskyi
Doctor of Technical Sciences, Professor
Department of Technology Management
Taras Shevchenko National University of Kyiv
Volodymyrska str., 60, Kyiv, Ukraine, 01033
E-mail: bao1978@gmail.com
A. Kuchansky
PhD, Associate Professor
Department of Information Technologies*
E-mail: kuczanski@gmail.com
S. Biloshchytska
PhD, Associate Professor
Department of Information Technology Designing and Applied Mathematics*
E-mail: bsvetlana2007@ukr.net
L. Chala
PhD, Associate Professor
Department of Artificial Intelligence
Kharkiv National University of Radio Electronics
Nauky ave., 14, Kharkiv, Ukraine, 61166
E-mail: larysa.chala@nure.ua
*Kyiv National University of Construction and Architecture
Povitroflotskyi ave., 31, Kyiv, Ukraine, 03037

1. Introduction

A table is understood as an arrangement of various types of data in rows and columns, or in the form of a more complex structure. Tabular presentation of information is widely used in scientific research, when presenting results of data analysis of considerable volume, etc. The look of tables varies considerably by structure, flexibility and designation and is different from the feature of content that should be represented in them. A peculiarity of tables in technical and scientific publications is that they are usually separated from the main text in a separate block that has numbering and name [1]. In the classic sense, a table is considered to be the context for reading [2], tables are also an integral part of information systems for data processing [3]. Authors [4] regard a tabular representation of data to be one of the steps of the process of their mapping and visualization.

In general, a table consists of orderly arranged rows and columns. The term "row" can be defined also as vector, record or tuple. The usual term "column" is often interpreted as field, attribute or property that has a specified title or a name. This name may a priori consist of a word or a sequence of words, be presented by numeric values, a formula (formulas), or a date. The intersection of the particular row and column defines a cell, which is filled with content. In general, the presentation of tables is very diverse: elements of the tables can be grouped according to certain attributes, split into segments, nested with varying degrees of nesting, contain annotations, objects of the type of formula, date, etc.

In the case of hiding plagiarism, the original table can be relatively easily modified: lines and columns transposed,

headings and field names changed. Also, if we speak about abusive borrowing, then the table data can be essentially modified. For example, if the original table contains results of numerical experiment, then in the borrowing these numeric data may be deliberately changed. An example of certain original table is indicated in Table 1 versus its modification in Table 2. All this considerably complicates comparing tables for the identification of near-duplicates.

Table 1

Example of original table

Company name	Income	Number of employees
Corporation TechBud	225 750	560
New Company	785 960	1220
Alpha-Bud	895 975	1150

Table 2

Example of the modification of original table

Company	Number of employees	In.
Alpha-Bud	1150	895 975
Corp. TechBud	560	225 750
N.Company	1220	785 960

The research, which is considered in present paper, is relevant, since it may prove to be a valuable tool to prevent abuse and plagiarism in higher education, as well as for the creation of mechanisms to ensure combatting plagiarism at the state level. The method proposed in the article might be used for the development of a module of software package to detect near-duplicates in thesis and diploma papers, as well as in the scientific publications.

2. Literature review and problem statement

The task of finding near-duplicate detection (NDD) is a traditional task of intelligent text analysis. In particular, for detecting similarities and near-duplicates in text data, the algorithm of calculating the Levenshtein distance is used [5], which are string metrics that allow the computation of difference between two sequences of characters. Also for finding a substring, in the text similar to the assigned sample, the Bitap algorithm with the Wu-Manber modifications is used [6]. The distances between lines in this case are defined in terms of the Levenshtein distance. For the problem on finding near-duplicates, the BK-tree algorithm is also applied [7, 8], which consists in the construction of a metric tree for discrete metric spaces. This algorithm may be used to identify similar lines with regard to dictionary. Paper [9] proposes to detect material suspicious of plagiarism on the basis of similarities analysis. The search methods for near-duplicates, based on locally-sensitive hashing, are considered in article [10]. Paper [11] presented mathematical formalization for the problem on searching for near duplicates in text data. Article [12] demonstrated a comparison of different methods of finding near-duplicates by the average time of algorithms operation and by the quantity of found words. The problem of finding near duplicates in numerical data is related to the identification of similarities in the time series based on the comparison to the sample using the method of nearest neighbors with assigned metric [13, 14]. Paper [15] proposes a system based on finding the samples and their comparison. Article [16] considers textural features that correspond to the visual perception of humans for the digital analysis of textures, as well as the recognition of graphic information. Paper [17] examines comparative analysis of different methods of image recognition. Article [18] describes a system that searches for keywords and frames, taking into account the feedback when processing images. Since the table can present text, graphic, and numeric data, then the reviewed publications may come in handy when detecting near-duplicates in tables. At present, however, there is no a clearly substantiated method to compare tables with regard to different ways of their representation.

The task of finding near-duplicates in tables is a process of identification of such tables that are most similar to each other. The similarity in this case is expressed by certain functional F that assigns distance between the tables. If this distance does not exceed some numerical threshold value, then the tables are considered to be similar, hence, the data in these tables contains near-duplicates.

Present article addresses the problem of detecting near-duplicates in the tables that represent results of experimental scientific research and are presented in the texts of thesis and diploma papers, scientific publications, etc. Since the data in these tables can be represented by different types, then the traditional methods for the identification of similarities in such data should be replaced with comprehensive or hybrid methods.

3. The purpose and tasks of the study

The purpose of present study is to build a table analysis algorithm for the detection similarities in them and to construct a hybrid method for the detection of near-duplicates in tables based on the methods of locality-sensitive hashing and nearest neighbors.

To achieve the purpose, it was necessary to solve the following tasks:

– determining the types and the classification of data representation in a tabular form;

– performing the indexing of the table content;

– construction of a hybrid method for the detection of near-duplicates in tables based on finding near duplicates in text data and a model for the identification of similarities in numerical data based on the nearest neighbor method.

4. The types of data that are represented in table cells

We may distinguish six different types of data represented in table cells:

1. Numeric type. In this case, only those numeric sets from the hierarchy of numbers are taken into account, which do not contain letters in a number record, as well as other symbols (root square, slash, etc.). That is, in this type we shall include all real numbers $\mathbb{R}$, $\mathbb{N}\subset\mathbb{Z}\subset\mathbb{R}$. It should be noted that during preliminary examination of the table, data of the numeric type with decimal separator should be unified, that is, in all tables that are analyzed by the system for finding near-duplicates, a decimal separator must be in the form of a full stop «.», or a comma «,». According to the ISO 31–0 international standard, both a full stop and a comma may be used as the decimal separator.

2. Text or string type. This type is represented by a finite sequence of symbols from a particular alphabet.

3. Formula type. This type is a graphic object that was created by using a certain markup language or equation editor: MathType, KFormula, MathCastmula, TeX, MathML, etc.

4. Date and time type. This type represents a record that includes a day, a month and year, sometimes a week number. According to the ISO 8601 international standards, there are two types of recording a date: year-month-day and year. month.day (for example, 2016–05–30 and 2016.05.30). According to the GOST P 6.30–2003 standard, it is allowed using the record day.month.year (for example, 30.05.2016). In the USA they use the record month/day/year (for example, 5/30/2016). For labeling, the record year+week is used (for example, 1623). All forms of recording dates can be clearly defined, thus, in the case of considering a table with cells of the date type, formats should be brought to the same type.

5. Picture type that is represented in the form of a graphic object.

6. A combined type, which includes several of the listed types at a time.

Let us consider the known types of tabular representation of information. An ordinary table is the table with a finite number of lines and columns that contains no grouping and merging multiple cells into one cell and each cell of the table contains data of the specified type (numeric, string, date type, formula). A multidimensional table is the table in which data are normalized and arranged in a specified hierarchy. An example is the multiplication table.

Tables are created using spreadsheet editors that are special computer programs for the construction, analysis and storage of data in a tabular form. All tabular editors operate by a uniform principle: each cell may contain numbers, text data or results of formula work using the contents of other cells and standard mathematical, statistical and financial operations and functions. In addition to creating and editing tables, a spreadsheet editor helps create charts and graphs, as well as there are tools to use compiled tables as databases. The most popular tools for creating and editing tables, which are used when writing scientific papers, are: Microsoft Excel, Corel Quattro Pro, Lotus 1–2–3, Gnumeric, LibreOffice Calc (open source software), KSpread, etc. [19]. As practice testifies, the vast majority of tables in the scientific articles and dissertational papers in Ukraine are compiled by using the Microsoft Excel software.

5. Model of indexing the table data

Assume that B is a certain input table, and

$$\overline{B}=\left\{B_1,B_2,\ldots,B_p\right\}$$

are the table, selected from the text and stored in a general table database, p is the number of tables in the database. The task is to define such a set of tables

$$\tilde{B}=\left\{B_1^*,B_2^*,\ldots,B_l^*\right\},$$

that $\tilde{B}\subset\overline{B}$, l<p, for which a condition is satisfied:

$$F\left(B,B_i^*\right)<\lambda,\quad i=\overline{1,l}, \tag{1}$$

where λ is the threshold value, and F is the distance between the tables. The existence in the base of at least one such table from set $\overline{B}$, for which condition (1) is satisfied, indicates that the table B was borrowed.

Cell K_{ij} of certain table B designates such element of the table, which is formed by the intersection of the i-th line and j-th column of this table.

Content or cell data K_{ij} of certain table B designates such value (numeric, text, date type, etc.) that corresponds to the given cell in the table. Let us designate the content of cell K_{ij} through $\mathrm{Cont}\left(K_{ij}\right)$, $i=\overline{1,r_1}$, $j=\overline{1,r_2}$, where r_1 is the number of lines in table B, and r_2 is the number of columns in table B.

As defined in the previous paragraph, the cell content is limited by the following types: numeric, text, date type, picture, formula, combined type. If the table contains pictures and formulas, they are separated for different examination. The content of date type, upon bringing it to the unified format, may be regarded as a normal text string. Thus, by simplifying possible types of the content of cells, we can assume that the content of the table takes the form of deuce:

$$B=\langle N,S\rangle, \tag{2}$$

where N is the sequence of numeric data of cell of table B, and S is the sequence of text data from cells of table B.

Assume that $I=\left\{1,2,\ldots,r_1\right\}$ is the set of indexes or numbers of table rows, and $J=\left\{1,2,\ldots,r_2\right\}$ is the set of indexes or numbers of columns in table B. Let us check all cells of the table and define the type of their data. If data of a certain cell belongs to a numeric type, the relevant content is represented as an element of set $\overline{N}$, if – to the text type, then it is added to set $\overline{S}$ by rule:

$$\overline{N}=\left\{k\;\middle|\;k\in\mathrm{Cont}\left(K_{ij}\right),k\in\mathbb{R},i\in I,j\in J\right\}, \tag{3}$$

$$\overline{S}=\left\{k\;\middle|\;k\in\mathrm{Cont}\left(K_{ij}\right),k\in T,i\in I,j\in J\right\}, \tag{4}$$

where T is the set of symbols of power L,

$$\mathrm{card}(T)=L:\ T=\left\{t_1,t_2,\ldots,t_L\right\},$$

t_i is the separate symbol, $i=\overline{1,L}$, $t_i\in A$, A is the set of atomic symbols of formal language or an alphabet.

Let us represent sets $\overline{N}$ and $\overline{S}$ in the form of a numeric sequence and a sequence of lines in accordance with length v and w, that is $N=\left\{n_1,n_2,\ldots,n_v\right\}$ – a sequence of numeric values of the contents of cells, $v=\mathrm{card}(N)$, $n_i\in N$, $i=\overline{1,v}$, and $S=\left\{s_1,s_2,\ldots,s_w\right\}$ is the sequence of lines of the contents of cells $w=\mathrm{card}(S)$, $s_j\in S$, $j=\overline{1,w}$.

Let us consider a separate representation of these sequences in the form convenient for the application of models for the identification of similarities and search for near-duplicates.

Let us consider sequence $S=\left\{s_1,s_2,\ldots,s_w\right\}$. Each of its element contains a text that may consist of one or more words. Let us check successively all elements of the sequence from s_1 to s_w and select words from them. A word of arbitrary element of sequence S is assigned as sequence:

$$S_n^\beta=\left\{t_1,t_2,\ldots,t_\beta\right\},$$

where $n \in \mathbb{N}$ is the serial number of word, β is the length of word,

$$t_j \in A,\ \ t_j \notin C,\ \ j = \overline{1,\beta},\ \ C = \{"\_",",",".","\ ","-",":",";","\#"\}$$

are all symbols of the elements of sequence S, which are not letters $S_n^{\beta} \in s_j,\ \ j \in \{1,2,\ldots,w\}$.

Let us form a new sequence from all words of the elements of sequence S, excluding in advance from consideration the so-called stop words. A list of stop words will be assigned as set

$$M = \{"і","та","але","або","тощо","і т.п.","і тд."\}.$$

Then the new sequence of words takes the form:

$$W = \{S_1^{\beta_1}, S_2^{\beta_2}, \ldots, S_m^{\beta_m}\},$$

where β_j, $j = \overline{1,m}$ are the lengths of words, and m is their number. The elements of such sequences represent words in the canonized form.

Using the sliding window method, let us build a set of sequences:

$$E_1 = \{S_1^{\beta_1}, S_2^{\beta_2}, \ldots, S_h^{\beta_h}\},$$

$$E_2 = \{S_2^{\beta_2}, S_3^{\beta_3}, \ldots, S_{h+1}^{\beta_{h+1}}\},$$

$$E_{m-h+1} = \{S_{m-h+1}^{\beta_{m-h+1}}, S_{m-h+2}^{\beta_{m-h+2}}, \ldots, S_{m-1}^{\beta_{m-1}}, S_m^{\beta_m}\},$$

where h is the size of window or a number of elements of the constructed sequences $E_1, E_2, \ldots, E_{m-h+1}$.

Next, by the locality-sensitive hashing method, we shall represent a set of sequences

$$F(W) = (E_1, E_2, \ldots, E_{m-h+1})$$

in the form of bit strings, that is,

$$\Delta(W) = (I(E_1), I(E_2), \ldots, I(E_{m-h+1})), \qquad (5)$$

where $I(E_k)$ is the index element that assigns the bit string that unequivocally represents sequence E_k, $k = \overline{1,m-h+1}$. That is,

$$I(E_k) = \{\delta_{k1}, \delta_{k2}, \ldots, \delta_{kc}\}, \qquad (6)$$

where $\delta_{kx} \in \{0,1\}$, $k = \overline{1,m-h+1}$, $x = \overline{1,c}$, c is the number of bits that represents the bit sequence.

Let us consider numeric sequence $N = \{n_1, n_2, \ldots, n_v\}$ of input table B and construct for it a set of sub sequences by the sliding window method, that is,

$$K_1 = \{n_1, n_2, \ldots, n_g\},$$

$$K_2 = \{n_2, n_3, \ldots, n_{g+1}\},$$

...

$$K_{v-g+1} = \{n_{v-g+1}, n_{v-g+2}, \ldots, n_{v-1}, n_v\},$$

where v is the number of elements in sequence N, and g is the size of window or the number of elements of sub sequences $K_1, K_2, \ldots, K_{v-g+1}$. As the elements of the constructed sub sequences are valid numbers, $n_i \in \mathbb{R}$, $i = \overline{1,v}$, then these subsequences may appear to be g-dimensional vectors. That is, if one assumes that we assigned space $\mathbb{R}^g$ that has an Euclidean structure, then it is possible to determine in this space metric ρ, between any two vectors of space $a \in \mathbb{R}^g$ and $b \in \mathbb{R}^g$: $\rho(a,b)$. Moreover, this metric will satisfy the axiom of identity, that is,

$$\rho(a,b) = 0 \Leftrightarrow a = b,$$

the axiom of symmetry:

$$\rho(a,b) = \rho(b,a)$$

and triangle axiom and for certain vector $c \in \mathbb{R}^g$:

$$\rho(a,c) \le \rho(a,b) + \rho(b,c).$$

This metric or measure of proximity (similarity) between such vectors that represent the numeric values of the tables contents, is integral, which will determine the degree of similarity of these tables.

6. A hybrid method for detecting near-duplicates in tables

According to the formulation of the problem, assume that B is the input table, and $B_1, B_2, \ldots, B_p$ are the tables selected and stored in a general tables database, p is the number of tables in the base. The task is to define such tables in the base, for which condition (1) is satisfied.

Assume that we built a sequence of text data

$$S = \{s_1, s_2, \ldots, s_w\}$$

and a sequence of numeric values

$$N = \{n_1, n_2, \ldots, n_v\}$$

for table B. Since the base with tables is already known, then it is obvious that each such table is indexed. That is, for each one of the tables $B_1, B_2, \ldots, B_p$, the sequences of text data

$$S^y = \{s_1^y, s_2^y, \ldots, s_w^y\}$$

and sequences of numeric data are known

$$N^y = \{n_1^y, n_2^y, \ldots, n_v^y\},\ \ y = \overline{1,p}.$$

It is also obvious that for the sequences of words, the index elements are assigned

$$I(E_{k_y}^y) = \{\delta_{k_y 1}^y, \delta_{k_y 2}^y, \ldots, \delta_{k_y c}^y\},$$

$$\delta_{k_y x}^y \in \{0,1\},\ \ k_y = \overline{1, m_y - h + 1},\ \ x = \overline{1,c},$$

where c is the number of bits that represents the bit sequence, m_y is the number of words in sequences,

$$W^y = \{S_1^{y,\beta_1}, S_2^{y,\beta_2}, \ldots, S_{m_y}^{y,\beta_{m_y}}\},$$

that include words S_j^{y,β_j} in the canonized form, β_j, $j = \overline{1,m_y}$ are the length of the words.

Let us build for string sequence $S = \{s_1, s_2, \ldots, s_w\}$ of input table B a sequence of words in the canonized form

$$W=\left\{S_1^{\beta_1},S_2^{\beta_2},\ldots,S_m^{\beta_m}\right\},$$

where β_j, $j=\overline{1,m}$ are the lengths of words, m is their number.

Next, by the sliding window method, we define sequences $E_1,E_2,\ldots,E_{m-h+1}$ and, by the locality-sensitive hashing method, we shall build index entries

$$I\left(E_k\right)=\left\{\delta_{k1},\delta_{k2},\ldots,\delta_{kc}\right\},$$

where $\delta_{kx}\in\left\{0,1\right\}$, $k=\overline{1,m-h+1}$, $x=\overline{1,c}$, c is the number of bits that represents the sequence.

c is the number of bits that represents the sequence.

We shall calculate the Hamming distances from each index elements sequence of the input table to the index elements of sequences of those tables that are in the base according to the following formula:

$$H\left(I\left(E_k\right),I\left(E_{k_y}^y\right)\right)=\frac{1}{c}\sum_{j=1}^{c}\left|\delta_{kj}-\delta_{k_yj}\right|, \quad (7)$$

$$k=\overline{1,m-h+1},$$

$$k_y=\overline{1,m_y-h+1},$$

$$y=\overline{1,p}.$$

If the condition is satisfied

$$H\left(I\left(E_k\right),I\left(E_{k_y}^y\right)\right)<\lambda_H \quad (8)$$

for the assigned in advance value of parameter $\lambda_H\in\left[0,1\right]$, then, with probability 1, we can argue that index entry with number k is similar to the index entry with number k_y in the table with number y. This indicates that the table with number y may be similar to the input table with threshold λ_H, that is, it contains a near-duplicate.

Let us build for numeric finite sequence

$$N=\left\{n_1,n_2,\ldots,n_v\right\}$$

of input table B a set of sub sequences

$$K_1,K_2,\ldots,K_{v-g+1}.$$

We also assume that for each of the tables $B_1,B_2,\ldots,B_p$ based on their sequences of numeric data

$$N^y=\left\{n_1^y,n_2^y,\ldots,n_v^y\right\},\ y=\overline{1,p},$$

there were built subsequences $K_1^y,K_2^y,\ldots,K_{v-g+1}^y$ by the sliding window method:

$$K_1^y=\left\{n_1^y,n_2^y,\ldots,n_g^y\right\},$$

$$K_2^y=\left\{n_2^y,n_3^y,\ldots,n_{g+1}^y\right\},$$

$$K_{v-g+1}^y=\left\{n_{v-g+1}^y,n_{v-g+2}^y,\ldots,n_{v-1}^y,n_v^y\right\}.$$

If we represent the built subsequences $K_1,K_2,\ldots,K_{v-g+1}$ and $K_1^y,K_2^y,\ldots,K_{v-g+1}^y$ in the form of vectors, then similarity measures between them are determined on the basis of Euclidean distance, city metrics or Minkowski distance. We shall receive the following y of matrices of distances:

$$\rho_1\left(K_u,K_r^y\right)=\sqrt{\sum_{j=r}^{g+r-1}\left(n_{j+u-r}-n_j^y\right)^2}, \quad (9)$$

$$\rho_2\left(K_u,K_r^y\right)=\sum_{j=r}^{g+r-1}\left|n_{j+u-r}-n_j^y\right|, \quad (10)$$

$$\rho_3\left(K_u,K_r^y\right)=\left(\sum_{j=r}^{g+r-1}\left|n_{j+u-r}-n_j^y\right|^t\right)^{\frac{1}{t}}, \quad (11)$$

$$y=\overline{1,p},\ u=\overline{1,v-g+1},\ r=\overline{1,v-g+1},$$

where t is the Minkowski distance parameter.

Next, we shall find minimum values in each row of matrices $\rho_\tau\left(K_u,K_r^y\right)$, that is, by $r=\overline{1,v-g+1}$ and obtain for each $y=\overline{1,p}$ of the distance:

$$\zeta_\tau\left(K_u,K_{min}^y\right)=\min_{r=\overline{1,v-g+1}}\left\{\rho_\tau\left(K_u,K_r^y\right)\right\}, \quad (12)$$

at $u=\overline{1,v-g+1}$, for fixed $\tau=\overline{1,3}$.

Let us normalize values of the obtained distances by formula:

$$\zeta_\tau^N\left(K_u,K_{min}^y\right)=$$

$$=\frac{\zeta_\tau\left(K_u,K_{min}^y\right)-\min\limits_{u=\overline{1,v-g+1}}\left\{\zeta_\tau\left(K_u,K_{min}^y\right)\right\}}{\max\limits_{u=\overline{1,v-g+1}}\left\{\zeta_\tau\left(K_u,K_{min}^y\right)\right\}-\min\limits_{u=\overline{1,v-g+1}}\left\{\zeta_\tau\left(K_u,K_{min}^y\right)\right\}}, \quad (13)$$

$$y=\overline{1,p},\ u=\overline{1,v-g+1},\ \tau=\overline{1,3}.$$

If the condition is satisfied

$$\zeta_\tau^N\left(K_u,K_{min}^y\right)<\lambda_\rho, \quad (14)$$

for the assigned in advance value of parameter $\lambda_\rho\in\left[0,1\right]$, then, with probability 1, we can argue that vector u is similar to the vector of table with number y, that is, the table contains a near-duplicate. The higher the value λ_ρ, the more stringent requirements to the search for near-duplicates are.

An algorithm for the detection of near-duplicates in tables $B_1,B_2,\ldots,B_p$ relative to table B, which is assigned by deuce (2), consists of the following steps:

1. Separate from the input table B graphic objects: images and formulas. These objects are examined separately. For comparing formulas, a method may be applied, which is based on the comparison between samples or templates.

A method of finding near duplicates in mathematical formulas includes the following stages:

1) identifying formulas in the analyzed texts;

2) creating samples based on the identified formulas;

3) comparing the samples of the found formulas between them;

4) checking the context of formulas that have identical sample, taking into account the presence of commonly used symbols in these formulas;

5) formulas with identical sample and content are considered to be near duplicates.

2. In the case when table B contains cells with data of the date type, it is proposed to bring all the dates in accordance with a unified format, for example, "day.month.year", and to explore the cell data as the cell with the content of the text type.

3. The entire content from cells of numeric type should be brought to a uniform type: decimal separator is to be reprsented in the form of a comma «,».

4. Delete column «No. of entry» from the table, which represents the numbering of lines if the latter exists.

5. Build a sequence of text data S and a sequence of numeric values N. When dividing, pay attention to the cells with combined content: in the case when a particular cell contains numeric and text data, the text of the given cell and numeric data are split into individual elements of the sequences.

6. Build for sequence S of the input table B a sequence of words in the canonized form W. Next, define sequences $E_1,E_2,\ldots,E_{m-h+1}$ and, by the method of locality-sensitive hashing, construct elements of index $I(E_k)$.

7. Next, calculate the Hamming distances from each index elements sequence of the input table to the index elements of sequences of those tables that are in the base by formula (7) and check condition (8) for the assigned threshold value $\lambda_H \in [0,1]$. If the condition is satisfied, it means that a near duplicate is identified.

8. Build for numeric sequence N of input table B a set of sub sequences $K_1,K_2,\ldots,K_{v-g+1}$.

9. Represent subsequences

$$K_1,K_2,\ldots,K_{v-g+1} \text{ and } K_1^y,K_2^y,\ldots,K_{v-g+1}^y$$

in the form of vectors and calculate the distances according to formulas (9)–(11). As a basis, one may choose one of these formulas.

11. Apply formulas (12), (13) and check condition (14). If, for the assigned threshold value $\lambda_\rho \in [0,1]$, this condition is satisfied, it can be argued that a near duplicate is identified. Value λ_ρ and λ_H are determined in advance by way of experiment.

12. When determining near-duplicates by this method, special attention should be paid to those tables, for which conditions (8) and (14) are satisfied separately, especially if the calculated distance is significantly less than the threshold. If these conditions are satisfied simultaneously for the specified values of λ_ρ and λ_H, then such a table unequivocally contains a near duplicate and should be separately examined by the expert if it was borrowed.

7. Discussion of results of research into detection of near-duplicates in tables

As a result of the research, we described and formalized a hybrid method for the detection of near-duplicates based on the method of locality-sensitive hashing and the nearest neighbor method. This method may be used in antiplagiarism-systems, and other systems that are designed to run intelligent analysis to identify the similarities in information presented in a tabular form. An application of the method is possible due to its special features: a comparison of tables that contain different types of data at a time (text, numeric), as well as the fact that the hybrid method for the detection of near-duplicates is relatively easy to implement because it is based on the known methods of nearest neighbor and locality-sensitive hashing.

An advantage of the described method is in the fact that, when comparing scientific papers for the detection of near duplicates, the tables that are identified in the texts can be verified on the basis of the developed hybrid method that takes into account the text and numerical data, presented in the tables, at the same time. A shortcoming of the method is that in the case of detecting in the tables objects of formulas or other graphic data, they are treated separately and are ignored in present method.

The research, which is the topic of this paper, is the continuation of research into the detection of near duplicates in text and graphic data. Some of the results of these studies were examined in articles [10–12]. The hybrid method constructed for the detection of near duplicates in tables is a part of a comprehensive research into the construction of methods for the analysis of research papers on plagiarism.

8. Conclusions

1. When determining the types of content in cells and performing the indexation of the table, it is possible to assume that a table is given in the form of a set of text and numeric data (2). In this case, data of the "date" type are represented in one of the known formats, while graphic data and objects of the "formula" type are examined separately from the tables.

2. The first stage in analyzing a table is the indexation of data. For text data, the finite sequences are formed from the words in the canonized form, from which, based on the method of locality-sensitive hashing, bit sequences are constructed. For numeric data, finite numeric sequences are formed.

3. For the detection of near duplicates in tables, a hybrid method can be applied that finds similarities between the text and numeric data separately and gneralizes the results. A similarity between data in the case of input text data is determined by the Hamming distance at the assigned threshold value. In the case of numeric data, similarity is determined based on the method of the nearest neighbours with specified metric distances, while the vectors, between which the distances are calculated, represent numeric content of the table with a decimal separator brought to a uniform format.

References

1. Fink, A. How to Conduct Surveys [Text] / A. Fink. – Thousand Oaks: Sage Publications, 2005. – 224 p.
2. Ehrenberg, A. S. C. A Primer in Data Reduction [Text] / A. S. C. Ehrenberg. – Wiley, Chrichester, UK, 1982. – 324 p.
3. Bertin, J. Graphics and Graphic Information Processing [Text] / J. Bertin. – Walter de Gruyter Berlin, New York, 1981. – 279 p. doi: 10.1515/9783110854688
4. Reading in Information Visualization: Using Vision to Think [Text] / S. K. Card, J. D. MacKinlay, B. Shneiderman (Eds.). – Morgan Kaufmann, San Francisco, 1999. – 712 p.

5. Su, Z. Plagiarism detection using the Levenshtein distance and Smith-Waterman algorithm [Text] / Z. Su, B.-R. Ahn, K.-Y. Eom, M.-K. Kang, J.-P. Kim, M.-K. Kim // 2008 3rd International Conference on Innovative Computing Information and Control. – 2008. doi: 10.1109/icicic.2008.422
6. Wu, S. A fast algorithm for multi-pattern searching [Text] / S. Wu, U. Manber // Technical Report TR-94-17. – Department of Computer Science, University of Arizona, 1994. – 11 p.
7. Burkhard, W. A. Some approaches to best-match file searching [Text] / W. A. Burkhard, R. M. Keller // Communications of the ACM. – 1973. – Vol. 16, Issue 4. – P. 230–236. doi: 10.1145/362003.362025
8. Baeza-Yates, R. Proximity matching using fixed-queries trees [Text] / R. Baeza-Yates, W. Cunto, U. Manber, S. Wu // Lecture Notes in Computer Science. – 1994. – P. 198–212. doi: 10.1007/3-540-58094-8_18
9. Shenoy, M. Automatic Plagiarism Detection Using Similarity Analysis [Text] / M. Shenoy // Advanced Computing: An International Journal. – 2012. – Vol. 3, Issue 3. – P. 59–62. doi: 10.5121/acij.2012.3306
10. Biloshchytskyi, A. Optimization of Matching algorithms by using local-sensitive hash sets of text data [Text] / A. Biloshchytskyi, O. Dikhtiarenko // Management of complex systems. – 2014. – Issue 19. – P. 113–117.
11. Biloshchytskyi, A. The method of elimination of erroneous coincidences text in electronic documents [Text] / A. Biloshchytskyi, S. Kristof, S. Biloshchytska, O. Dikhtiarenko // Management of Development of Complex Systems. – 2015. – Issue 22 (1). – P. 144–150.
12. Biloshchytskyi, A. The effectiveness of methods for finding matches in texts [Text] / A. Biloshchytskyi, O. Dikhtiarenko // Management of complex systems. – 2013. – Issue 14. – P. 144–147.
13. Kuchansky, A. Pattern matching method for time-series forecasting [Text] / A. Kuchansky, V. Nikolenko // Management of Development of Complex Systems. – 2015. – Issue 22. – P. 101–106.
14. Kuchansky, A. Selective pattern matching method for time-series forecasting [Text] / A. Kuchansky, A. Biloshchytskyi // Eastern-European Journal of Enterprise Technologies. – 2015. – Vol. 6, Issue 4 (78). – P. 13–18. doi: 10.15587/1729-4061.2015.54812
15. Mojsilovic, R. Matching and retrieval based on the vocabulary and grammar of color patterns [Text] / R. Mojsilovic, J. Kovacevic, J. Hu, R. J. Safranek, S. K. Ganapathy // IEEE Transactions on Image Processing. – 2000. – Vol. 9, Issue 1. – P. 38–54. doi: 10.1109/83.817597
16. Tamura, H. Texture features corresponding to visual perception [Text] / H. Tamura, S. Mori, T. Yamawaki // IEEE Transactions on Systems, Man, and Cybernetics. – 1978. – Vol. 8, Issue 6. – P. 460–473. doi: 10.1109/tsmc.1978.4309999
17. Zhang, D. Content-Based Shape Retrieval Using Different Shape Descriptors: A Comparative Study [Text] / D. Zhang, G. Lu // IEEE International Conference on Multimedia and Expo, 2001. ICME 2001. – 2001. doi: 10.1109/icme.2001.1237928
18. Quack, T. A System for Largescale, Content based Web Image Retrieval [Text] / T. Quack, U. Monich, L. Thiele, B. Manjunath // MM'04. – 2004. – P. 120–123.
19. Liebowitz, S. Network Effects and the Microsoft Case [Text] / S. Liebowitz, S. E. Margolis // Dynamic Competition and Public Policy. – 2001. – P. 160–192. doi: 10.1017/cbo9781139164610.007