Some approaches for improving quality of tabular data

V.V. Paramonov^{1[0000-0002-4662-3612]} and E.A. Lomaeva^{2[0000-0002-2373-0108]}

¹ Matrosov Institute for System Dynamics and Control Theory, SB RAS, Irkutsk,

Russia

slv@icc.ru ² Institute of Mathematics and Information Technologies of Irkutsk State University, Irkutsk, Russia ekaterina7lomaeva@gmail.com

Abstract. A spreadsheet is one of popular forms for presentation and transferring data of the same types. The area of using this kind of documents is very widespread. Extraction tables from spreadsheets and their understanding are significant tasks that allow getting useful information for further use, for example in processes of integration data that obtained from various sources. As rule tables in spreadsheets create by humans and for humans use. This feature could be the reason that tables may contain messy data such as misprints, errors of calculation, incorrect structure etc. It leads to the complication of automated table processing and understanding. This paper has discussed some approaches to data cleanse that improve the quality of tabular data. The approaches consist of checking and correction of cells calculation and spelling errors. We use phonetic words similarity to correct spelling mistakes in words and heuristic algorithms to detect calculated values in cells.

Keywords: spreadsheet \cdot data quality \cdot data cleanse \cdot data correction \cdot phonetic algorithms.

1 Introduction

Spreadsheet is a convenient format of documents to store and present sets of semistructured data. Since their inception in the 1980s, they have revolutionised the organisation and storage of data [4]. This kind of documents are used everywhere, where required to present information of the same type, from everyday tasks, such as planning a family budget, to professional ones (storing records about the company's clients and various kinds of statistical reports). Thus, spreadsheets are a valuable source of information. Extraction and subsequent integration of data from spreadsheet within the subject area to do multicriteria analysis, to identify dependencies between various factors, and to find patterns.

Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Typically, a person is the author of spreadsheets. Most documents of this type are created by people rather than automatically by machines. Consequently, data and their organisation may have errors. Some studies report that about more 90% of documents contain some errors [6,1] of various types. It may be different typos, calculation errors, the inconsistency of measurements, etc. The presence of errors leads to a decline in data quality. Low data quality complicates automated table processing such as data extraction and understanding. This is especially important in data integration processes. In this paper, we propose two approaches that allow to improve the quality of textual (on Russian language) and numerical values in spreadsheets.

2 Background

To integration, data must be carefully assembled from a variety of sources around the domain, cleansed and quality-assured [3]. Messy data complicate the processing process. For example it is impossible to correctly compare data from different sources with classifiers values or among themselves. Thus one of the important issues before data integration is a data cleanse. Data cleanse is the process of identifying and correcting errors and inconsistencies in order to improve the quality of the initial (raw) data [13]. This process includes many aspects, such as identifying spelling errors, missing data, the presence of fictitious, encoded, composite values, logical inconsistencies. Different methods are used for detecting and cleansing such errors. These are fuzzy string comparison, phonetic comparison, neural networks usage etc. This paper is reviewed the errors related to spelling and calculation.

Estimation of strings similarity is possible with fuzzy string comparison methods. The algorithms are used to implement fuzzy search that based on the system of associative access to words that are contained in the text index of the full-text document storage. The letters that form word(s) act as search units. The search process is significantly accelerated with the help of a specially created index containing fragments with links to words in which these fragments were encountered. The neighbourhoods of permissible distortions are set, with the help of which the accuracy and completeness of the search are easily regulated [10]. The size of the distortion is the percentage of different fragments and the allowed displacements of their positions. Then the words are selected, the fragments of which coincided with the fragments of the query words. Fuzzy string comparison methods are independent from the language alphabet. For raising the quality of misspelling detection the features of the language should be used.

In Russian language, following errors are possible [11]:

- morphological i.e. when a person tries to write all audible sounds by letters [12];
- phonologic, i.e. preservation of a writing of phonemes spelling regardless the word of change;
- phonetic, i.e. words are written as they heard;

 traditional due to historic or traditional style of old times or as in a language this word was borrowed from.

A lot of spelling errors in Russian language are associated with language phonetic norms. The type of mistakes generally depends on persons' education level [9]. Some of mistakes may be detected and fixed using phonetic algorithms. Algorithms of phonetic coding are based on the sequence of letters in a word and their pronunciation rules, converted into an encoding text (code, index, key). Thereby phonetic algorithms index words by their sound. The conclusion about the similarity of two different words in sound is based on the coincidence or similarity of their coding text. Phonetic coding algorithms are effective when comparing acoustic data with text samples. They have not lost their relevance even despite the emergence and development of new models and methods of speech recognition, as they are the basis for their applying in practice. Rules of words phonetic coding depend on language phonetic norms [2] and useful for estimation similarity of words by their pronunciation.

Data cleansing processes concern not only with strings. Numerical values may require a clean also. Some of them may have different formats of presentation, may require to undergo cleansing or may be as a result of numerical dependency.

Checking and verification of numerical data in tables that are result of calculations, i.e. "sum", "average", "minimum", "maximum", etc. is also reviewed in the paper. Results of calculation are redundant information in data integration processes. These values also may contain errors, because different techniques could me used for computation (not necessarily of using specialised software tools). Thus, identification of calculated values is one of the ways to improve data quality and, accordingly, to simplify data analysis procedures, including during integration. Detection of calculated values is one of the components of data cleanse. Consider some approaches to detection and correction types of messy data listed above.

3 Typos and misspelling correction

3.1 Technique of correction

Often different spelling errors could arise in tables that are formed by persons without using special dictionaries, classifiers etc. Types of spelling errors in this case depend on particularities of the language [11].

Textual information, presented in tabular view has some specificity. As a rule, these words which are names of something and written in the singular and nominative case. In this instance, possible not to use rules of words formation for estimation of their similarity. It is possible to apply phonetic algorithms in this case (evaluate not only strings similarity).

Well-known phonetic algorithms are based on words coding using Latin alphabet characters and the pronunciation rules of English language. Thereby we suggest to use Polyphon [8] algorithm for coding words of Russian language. The common stages of the Polyphon are as follows:

- substitution of Latin letters which are similar to Russian with Russian ones;
- removal of all non-Russian alphabet characters from the input string;
- modification of letters before dividers, i.e., special letters as "ъ" and "ь";
- transformation of doubled characters into one (e.g. "oo" to "o");
- reduction of a row of the same letters into one letter [8];
- transformation of specially defined character sequences (e.g. "cq" to "m").

In details, stages of Polyphon are describe in [7].

Use of phonetic algorithms in combination with fuzzy string comparison algorithms increases quality of identification and automated elimination of spelling errors in the initial (raw) data. In this research, we use the following scheme of algorithms combination:

- for each word in the dictionary, its Polyphon code is calculated and divided into trigrams;
- for each value row from user table, the Polyphon code is also calculated, and divided into trigrams;
- trigrams of Polyphon codes are compared;
- the originals of phonetically similar reference words are compared character by character with the original words from the tables using "similarity points" (in percentage).

If some words have the same level of "similarity points", it indicates phonetic similarity. In this case, fuzzy string comparison is used.

3.2 Experimental results

The accuracy of the words matching was made. We matched in pairs each misspelled word with each reference word from Ozhegov' [5] dictionary. The identical (have more than one meaning) words were reviewed as one word. Thus, the initial amount of words for error introduction is 11601. In these words some spelling errors were made [8]. The approach allowed to correctly identify 93.3% of unique words, 6% of words were incorrectly identified or had identical codes and points of similarity for different words, 0.7% of words are not identified at all.

The results of the experiment demonstrate that fuzzy phonetic comparison with trigrams has results worse results than by using prime coding [7,8]. However we were able to raise uniqueness of words identification. Accordingly its allow to improve the quality of textual data cleanse.

4 Numerical dependencies detection

Numerical values are the basis for data analysis. Some of values are the results of calculation ("sum", 'average", "minimum", "maximum", etc.) of another values. This values could be as a result of hidden numerical dependencies. For example when the decision about dependency acceptable after data analysis only but not from the columns/rows labels. Detection of such values significant for data processing. Firstly, these values are redundant in the consolidated database, due to they may be calculated with insignificant computational costs. Some of the numerical dependencies may contain errors. Authors of tables may use different ways for getting these values, both using special spreadsheets tools like "formulas" and calculate with a calculator. The variety of possibilities increases the likelihood of errors. We suggest a set of heuristics for identification of numerical dependencies, including hidden dependencies.

The calculated value can be indicated by a label in the header, set by formula, and to be a result of numerical dependencies. The first two of them require verification. Thus, all of the features are used for detection and verification of these values. The complex approach allows for increasing the quality of data deletion.

4.1 Table labels analysis

One way to identify calculated values (numerical dependencies) is to search for keyword values in table labels (headers). It is supposed that a table has hierarchical header or table itself presented in canonical form. The variety of different names (table 1) of keywords for denotation of numerical dependencies on Russian and English are supported.

Table 1. Groups of keywords

	#	Group	Keywords						
	1	sum	сумма, сум, итог, итого, всего, sum, summarising						
2		average	среднее, средн, average, avg						
3 m		minimum	минимум, мин, наименьшее, minimum, min						
	4	maximum	максимум, макс, наибольшее, maximum, max						

If a keyword found in the cell, a hypothesis is put forward about the location of the calculated values and their type. If a keyword is found in the header row, the calculated column is considered, and the dependency search is performed to the left and right of it. If the keyword is located in the column header, an assumption is made about the calculated row, and then the analysis is performed at the top and bottom of it.

4.2 Analysis of formulas

When parsing a column, for each row, it searches for the range that contains the initial position of the calculation. To do this, the cells from the calculated column to the beginning of the table are searched. For each cell, conditions which depend on the type of calculations are checked, i.e. analyses all cells from the calculated column to the beginning (ending) of the table. If a formula is written in a cell from a supposedly calculated column, it is analysed. Thus this algorithm is analysed all the cells to find the beginning of calculations, to extract this data from the formula, thereby reducing the amount of processed data.

The formula in the cell is retrieved as a string and processed using regular expressions. Such formulas as SUM, MIN, MAX, AVG are analysed. For the formula processing, its arguments must be inside the table and correspond to the search direction. If one of these conditions is absent, or the formula is written in an inappropriate form, a numeric value is extracted from the formula and an iteration over the cells is performed to find dependencies using the technique of cells data analyses.

4.3 Cells data analysis

Process of numerical dependencies in cells detection assumes sequential processing of all cells in the table in the case when borders of area were not established on the base of the keywords in labels or values of formulas. Consider reviewing the technique of cells data analyses for dependencies, presented in table 1 on the sample of "sum" dependency. All other dependencies (given in 4.2) are made analogously in regard to formulas. For each type of dependency, a search is carried out in columns from right to left, left to right, top-down and bottom-up. The sample of technique is presented in table 2.

Table 2. Example of a table for "sum" type dependencies

	А	В	C	D	Е
1	1	5	10	15	30
2	0	2	4	6	12
3	2	3	data	9	18

Let columns A-D in the table 2 were already processed and no calculated values were found in them. Consider the search of a dependency of "sum" type for column E. For each row to the left of the requested column in question, a search is performed for the cell that contains the first summand for the sum from the cell in the calculated column. For example, in row 1, the columns from D to A are explored, at each step the contents of the cells are summed up and compared with the number in cell E1. The number of terms must be greater than one since otherwise, the case is possible when the column contains duplicate values and not the sum.

The situation is possible when the term is suitable for several columns, not the one. It is demonstrated in row 2 of table 2. Therefore, a range of column indexes is written for each row, which can be the first summand for the assumed sum. The acceptable border range check for other rows. If the number of rows with the first term in one column is 80% or more of the total number of rows in the table, it is concluded that the column for which the dependency search was performed, is the sum - i.e., the calculated value.

5 Conclusion

Tabular data, such as reports, statistical reviews are a valuable source of information. However, some of the tables might have errors and negatively affect to automated table processing and understanding. In this regard, data cleansing is needed as one of the first steps of data integration. In this paper we suggest two approaches that allow to improve quality of tabular data. The suggested in the paper technique is based on phonetic coding and its codes fuzzy comparison for textual values and calculated values detection for numerical data are presented. It allows to clean tabular data from messy values, identify some of redundant data, and check the correctness of calculated values.

Acknowledgements

The research was supported by the Program of the Fundamental Research of the Siberian Branch of the Russian Academy of Sciences, project num. IV.38.1.2 (reg. num. AAAA-A17-117032210079-1). Results are achieved using the Centre of collective usage «Integrated information network of Irkutsk scientific educational complex.

References

- Abraham, R., Erwig, M.: Header and unit inference for spreadsheets through spatial analyses. In: Proceedings. 2004 IEEE Symposium on Visual Languages and Human Centric Computing(VLHCC). vol. 00, pp. 165–172 (09 2004). https://doi.org/10.1109/VLHCC.2004.29
- Alotaibi, Y.A., Meftah, A.H.: Review of distinctive phonetic features and the arabic share in related modern research. Turkish Journal of Electrical Engineering and Computer Sciences 21, 1426–1439 (2013)
- Ciganek, J.: Design and implementation of open-data data warehouse. In: 2019 6th International Conference on Advanced Control Circuits and Systems (ACCS) & 2019 5th International Conference on New Paradigms in Electronics & information Technology (PEIT). IEEE (Nov 2019). https://doi.org/10.1109/accspeit48329.2019.9062879, https://doi.org/10.1109/accs-peit48329.2019.9062879
- Guerrero, H.: Excel Data Analysis. Springer International Publishing (2019). https://doi.org/10.1007/978-3-030-01279-3
- 5. Ozhegov, S.: Dictionary of Russian language. About 53000 words. (Словарь русского языка: Ок. 53 000 слов) (in Rusian). Oniks, World and education, Moscow, 24 edn. (2007)
- 6. Panko, R.R.: Spreadsheet errors: What we know. what we think we can do. CoRR **abs/0802.3457** (2008)

- Paramonov, V., Shigarov, A., Ruzhnikov, G., Cherkashin, E.: Phonetic string matching for languages with cyrillic alphabet. In: Advances in Intelligent Systems and Computing, pp. 301–311. Springer International Publishing (Aug 2018). https://doi.org/10.1007/978-3-319-99981-4 28
- Paramonov, V.V., Shigarov, A.O., Ruzhnikov, G.M., Belykh, P.V.: Polyphon: An algorithm for phonetic string matching in russian language. In: Communications in Computer and Information Science, pp. 568–579. Springer International Publishing (2016). https://doi.org/10.1007/978-3-319-46254-7_46
- 9. Parubchenko, L.: Hypercorrection errors (in russian). Russian Literature pp. 23–27 (04 2005)
- Sharapov, R., Sharapova, E.: The problem of fuzzy duplicate detection of large texts. In: CEUR Workshop Proceedings. vol. 2212, pp. 270–277 (2018)
- Skripnik, Y., Smolenskaya, T.: Phonetics of modern Russian Language (in Russian), p. 152. VoSIGI, Stavropol (2010)
- Valgina, N., Rozental, D., Fomina, M.: Modern Russian Language: Textbook (in Russian). Logos, Moscow, 6-th edn. (2002)
- Zhao, L., Yuan, S.S., Peng, S., Wang, L.T.: A new efficient data cleansing method. In: Lecture Notes in Computer Science, pp. 484–493. Springer Berlin Heidelberg (2002). https://doi.org/10.1007/3-540-46146-9 48