Intelligent Control and Automation, **2013, 4, 227-234**
http://dx.doi.org/10.4236/ica.2013.42027 Published Online May 2013 (http://www.scirp.org/journal/ica)

Identification of Categorical Registration Data of Domain Names in Data Warehouse Construction Task

Rasim Alguliev, Rena Gasimova
Institute of Information Technology of ANAS, Baku, Azerbaijan
Email: director@iit.ab.az, renakasumova@gmail.com

Received January 31, 2013; revised March 1, 2013; accepted March 8, 2013

Copyright © 2013 Rasim Alguliev, Rena Gasimova. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

ABSTRACT

This work is dedicated to formation of data warehouse for processing of a large volume of registration data of domain names. Data cleaning is applied in order to increase the effectiveness of decision making support. Data cleaning is applied in warehouses for detection and deletion of errors, discrepancy in data in order to improve their quality. For this purpose, fuzzy record comparison algorithms are for clearing of registration data of domain names reviewed in this work. Also, identification method of domain names registration data for data warehouse formation is proposed. Decision making algorithms for identification of registration data are implemented in DRRacket and Python.

Keywords: Domain; Domain Name System; Registrar; Registrant; Category Data; Data Warehouse; Data Clearing; Fuzzy Search Algorithms; Damerau-Levenstein Distance; Decision Tree

1. Introduction

Decision support systems (DSS) have been used for formation of administration decisions, *i.e*. discovery of the best alternative for a lengthy period of time. Majority of commercial, social and governmental organizations no longer make serious decisions without using elements of computer analysis [1]. At all stages of complex administrative decision making—from identification of the problem to control of realization of made decision—the more accurate, timely and complete is the appropriate information, the more effective will be the decision. Information support of the decision making process determines the effectiveness of decision at all stages [2].

Modern approach to automation of decision making support is based on use of data warehouse (DW) concept. Rampant development of information technologies, and data collection, storage and procession means in particular, allows to collect vast volumes of data which require analyzing. DW provides analysts, executives and top managers with capability to study large volumes of interdependent data using fast interactive information reflection on different levels of detailing from different points of view in accordance with notion of the user on subject field [3,4].

Bill Inmon, author of DW concept, defined them as "subject oriented, integrated, unchanged, supporting data collection, organized to support management", designed to act as "unique and only source of truth", providing managers and analysts with trustworthy information necessary for operative analysis and decision making [5,6]. Richard Hackathorn, another founder of this concept, wrote that the objective of DW—is to provide "unique image of existing reality" for organizations [7]. Ralph Kimball, one of DW authors, described the warehouse as "a place, where people can access their data". He also formulated in [8,9] main requirements to data warehouses.

For correct comprehension of data concept, it is necessary to understand following principal moments: DW concept—is not a data analyzing concept; rather it is a concept of preparation for data analyzing; DW concept does not predetermine architecture of purposive analytical system. It describes the processes to be implemented in the system, but not the where precisely and how these processes must be implemented. The main objective of warehouses—is creation of a single logical presentation of data, contained in multitype data bases (DB) or in other words, a single corporate data model.

2. Relevance

Apparently, at modern development level of information technologies, simple relocation of data from one DW to

another does not cause difficulties. However if we simply relocate the data from all sources to a single DB, then we will obtain a "dump", an uncoordinated set of data. In order to create something more accessible for analysis by the final user, it is necessary to coordinate the data entering the DB from sources of data warehouse. In other words, to solve the main task of DW construction: to create the most coordinated, subject-oriented, integrated, time dependant data set.

Filling of warehouses, as a rule, is carried out by information from several data sources. Development or consequential support of qualitative information storage and processing systems is a complex problem. Human factor and partial absence of control at submittal lead to occurrence of distortions in data. Misprints and omissions are present almost in all details of saved objects, as well as in identification sets. During entrance stage of information into the data base, human factor is the main reason for occurrence of distortions. Damerau demonstrated that 95% of errors during typing by a person are transpositions [10].

Upon development of DW, very little attention is paid to cleaning of incoming information. Apparently, larger the volume of warehouse, the better it is considered. This is an erroneous practice and the best way to turn the data warehouse into a disposal dump. It is necessary to clean data. Information is heterogeneous and is collected from different sources.

Exactly presence of point sets of information collection makes the clearing process especially relevant. Generally speaking, errors are always committed, and it is impossible to completely dispose of them. Possibly, sometimes it is more reasonable to accept them, rather than spend money and time to get rid of them. But, in general cases, it is necessary to aim to reduce the amount of errors to an acceptable level so ever.

There are different kinds of errors. There are also errors characteristics to a certain subject field or task. Errors, which do not depend of task: contrariety of information; data omissions, anomalous values; noise; data entry errors etc. There are different kinds of solutions for each of these problems. Data omissions are a very serious problem for majority of DW.

Due to lack of information, as a matter of actual practice, application of DW is realized poorly or with significant limitations. Quantity of errors during data input is excessive, for example misprints, deliberate data distortions, inconsistence of formats, excluding typical errors related to specifics of data input application operation.

3. Objective

Domain is a domain namespace field and is characterized with independence of data allocation, inclusion of information system in domain contents, presence of special information systems (DNS servers) containing data on domain names allocated in domain and carries out the function of domain name space organization [11].

Registration data of domain include: domain name (domain), registry identifier (registrar), full name of the physical person (person), contact address of the physical person (address), domain administrator identifier (admin-o), organization identifier for administrative communication (admin-c), title of organization (organization), domain registration time (created), domain free date (free-date), telephone numbers with international codes (phone), e-mail address (e-mail), list of DNS servers supporting domain (nserver), domain type (type), information source (source), domain registration payment time (paid-till) [12].

Clearing of domain name registration data is carried out in the works and domain names registration data identification method is developed based on decision tree apparatus application. Decision trees are selected as the main algorithmic approach for construction of effective data integration system.

During construction of data warehouse, problems occur related to misprints and omission of data. As registration data of domain mainly consists of categorical data, it was decided to apply fuzzy search algorithms for cleaning of these data.

Following tasks are formulated for research purposes:

1) Processing of domain name registration data, data clearing using Damerau-Levenstain algorithm;

2) Identification of domain name registration data using decision tree construction.

4. Solution

4.1. Algorithm Choice

Generalized string matching task which includes detection of substrings of text strings is also called fuzzy string matching task. This task is important for cases where errors are taken into account. Review of fuzzy comparison algorithms is provided in work [13-17].

Fuzzy search algorithms (also known as fuzzy string search) are the foundation of spell check and complete search systems such as Google or Yandex. There are many methods of analysis [18-25] and fuzzy (inexact) string matching [26-29]. The most popular of these fuzzy string matching methods are the methods of calculation of editing spacing [30-32]. Generally, metrics numerically calculating the value of transformation of one line to another is considered as spacing editing. There are different several operations, each of which can have a value of its own: character stuffing, deleting, replacement and transposition of proximate symbols. There are dif-

Copyright © 2013 SciRes.

ferent fuzzy string matching algorithms, which are based on different editing distances. Hamming distance—is a number of positions, in which corresponding symbols of two words of the equal length are different [33]. In more general cases, Hamming distance is applied to lines of same length of any q alphabets and serves as the difference metrics of objects of equal dimensions. Hammings distance is usually used in bio-informatics and genomics.

If matching of two strings of different lengths is allowed, then as a rule, insertion and deletion are also required. If they are given the same weight as replacement, minimal general value of transformation will be equal to one of the metrics proposed by Levenstein [34]. As a result, a more general task for a voluntary alphabet was associated with his name. Gasfield made a significant contribution in study of this issue [35].

Levenstein distance and its generalization is actively applied for correction of errors in words (in search systems, data bases, during text input, during automatic detection of scanned text or speech), for comparison of text files, in bio-informatics for comparison of genes, chromosomes and proteins. From application point of view, determination of distance between words or text fields according to Levenstein has following disadvantages:

- Comparatively large distances are made upon rearrangement of words or parts of words;
- Distances among completely different short words become small, while distance between very similar long words become significant.

As a result of analysis of different DNS data bases, it is known that, contents of records of same fields of registration data of domain names can be expressed in forms different (not identical) in content. The reason for occurrence of difference of field value might be lack of information, misprints, use of abbreviations, duplication of records etc.

Upon entry of domain name registration data in DW, abbreviations, misprints, omissions, double recordings and other distortions are encountered. In order to increase the quality of input registration data such as "registrar", person", "address", "organization", "admin-o" etc, prevention of errors and inconsistencies of duplications in records is required. For example, in names of countries (cities) misprints can be as Kanata (Canada), Russian (Russia), Frankfrut Am Main (Frankfurt Am Main) etc; organization identifier registration data (registrar) "MONIKER ONLINE SERVICES, INC." can be identified as "MONIKER, INC.", "MONIKER", "MONIKERS ONLINE SERVICES". In this case, words included in phrases must be processed separately. Despite difference of these strings, it is clear that, all of these titles stand for the same registrar. But let's also note that, upon comparing strings "MONIKER ONLINE SERVICES, INC." and "MONIKER, INC." using Levenstein metrics, we receive a larger value for editing distance. In cases where it is necessary to process the words contained in phrases separately from each other and/or apply algorithms that can compare similar lines for equivalence (for example, full entrance of one line into another as subsequence) and detect them as "similar".

In reviewed paper, let's use Damerau-Levenstein distance—difference measure of two strings of symbols, defined as minimal quantity of insertion, deletion, replacement and transposition (rearrangement of two proximate symbols) operations necessary for transfer of one string to another. It is the modification of Levenstein distance, and differs from it by addition of transposition operation.

Editing distance determined in such way can be calculated using dynamic programming method [36]. Also, there is an algorithms for this, requiring O(MN) operations, where M and N—are lengths of compared lines, and it is required to calculate MN of elements of so called dynamic programming matrix in order to find the distance value.

4.2. Symbol Fields for Registration Data Recording

As a rule, symbol fields consist of a string which contains one or several words, divided in gaps and punctuation marks. Nonstandard phrases are used in fields. Data is entered manually by an operator, often in distorted condition. In this regard, punctuation marks such as ".", ",", ":", " " ", "-" etc. that do not carry a functional significance, are replaced by "blank" sign.

Name of physical person (person), contact address of physical person (address), domain administrator identifier (admin-o), organization identifier for administrative communication (admin-c), organization title (organization) are the key fields upon identification of registration data.

Components of these fields can be present in random order, which is a difficult task for automatic processing of such information. In order to implement comparison of two lines containing such information, it is necessary to dismember each field to its contents, then compare only those which have identical meaning, for example in the address, compare the name of the city to the name of the city, name of the street to the name of the street. For content analysis, we enter so-called, samples. For example, sample address template is a combination of address components:

[apartment], [block], [building], [street], [city], [district], [index], [country].

For example, for domain—azerbaijan.info, sample address template is important:

, , 4676, Admiralty Way, Marina del Rey, California,

Copyright © 2013 SciRes.

90292-660, US.

Difference measure of attributes' values, for example such as contact address of a person will be calculated using Damerau-Levenstein distance

4.3. Decision Tree

Decision trees is the most comfortable decision making method for record (object) identification, for their demonstrativeness while use, minimal calculation resources and simplicity of realization. Decision tree—is one the methods of automatic analysis of vast amounts of data. Decision trees—is a method of rule presentation in hierarchic, consecutive structure, where each object corresponds to single knot that gives decision. Under rule, we understand a logical construction, presented in "if…then" form. Main advantages of decision trees are generation of rules in fields where experts formalize their knowledge with difficulty; extraction of rules in natural language; intuitively understandable classification model; high forecast precision, comparable to other methods (statistics, neural networks); construction of non-parametric models [37].

Suppose that we are given any educating set of T, containing objects, each of which is characterized by m attributes, while one of them points at affiliation of the object o a certain class. Construction idea of decision tree from T set, first expresses by Hunt, is demonstrated in accordance with R. Queenlan [38].

Let's label classes (values of class marks) through $\{C_1, C_2,\cdots, C_k\}$, then there are 3 situations:

- Set of T contains one or more examples related to one class C_k. Then decision tress for T—is a list, determining the $C_к$ class.
- Set of T does not contain any examples, *i.e*. it is an empty set. Then again, the list and the class associated with the list are selected from another set different from T, let's say from sets associated with the parent;
- The Set of T contains examples affiliated with different classes. In this case it is necessary to divide Set of T to some subsets. For this purpose, we choose one of the attributes that has two or more different values $O_1, O_2,\cdots, O_n$. T is divided into subsets of $T_1, T_2,\cdots T_n$, where each subset T_i; contains all examples that have a value of O_i for selected attribute. This procedure will be recursively continued until the final set will consist of examples related to the same class.

Above described procedure is the basis of many modern construction algorithms of decision trees. Obviously, upon using given methodology, construction of decision tree will be implemented from top downward. Currently, there is a significant number of algorithms, realizing decision trees CART, C4.5, Newld, ITrule, CHAID, CN2 etc [39,40]. But following two are the most widespread and popular: CART (Classification and Regression Tree) and C4.5.

Decision tree will be generated following way: for exact identification of the objects, we will use fields univocally identifying the object. Domain registration data will be identified by following attributes: registrar, person, address, admin-o, admin-c, organization, created, updated, free-date, phone, e-mail, nserver, type, source. Intermediate nodes of the tree correspond to these attributes, and arches—to possible alternative comparison values of these attributes "+" (same), "±" (similar), and "–" (different). Tree leaves are indicated as one of three classes as "=" (compared objects are identical), "≠" (objects are different), "?" (unknown). The example of the tree for identification of registration data domain names is provided on **Figure 1**.

Following are the most significant among attributes: registration identifier (registrar), full name of the physical person (person) contact address of the physical person (address), domain administrator identifier (admin-o), organization identifier for administrative communication (admin-c), title of organization (organization), list of DNS servers supporting domain (nserver).

Less significant, but as informative attributes are: domain registration time (created), domain update time (updated), domain free date (free-date), telephone numbers with international codes (phone), e-mail addresses (e-mail), domain type (type), domain registration payment time (paid-till), information source (source) etc.

Decision tree is formed based on the knowledge of experts of the subject field. Depth of the tree is selected based upon objects necessary for precise identification.

4.4. Software Implementation

Software implementation of identification algorihms of registration data of domain names is implemented on

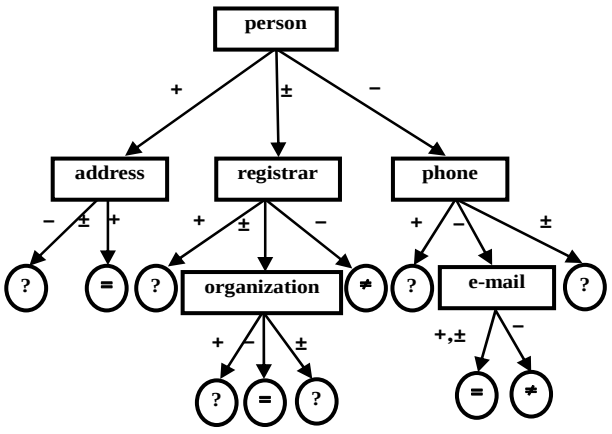


Figure. 1. Decision tree for identification of registration data of domain names.

Copyright © 2013 SciRes.

DRRacket language. Racket (former PLT Scheme) is the multi-paradigm programming language in Lisp/Scheme family, which also serves as a platform for creation of languages, design and realization. Programming language is known due to its extensive macro system, which allows creating built-in and subject-oriented languages, language constructions, such as classes or modules, as well as separate Racket dialects with different semantics. Distribution platforms of free and open software is spread under LGPL license [41-43].

We use Racket, in order to demonstrate that without crochets and other unused symbols, it is easy to understand complex program codes in programming language. Despite the fact that Racket language is simple and easy for perception, it is quite difficult to find errors committed in the program. a fraction of decision algorithm for identification of registration data in DRRacket language is provided in **Figure 2**.

For comparison of results, the identical algorithm is implemented in Python language which is used for solution of wide range of tasks of applied programming [44-47]. Same algorithm is presented in Python program on **Figure 3**.

Conducted experiments with above described decision tree gave the identical positive results in DRRacket and Pythin programs.

5. Conclusions

Modern approaches to solution of this task are related with construction of DW allowing "liberating" of information from strict frames of operating systems and better comprehend the problems of real activity.

DW provides high speed of data reception, ability to receive and compare, as well as consistency, completeness and authenticity of data. After withdrawal from the source, data are loaded into the warehouse, in order to implement cleaning and simultaneously provide comprehensive support of data cleaning [48].

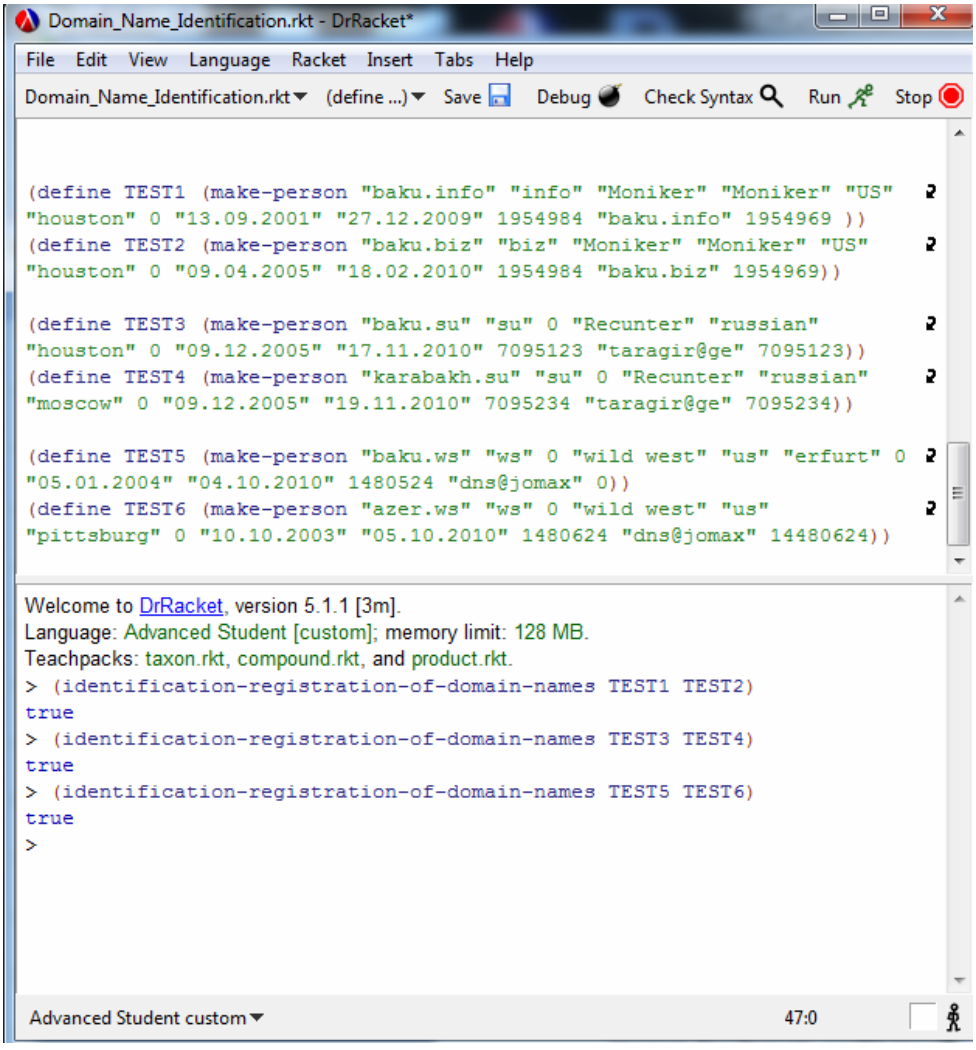
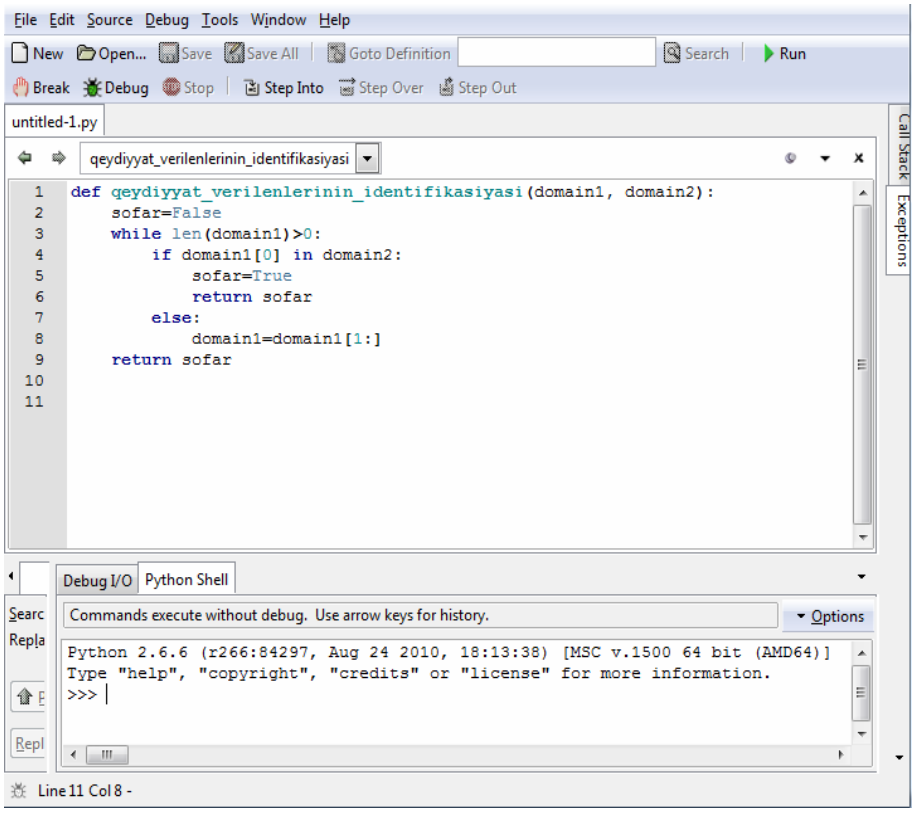
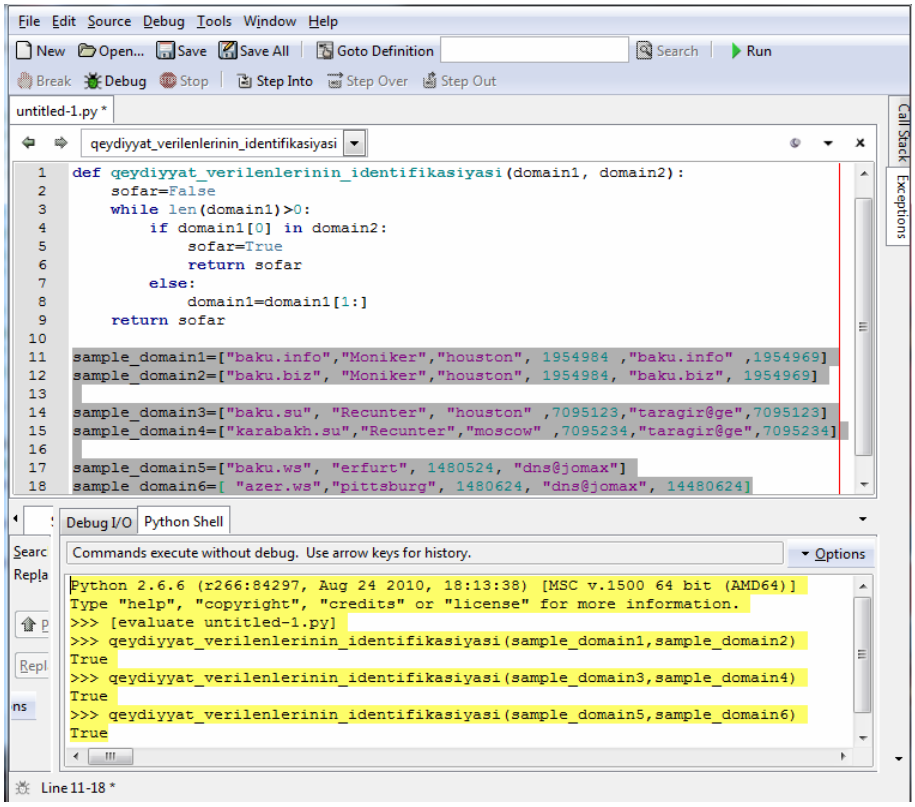


Figure 2. Decision algorithm for identification of registration data of domain names in DRRacket program.

Copyright © 2013 SciRes.



(a)



(b)

Figure 3. (a, b) Decision algorithm for identification of registration data of domain names in Python program.

Copyright © 2013 SciRes.

Domain names registration data category identification was reviewed in this work. Abbreviations, misprints, omissions, conscious data corruption, record duplicates etc which were allowed on information collection stage are considered as errors. Currently, variety of alternative proximity functions were proposed, but from our point of view, in order to conduct clearing and consistency checking, Damerau-Levenstein distance most accurately corresponds to intuitive similarity concept. Also, domain name registration data records identity method based on decision tree was proposed.

Developed identification method was implemented for creation of a DW on information from several DNS Servers based on the example of domain name registration data serving the interests of the Republic of Azerbaijan.

REFERENCES

[1] E. A. Trachtengertz, "Evolution of Administrative Decision Making Support Systems of Computers," *Information Technologies*, No. 1, 2006, pp. 15-31.

[2] I. P. Belyayev, "Art of Data Analysis," *Information Technologies*, No. 5, 2003, pp. 31-37.

[3] V. V. Przhiakovski, "Complex Analysis of Large Volume Data: New Perspectives of Computerization," *SUBD*, No. 4, 1996, pp. 71-83.

[4] A. A. Sakharov, "Construction Concept and Realization of Information Systems Oriented on Data Analysis," *SUBD*, No. 4, 1996, pp. 55-70.

[5] W. H. Inmon, "Building the Data Warehouse," 2nd Edition, John Wiley & Sons, Inc., New York, 1993, p. 298.

[6] W. H. Inmon, "Building the Data Warehouse," 3rd Edition, John Wiley & Sons, Inc., New York, 2002, p. 412.

[7] W. H. Inmon and R. D. Hackathorn, "Using the Data Warehouse," John Wiley & Sons, New York, 1994, p. 285.

[8] R. Kimball, "The Data Warehouse Toolkit," John Wiley & Sons, New York, 1996, p. 388.

[9] R. Kimball and M. Ross, "The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling," 2nd Edition, John Wiley & Sons, New York, 2002, p. 464.

[10] F. J. Damerau, "A Technique for Computer Detection and Correction of Spelling Errors," *Communications of ACM*, Vol. 7, No. 3, 1964, pp. 171-176. doi:10.1145/363958.363994

[11] A. G. Sergo, "Domain Names," Bestseller, Moscow, 2006, p. 368.

[12] A. A. Venedrukhin, "Domain Wars," Piter, Saint Petersburg, 2009, p. 224.

[13] G. M. Landau and U. Vishkin, "Fast String Matching with k Differences," *Journal of Computer and System Sciences*, Vol. 37, No. 1, 1988, pp. 63-78. doi:10.1016/0022-0000(88)90045-1

[14] G. M. Landau and U. Vishkin, "Fast Parallel and Serial Approximate String Matching," *Journal of Algorithms*, Vol. 10, No. 2, 1989, pp. 157-169. doi:10.1016/0196-6774(89)90010-2

[15] Z. Galil and R. Giancarlo, "Data Structures and Algorithms for Approximate String Matching," *Journal of Complexity*, Vol. 4, No. 1, 1988, pp. 33-72. doi:10.1016/0885-064X(88)90008-8

[16] A. N. Borisov and A. V. Alekseyev, "Processing of Fuzzy Information in Decision Making Systems," Radio and Communication, Moscow, 1988, p. 304.

[17] A. Y. Solodkov, "Object Identification Algorithms in Data Bases," *System Integration*, Vol. 12, No. 90, 2004, pp. 52-56.

[18] A. E. Yermakov and V. V. Pleshko, "Parsing in Statistical Text Analysis Systems," *Information Technologies*, No. 7, 2002, pp. 15-17.

[19] L. A. Zade, "The Concept of a Linguistic Variable and Its Application to the Adoption of Approximate Solutions," Mir, Moscow, 1976, p. 167.

[20] K. Iberla, "Factoral Analysis," Statistics, Moscow, 1980, p. 398.

[21] B. Carnigan and P. Pike, "Programming Practice," Nevsky Dialect, Saint Petersburg, 2001, p. 288.

[22] T. Connolly, K. Begg and A. Strachan, "Data Bases. Design, Realization and Maintenance. Theory and Practice," Williams, Moscow, 2000, p. 1120.

[23] T. H. Coremen, C. I. Laserson, R. L. Rivest and K. Stein, "Algorithms: Structure and Analysis," 2nd Edition, Williams, Moscow, 2005, p. 1296.

[24] A. Coffman, "Introduction to Fuzzy Sets Theory," Radio and Communication, Moscow, 1982, p. 432.

[25] O. I. Larichev and E. M. Moshkovich, "Qualitative Decision Making Methods," Fizmatlit, Moscow, 1996, p. 217.

[26] A. Ehrenfeucht and D. Haussler, "A New Distance Metric on Strings Computable in Linear Time," *Discrete Applied Mathematics*, Vol. 20, No. 3, 1988, pp. 191-203.

[27] U. Masek and M. S. Peterson, "A Faster Algorithm for Computing String-Edit Distances," *Journal of Computer and System Sciences*, Vol. 20, No. 1, 1980, pp. 785-807. doi:10.1016/0022-0000(80)90002-1

[28] P. H. Sellers, "The Theory of Computation of Evolutionary Distances: Pattern Recognition," *Journal of Algorithms*, No. 1, 1980, pp. 359-373.

[29] E. Ukkonen, "Approximate String Matching over Suffix-Trees," *Proceedings of the 4th Annual Symposium on Combinatorial Pattern Matching*, Padova, 1993, pp. 229-242. doi:10.1007/BFb0029808

[30] D. M. Sunday, "A Very Fast Substring Search Algorithm," *Communications of the ACM*, Vol. 33, No. 8, 1990, pp. 132-142. doi:10.1145/79173.79184

[31] A. Hume and D. Sunday, "Fast String Searching," *Software—Practice and Experience*, Vol. 21, No. 11, 1991, pp. 1221-1248. doi:10.1002/spe.4380211105

[32] A. V. Aho, "Algorithms for Finding Patterns in Strings," In: J. van Leeuwen, Ed., *Algorithms and Complexity, Handbook of Theoretical Computer Science*, Elsevier Sci-

Copyright © 2013 SciRes.

ence Publishers, Amsterdam, 1990, pp. 255-300.

[33] R. W. Hamming, "Coding and Information Theory," Englewood Cliffs, Prentice-Hall, Upper Saddle River, 1980, p. 239.

[34] V. I. Levenstein, "Binary Codes Capable of Correction Deletions, Insertions and Reversals," *Soviet Physics Doklady*, Vol. 163, No. 4, 1965, pp. 845-848.

[35] D. Gasfield, "Lines, Trees and Sequences in Algorithms," *Informatics and Computational Biology*, NEvsky Dialect, Petersburg, 2003, p. 654.

[36] R. A. Wagner and M. J. Fischer, "The String-to-String Correction Problem," *Journal of the ACM*, Vol. 21, No. 1, 1974, pp. 168-173. doi:10.1145/321796.321811

[37] O. G. Berestneva and E. A. Muratova, "Construction of Logical Models Using Decision Tree," *Tomsk Polytechnic University News*, Vol. 207, No. 2, 2004, pp. 55-61.

[38] J. R. Quinlan, "Generating Production Rules from Decision Trees," *Proceedings of the* 10*th International Joint Conference on Artificial Intelligence*, Morgan Kaufmann, 1987, pp. 304-307.

[39] J. R. Quinlan, "C4.5 Programs for Machine Learning," Morgan Kaufmann Pub., San Mateo, Vol. 240, 1993, p. 302.

[40] J. R. Quinlan, "Improved Use of Continuous Attributes in C 4.5," *Journal of Artificial Intelligence Research*, Vol. 4, 1996. pp. 77-90.

[41] R. B. Findler, C. Flanagan, M. Flatt, S. Krishnamurthi and M. Felleisen, "DrScheme: A Pedagogic Programming Environment for Scheme," *Proceedings of the International Symposium on Programming Languages*: *Implementations*, *Logics*, *and Programs*, September 1997, pp. 369-388.

[42] R. B. Findler, J. Clements., C. Flanagan, M. Flatt, S. Krishnamurthi, P. Steckler and M. Felleisen, "DrScheme: A Programming Environment for Scheme," *Journal of Functional Programming*, Vol. 12, No. 2, 2002, pp. 159-182.

[43] S. Manuel and J. B. Hans, "Understanding Memory Allocation of Scheme Programs," *ACM SIGPLAN International Conference on Functional Programming*, Vol. 35, No. 9, 2000, pp. 245-256.

[44] M. Lutz, "Programming in Python," 4th Edition, Symbol-Plus, Saint Petersburg, 2011, p. 992.

[45] M. Dawson, "Programming in Python," Piter, Saint Petersburg, 2012, p. 432.

[46] I. A. Khahaev, "Workshop on Algorithms and Programming in Python," *Textbook*, Alt Linux, Moscow, 2010, p. 126.

[47] D. M. Beesley, "Python," 4th Edition, *Detailed Directory*, Symbol-Plus, Saint Petersburg, 2010, p. 864.

[48] E. Spearley, "Corporate Data Warehouses. Planning, Development, Realization," Vol. 1, Williams Publishing House, Moscow, 2001, p. 395.

Copyright © 2013 SciRes.