Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 102 (2016) 121 – 128

Procedia Computer Science

12th International Conference on Application of Fuzzy Systems and Soft Computing, ICAFS 2016, 29-30 August 2016, Vienna, Austria

Linguistic resumes in software engineering: the case of trend summarization in mobile crash reporting systems

Konstantin Y. Degtiarev [a]*, Nikita V. Remnev [a,b]

[a] *National Research University Higher School of Economics (HSE), Faculty of Computer Science, School of Software Engineering, 3 Kochnovsky pass., 125319, Moscow, Russia*

[b] *Ru-Beacon (Empatika company's project)*

Abstract

The construction of time series linguistic summaries is a topic that draws attention of researchers for many years. The full-fledged software implementation (the pilot web-application) that supports the complete process of linguistic summarization of time series construction is presented in the paper. The program can be used in professional groups for discussions and rapid data analysis. Virtual mobile crash reporting system (MCRS) supplies the test input data used as an example.

© 2016 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Peer-review under responsibility of the Organizing Committee of ICAFS 2016

Keywords: Time series; linguistic summary; software engineering; software development; summarizer; validity of linguistic summary; fuzzy set; linguistically quantified proposition; mobile crash reporting system; analysis of trends; graphical reports; web-application

1. Introduction

In the age of information technology's impetuous progress that influences daily human activity, comprehension, analysis and processing of intensive data flows assume ever greater importance. The power of such flows compels people to resort to the help of means to extract factual entities and construct summary based on domain-specific information brought to their notice via *organa sensuum*. The important fact of obtaining the information should be supplemented with a sole aspect of what we derive from it. Such findings may facilitate perceptibly different types

* Corresponding author. Tel.: +7-495-772-9590.
E-mail address: kdegtiarev@hse.ru

1877-0509 © 2016 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
Peer-review under responsibility of the Organizing Committee of ICAFS 2016
doi:10.1016/j.procs.2016.09.378

of routine tasks related to grasping the meaning and key points of problems, predicting the future, decision-making, and so forth. Summarization of data is ingenuously associated with the ability of humans "to *communicate* observations of the world in a useful and comprehensible manner" that is convenient for use by both individuals and companies[13,14].

According to Merriam-Webster dictionary, the word ′**summary**′ is defined as "using few words to give the most important information about something". The construction and use of such summaries accompany any kind of human activity, and thriving fields of IT and software engineering are not an exception. Multifarious human activity would not be possible without efforts aimed at the development of software to run on hardware platforms of mobile devices, desktop computers and servers. In the field of software engineering, data summarization plays significant and useful role in respect of construction of *units of communication*[13] used in discussion of project's details with heterogeneous groups of people embracing various types of stakeholders, developers, testers of software products.

It is natural that the data (information) in use are always connected with a time factor that has a significant impact on the conclusions derived on the basis of such information. Timeline has a great importance in information handling, since it sets up a base to explain association with events that may influence these changes. Definitely, available data can be visualized by graphs giving proper account to time stamps, if any. Graph's granularity as a main drawback of such representation may conceal from view important, but quite short periods of changes that make the findings not complete. This fact lays the basis for the whole research area to deal with the construction of time series linguistic resume, bearing in mind that natural language is mainly "a system for describing perceptions, which are intrinsically imprecise", therefore most of phrases and sentences formulated in natural language are fuzzy[15].

Software development process is complex multiphase sequence of actions (with repetitive back-offs to previous steps, if needed) appreciably linked to human contacts and diverse descriptions in natural language. Problem analysis, preparation of specifications, planning of development steps and definition of software architecture can be designated as revealing examples. The output of development process is the software product, the quality of which can be good, endorsed by users, but generally not perfect. Development teams have to gather information about the state of the program after its deployment, problems that occur during the runtime, issues that cause failure (crash) – e.g. the number of crashes, backtraces of the process's threads, usage of CPU resources, etc. Nowadays, the collecting and storage of crash reports are fully automated. The analysis of reports pursues the long-term object to obtain accurate interpretation and promptly fix the origin causing the crash. It seems reasonable to conjecture that the application of algorithms for constructing linguistic resume of time series may facilitate routine tasks of developers and influence positively the time management of software maintenance.

Crash reporting systems consist of two main modules, which ensure operation of the whole system – the first one is a built-in mobile application module to detect critical errors and to send error reports (codes, etc.) to server system. The latter being the second module of the system analyzes reports and presents processed information as graph(s). Systems per se are not engaged in further discussion – they are simply treated as ′black boxes′ that receive errors as inputs and present information relating to errors in the graphical form. Analysis of such graphs can be considered as a key constituent of *exploratory data analysis* (EDA) aimed at visual revealing of patterns.

Graphical reports are rather attractive, but they suffer from evident shortcomings mentioned in passing above. The information concerning errors usually consolidates several features that cannot be displayed on the same graph discernibly. Users having little or no experience with software development can be in a predicament to interpret data, to understand data format as well as the meaning of constituent features of the error report shown to him (her).

The construction of short linguistic summary on the ground of obtained data in respect to crash experienced in practice seems to be clear and extremely convenient way out for the user of such system. The rest of the paper is organized as follows: at first (Section 2), selected core research studies on time series linguistic summarization (TSLS) and related topics are briefly discussed. Following basic publications by renowned scientists L. Zadeh, R. Yager, J. Kacprzyk and S. Zadrożny, Sections 3 and 4 cover some general ideas as well as distinctive features relating to the process of TSLS construction; the importance of summaries in various situations associated with the field of software engineering is also elucidated here. Section 5 is devoted to the description of the scope of pilot web-based application aimed at construction of linguistic summaries. Conclusion and final remarks are drawn in Section 6.

2. Linguistic Summarization. Related Work

One of the first papers entirely dedicated to the description of linguistic summary essence belongs to R. Yager[1]. The paper presents the way to summarize data set in the form of linguistic values that can be quantified as fuzzy

subsets X_i of a given base set U. Linguistic values or words that constitute quantified sentences can be associated with X_i expressing their meaning as membership grades $\mu(x)$ of the element $x \in X_i$. Each summary[2] is also characterized by the quantity of agreement *Q* (the proportion of data that satisfy restrictions imposed by summarization statement) coupled with calculated summary's truth value *T*. Properties of summarization and informativeness of a summary in respect of general data sets are also discussed[13] by Yager R., Ford K. and Cañas A. The authors discuss the system architecture of summarizer as a potential method "merged" to typical for databases query environment. Kacprzyk J. and Zadrożny S. consider[16] the construction of summaries of large data sets in the form of quantified propositions (see R. Yager[13]). Certain elements and relations of linguistic summaries cannot be obtained automatically; the finding substantiates the idea to switch over to construction of summary in interactive manner allowing to specify class of summaries with a following check of database against such user's request to pick out the "best" possible variant.

Any authentic business activity is rigidly related to processes of incessant making decisions based on perception of the current situation and existing constraints. It is noted[13,14,17] that linguistic summaries may act as a contracted verbalization units adequately perceptible by all parties involved into problem solving.

Besides, Kacprzyk J. and Yager R.[18] discuss calculation of validity (truth value) of linguistic summaries in their basic form (linguistically quantified propositions that constitute the summary) and cover some approaches to derive validity criteria, i.e. truth of the statements having general unweighted form "*Q*′_of_′<object>′s_are_′F", <object> := y_i; $Y=\{y_i\}$ – a set of objects that form a basis of descriptions at hand (e.g. "*project*", "*employee*"). Symbol *Q* denotes a linguistic quantifier (e.g. "*most*"), F is a property (e.g. "*underestimated*") possessed by the object. The form shown above can be augmented with importance factor B ("*Q*′_of_′B′_′<object>′s_are_′F"), where B is a term that specifies the weight of object(s), or a sign marking out some objects among others (e.g. "*core*", "*important*", etc.). In the first place the discussion covers the measure of informativeness of a summary[1,13] and the use of genetic algorithms for linguistic summaries mining[20]. Relying on several objectives (accuracy, coverage of the whole time series length, brevity) characterized by different degrees of significance, Castillo-Ortega R., Sánchez D., Marin N. and Tettamanzi A. adopted a non-dominated genetic algorithm to deal with specificities of linguistic summarization process[22].

In the context of topic's subject, time series deserves a separate mentioning; the term itself stands for a sequence of observations (values) collected sequentially in time – in many cases these values are bound to equally spaced time lags[24]. Since time series is a subject for research in different fields related to data processing, the works of Kacprzyk J., Wilbik A., Zadrożny S.[25,26] are devoted just to deriving of time series summaries on the basis of trends observed in data. The foundation of the approach displays itself through calculation of linguistically qualified propositions that utilizes fuzzy sets as core of computing with words (CWW) paradigm and human perceptions modeling. The resort to compact verbal forms, according to neat remark by R.Yager, favors "more structured use of words" and reduces the loss of information in daily communication, or at the human-machine interface supporting operations with words to a certain extent[27]. Trends exposed in time series can be expressed as straight lines being single parts of piecewise-linear approximation of time series values. The set of distinctive features (slope of the line, length of time period, during which the identifiable trend has been observed, etc.) may serve as appropriate trend's characteristic.

In addition, Kacprzyk J., Yager R., Zadrożny S. and Wilbik A. review the construction of linguistic summaries of data sets (databases)[3,4,5,6]. Authors give special value to two cases to define summaries, namely, static and dynamic ones. The latter implies the analysis of peculiarities of revealed trends connected with some data attributes, types of observed behavior. Kacprzyk J. and Zadrożny S. examine extensively the verbalization of the results of Web server logs analysis by means of linguistic summaries embracing both static and dynamic cases[7,19].

Talking about the construction of time series linguistic resume, the paper by Alvarez-Alvarez A., Sanchez-Valdes D., Trivino G., Sánchez A., Suárez P.D.[11] on the compilation of reports regarding situations on the roads is worth mentioning. Also, Castillo-Ortega R., Marín N. and Sánchez D.[8] consider the construction of linguistic resume for widely used nowadays *data cubes* ensuring rather handy and flexible access to data (multidimensional model) being summarized. The model grounded on extraction from time series messages to express elicited pieces of knowledge and its semantics (viz. *protoforms* and their instances), transformation of messages into text, and quality framework to ensure that the text suits the needs of the target audience is discussed by Marín N. and Sánchez D.[21]

Time series data mining based on human perceptions represented as verbal constructs ("*big number of errors*", "*sharp increasing*", etc.) is covered by Batyrshin I.Z. and Sheremetov L.B.[23]. Since the meaning of perceptions can be precisiated differently, much depends on peculiarities of linguistic description and objectives of a given task. The research in the field of human perception modeling enables to consider construction of resume in the context of time series decision-making based on precisiation of perceptions and its use in verbal patterns, rules or relations[23].

3. Time Series and Linguistic Summaries. Linguistically Quantified Propositions

The field of software engineering is linked to development and maintenance of complex programs composed of a big number of interrelated components. The software project development process that stipulates precise planning observing established standards and avowed procedures usually lasts over a long period of time. In accordance with the framework for software measurement validation[28], projects and software as generalized entities (E_1 and E_2) of the framework possess certain attributes – under all existing distinctions between collections of such attributes for each particular entity, we can denote hypothetical set of attributes as $\{a_i\}$, $i=\overline{1,k}$, for the reference purposes. In some cases attributes can be measured, i.e. at each moment of time t_j, $j=\overline{1,p}$, certain attribute a_i can be associated with some characteristic (e.g. number n_j) on a chosen unit scale. As an example one can mention the number of discovered errors during software testing (per diem), the number of recorded software crashes (per hour, per diem), etc. In other words, the matter is about *time series data*, a set of collected values at time points t_j, $j=\overline{1,p}$.

Even relatively small software projects imply the execution of scheduled amount of work by the team. Thorough planning of the development stages and the establishment of effective communication between team members are major constituents of successful project activity. Holding discussions and presentations (seminars) within the team, with stakeholders requires the use of various numeric data, graphs (e.g. graph representation of time series) coupled with *brief verbal summaries* of data in the capacity of utterly appealing and well-comprehended forms of expounded material. IT and software engineering are among those emerging fields where linguistic summaries as messages that co-opt information in concentrated form can be in called-for status since people having different, or even conflicting, interests, levels of expertise, etc. get involved in activities (e.g. requirements elicitation, analysis, software validation) employing mixed data with prevailing verbal information in conjunction with more discrete number-based chunks.

In this paper we deal with linguistic summary of the information (time series) following the works[1,3,6]. In many cases people have to deal with collections of data entities that are tangled to be understood directly. There is a need to extract from database succinct and concise templates (quantified propositions) as introduced by L. Zadeh[2].

The construction of linguistic summary (resume) of data provides for using a set of objects $Y=\{y_1, y_2, ..., y_n\}$ (by way of general example, we may conceive of software products), and a set of attributes $A=\{A_1, A_2, ..., A_m\}$ characterizing Y – the number of critical errors in the code or the version of the software product can be potential options to consider[26]; particularly, the value $A_j=A_j(y_i)$ of j-th attribute may stand for the number of critical errors in software product (code) y_i, $i=\overline{1,n}$. Linguistic summary is based on three principal constituents. The first one is a summarizer S that represents an attribute endowed with the meaning (linguistic value) defined on the domain of the attribute (e.g. A_{j1} ="*the number of critical errors*" having the value "*small*"). If the summary contains several similar constructions, then the main attribute with the value is treated as summarizer, and optional attributes are regarded as qualifiers (R). For instance, linguistic resume (LS) "*most old versions of software product have small number of critical errors*" can be rewritten in terms of introduced terms as "*most* $R\,A_{j2}(y_i)$ *have* S", where S is summarizer for the attribute A_{j1} shown above; it is coupled with the qualifier R = "*old*" for the attribute A_{j2} = "*version of software product*", i.e. "version of" $(y_i)=A_{j2}(y_i)$, $j1, j2=\overline{1,m}$. Since any particular attribute A_j works in inseparable liaison with an object y_i, the statement "*most* $R\,A_{j2}(y_i)$ *have* S" can be rewritten in the form "*most* $R\,y$'s *have* S" (software products in plural are denoted by *y*'s). It should be also mentioned that linguistic summary concerning the software product LS_1 = "*most of versions have small number of critical errors*" differs from the resume LS_2 = "*most of old versions have small number of critical errors*" on account of presence of the qualifier "*old*" for the attribute "*version of software product*" in the latter (note: j1 and j2 indices are treated here as compound single ones allowing simply to discern the designations of attributes used in the text).

The second constituent to mention has to do with the quantity in agreement Q, i.e. linguistically expressed indicator "of the extent to which the data satisfy the summary"[4]; quantifier "*most*" as a relative case of Q is a sample that fall within the essence of the provided explanation. Validity T is the third component of linguistic summary[25] – it

is the number in [0,1]-range stating the truth of linguistic summary. The closer the value of validity parameter to 1, the closer the statement to "true" status. Usually resumes with the highest values of *T* are of primary interest.

The basis of the summaries cited as example above is linguistically quantified proposition considered by Lotfi A. Zadeh[2]. Particularly, the statement LS_1 = "*most of versions of software product have small number of critical errors*" can be generalized as "*Q y*'s *are S*", whereas LS_2 – "*Q R y*'s *are S*" (′are′ ↔ ′have′).

4. Construction of Time Series Linguistic Summaries. General Ideas

The method of linguistic resume time series construction was suggested and developed minutely by Kacprzyk and Zadrożny[12]. The core of this approach is the segmentation of time series, i.e. linguistic resume is constructed not on the basis of data set per se (individual points), but proceeding from the segments (or, alternatively, *trends*) as "linearly increasing, stable or decreasing functions"[26] that allow to represent a sequence of points in the form of piecewise linear function. Individual sections of this function are characterized by divergent lengths and slopes. Generation of piecewise linear function may be performed in different ways, thus opening a way to consider groups of algorithms described by Keogh, Chu, Hart and Pazzani[9,10]. They cover three main algorithms (*sliding windows*, *top-down* and *bottom-up*) to perform segmentation of time series. Stopping criteria or incorrect border as "repeat-until" condition of algorithms is a method to calculate the error (segment's quality) as follows: best-fit line for points in segment is drawn, and the sum of distances between points in segment and line is calculated.

Trends (or, segments) obtained from time series have three basic characteristics – namely, (a) *dynamics of changes* that conforms to the speed of change (linguistic variable) in time series consecutive values, (b) *duration* that is the length of time series trend, which is also a linguistic variable, and (c) *variability* that describes the representation of data combined into the segment (actually, it's a set of linguistic labels, e.g., "*quickly increasing*", "*increasing*", "*constant*", and so on associated with fuzzy sets). Peculiarities of trends can be expressed in the form of membership functions. The dynamics of changes is estimated by line's slope (slope ratio) that passes through first and last points of the segment, the gradient (the angle α formed by the line and the positive direction of abscissa axis) enables to define the value of linguistic variable describing given segment. For example, for characteristic *′dynamics of changes′* the following values can be suggested: { $u(\alpha)$ = *quickly increasing* , if $\alpha \geq 70°$ }, { $u(\alpha)$ = *increasing* , if $50° \leq \alpha < 70°$ }, … (values *slowly increasing*, … and *slowly decreasing*) … , { $u(\alpha)$ = *decreasing* , if $-70° \leq \alpha < -50°$ }, { $u(\alpha)$ = *quickly decreasing* , if $\alpha < -70°$ }. The values of α should not be treated as absolute ones, so long as special features of the problem at hand, data entities, etc. may give cause to alter cited suggestion. On the other hand, it's not worthy to diverge from the empirical psychological ′magic′ threshold 7±2 related to human's ability to distinguish efficaciously given number of classes (or, terms) while processing trend information ($u(\alpha)$ values shown earlier).

To construct linguistic summary, templates proposed by Kacprzyk and Zadrożny[12] are used; these templates are based on linguistically quantified propositions (LQP)[2]. In particular, the summary "*Among all trends, most are quickly decreasing*" relies on the simple template "*Among all segments, Q are S*", whereas for "*Among all short trends, most are quickly decreasing*" template "*Among all R segments, Q are S*" containing qualifier *R* (we can talk about enhanced version of the preceding template) serves as a base. In this case the linguistic quantifier (*Q*) that corresponds to the "*number of segments*" term can be associated with several habitual and properly perceived graded quantifiers ("*most*", "*least*", etc.). In case of time series linguistic resume, their components, i.e. qualifier *R* and summarizer *S* represent trend characteristics. In the summary "*Among all trends, most are quickly decreasing*" summarizer *S* embodies the dynamics of changes observed in position of segments. Qualifier *R* in the summary "*Among all short trends, most are quickly decreasing*" states the duration (length of the trend).

Information contained in linguistic resume carries important dedicated messages to specialists in the knowledge domain. The truthfulness (verity) *T* peculiar to resume as the number from the unit interval is a valuable information batch that is attached to the resume. Calculation of the truthfulness of summary is carried out in the following way (*S* and *R* are fuzzy sets in *Y* represented by membership functions $\mu_S(y_i)$ and $\mu_R(y_i)$, correspondingly)[4,25]:

$$T\left(\textit{Among all } y\text{'s, } Q \textit{ are } S\right) = \mu_Q\left(\frac{1}{n}\cdot\sum_{i=1}^{n}\mu_S(y_i)\right),\ T\left(\textit{Among all } Ry\text{'s, } Q \textit{ are } S\right) = \mu_Q\left(\frac{\sum_{i=1}^{n}\mu_S(y_i)\wedge\mu_R(y_i)}{\sum_{i=1}^{n}\mu_R(y_i)}\right)$$

The calculation of truth value *T* of each linguistic resume is accompanied in the sequel by ranking results obtained with the view of revealing one or several summaries with the highest *T*'s values (or, with those values of *T* that exceed a given trust threshold; in the program a simple slider can be used to control the threshold value).

5. Plain visualization of data vs. construction of linguistic summaries. Software implementation

The process of data sets' linguistic summaries construction is based on using mathematical methods that lead to obtaining a set of quantified propositions, which are of the form of short enough, simple or slightly more complex expressions in natural language. In the present case, *formal computational schemes* aimed at analysis of data with due regard for time dimension within the scope of specific problem and *resultant verbal constructs* are brought together. This row has to be widen at the expense of *graph representation* of time series. Such "graphic portrayal" of data sequence does not allow to accentuate the attention on fragments of the graph that require special attention or extra analysis. At first sight, the use of accompanying text or elucidations on the graph may help to understand key trends. However, their effect can be opposite though owing to superfluous overburdening the graph.

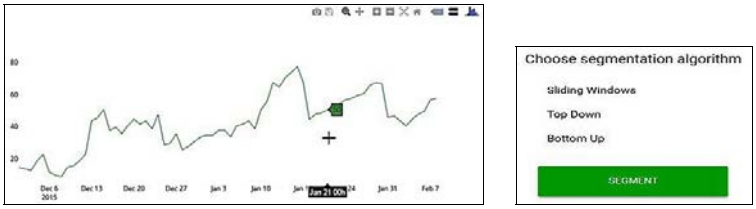

Fig. 1. Visualization of data (70 points, real data covering 2+ months) and the choice of segmentation algorithm.

Taking into account the prospects and applied focus of linguistic summarization topic, the full-fledged software implementation of algorithms mentioned above is of vital practical interest. Despite of active research into the theme in recent years, programs that support convenient and customizable realization of such algorithms is not available, to the best of authors knowledge. The task of time series linguistic resume construction corresponds to two clauses that have to do with (A) time series analysis by means of segmentation, and (B) calculation of summaries. With respect to computer programs, these subtasks are normally implemented separately, although even light version of subtask's (B) software implementation is hard to found, if possible at all. It always looks attractive to have a convenient tool that can be used in professional groups for discussions and rapid data analysis engaging all members of such groups. With this idea in mind, the pilot web-based application making use of Google App Engine platform and written in Python was prepared in the School of Software Engineering @ HSE. Modern web-technologies for client and server-side application parts, including framework AngularJS, libraries Angular Material, Plotly.js, Highcharts, jQuery, were brought into play in development process. The program supports the complete process of construction of time series linguistic summaries, starting from data setting and ending with obtaining linguistic resume proper.

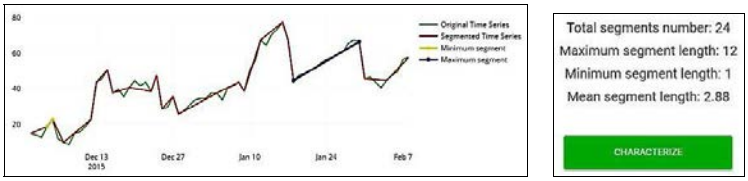

Fig. 2. Results of time series segmentation using sliding windows algorithm.

The whole operation loop covers all required stages of data processing and includes (1) downloading time series data, (2) choosing time series segmentation algorithm to run, (3) displaying the results of segmentation, (4) defining basic characteristics of segments (dynamics of changes, variability and duration), and (5) constructing linguistic summaries based on specified characteristics and displaying them in descending order of validity *T* (truthfulness).

For example, we may consider time series that covers the number of identified critical errors per day in some hypothetical software product – real data (dates and the number of errors revealed by the team at particular date), 70

points, uploaded from external file as visualized in Fig. 1. After choosing segmentation algorithm (here the choice falls on Sliding Windows option, Fig. 1), the results of segmentation are displayed as shown in Fig. 2 (several time series can be shown at the same time; easy-to-use mechanism is provided to support quick exclusion and back-off of graph within the view form). Selected fragments of graphs can be magnified and zoomed out as needed with the aid of mouse key combinations. The program also displays explanations regarding performed segmentation – namely, total number of segments, min and max lengths of segments, the average (mean) length of segment (Fig. 2).

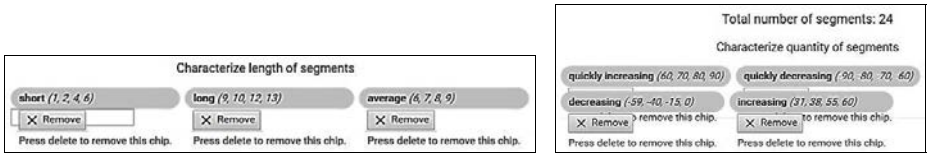

Fig. 3. Definition of lengths of segments and dynamics of changes as linguistic values. Choice of variability metric.

Once results of segmentation are shown, a user can evaluate segments obtained. This stage has direct reference to construction of linguistic resume. Specification of required characteristics (trapezoidal fuzzy numbers are used) is the most time-consuming stage of working with the program, none the less, the whole process becomes simpler because some pieces of statistical data based on the results of segmentation are displayed in the same window having well-engineered layout.

At the next step, a user is invited to estimate (specify values of parameters of membership functions representing input linguistic labels – e.g. "*short*", "*long*") lengths of segments, the number of segments and dynamics of changes observed (Fig. 3). To evaluate the dynamics of segments change, a diagram showing the percentage of angle ranges observed between any two points of time series is also displayed (Fig. 3). The program allows a user to choose the most convenient method to calculate segments variability – statistical metrics, such as difference between minimum and maximum, variance, square root of the variance and the mean absolute derivation are offered. Extra functionality allows simplification of the work with linguistic summaries; a user has a possibility to specify the minimum limit value (truth value filter) for the validity of summaries and also search for resultant summaries using text fragments.

For example, after choosing Top-Down segmentation algorithm (consider this case without stinting ourselves just to the option mentioned above), the program may display the following results (with the reference to linguistic terms and parameters of trapezoidal membership functions: length – "*long*" (6, 7, 10, 12), "*short*" (0, 1, 3, 6); number of segments – "*most*" (14, 16, 26, 28), "*average*" (7, 9, 12, 15); dynamics of changes – "*increasing*" (50, 60, 85, 90), "*decreasing*" (-90, -85, -60, -55); variability – "*low*" (0, 5, 15, 20), "*high*" (20, 30, 60, 70)): maximum segment length – 12, minimum segment length – 1, … , maximum value – 77 (as shown in Fig. 2 and 3). Dynamics of changes are expressed in the form: from 61° to 90° – 52.86%, from 31° to 60° – 14.29%, …. , from -59° to -30° – 5.71%, from -90° to -60° – 22.86% (Fig. 3).

As a result of TSLS, the following list (its part that covers summaries with the highest validity *T* value) is generated by the web-application – just a sample is shown: { Most of trends are short **100.00%** }, { Most of short trends with low variability are increasing **100.00%** }, … , { Average of short trends are increasing **52.08%** }, … . Comparing time series graph, segments obtained and constructed summaries, it could be noticed that the results quite accurately reflect the overall data appearance. Proper and clearly checked choice as well as further definition of linguistic values give an opportunity to bring summaries closer to user's (parties concerned within the project) perception of both observed situation in whole and particulars attracting attention. The modified version of the program could also deal with additional characteristics of segments, such as time dimension to obtain results related to specific time period(s), or complexity of linguistic summaries – the latter requires further theoretical elaboration on algorithms in use. Special data features could also be included in summaries as text fragment(s).

6. Conclusion

The developed and presented here web-application should be considered not in the capacity of ordinary training program, but as a tool of expert time series processing seeking to construction of linguistic summaries with the employment of main algorithms (*sliding windows*, *top-down*, *bottom-up*) proposed up to now.

The utility and appeal of linguistic summaries are related to decent enough transparency of results obtained that is important in project activities that bring together different groups of people. This is just covers the case of software-

intensive systems development and use, where linguistic resume can be treated as a way of concentrated verbal description, though under certain constraints and simplifications, of observable cases and phenomena to be partly classified as data-driven. The improvement (modification) of application software oriented towards segmentation of time series and construction of summaries should be done “hand in hand” with broadening of theoretical research into the field of linguistic summarization as elemental of the growing Computing with Words (CWW) domain.

References

1. Yager R. A new approach to the summarization of data. *Information Sciences.* 1982; **28**-1. p. 69-86.
2. Zadeh L. A computational approach to fuzzy quantifiers in natural languages. *Comput. and Math. with Applications*. 1983; **9**-1. p. 149-184.
3. Kacprzyk J. Fuzzy logic for linguistic summarization of databases. *IEEE International Fuzzy Systems Conference*. 1999. p. 813-818.
4. Kacprzyk J, Yager R. Linguistic summaries of data using fuzzy logic. *Int. Journal of General Systems*. 2001; **30**-2. p. 133-154.
5. Kacprzyk J, Zadrożny S. Data mining via protoform based linguistic summaries: Some possible relations to natural language generation. *CIDM*. 2009. p. 217-224.
6. Kacprzyk J, Yager R, Zadrożny S. A fuzzy logic based approach to linguistic summaries in databases. *Int. Jour. of Applied Mathematical Computer Science*. 2000; **10**. p. 813-834.
7. Kacprzyk J, Zadrożny S. Summarizing the contents of web server logs: a fuzzy linguistic approach. *FUZZ-IEEE*. 2007. p. 1-6.
8. Castillo-Ortega R, Marín N, Sánchez D. Fuzzy quantification-based linguistic summaries in data cubes with hierarchical fuzzy partition of time dimension. *LNCS*. 2009; **5788**. p. 578-585.
9. Keogh E, Chu S, Hart D, Pazzani M. An online algorithm for segmenting time series. *Proc. 2001 IEEE Int. Conf. on Data Mining*. 2001.
10. Keogh E, Chu S, Hart D, Pazzani M. Segmenting time series: a survey and novel approach. In: *Data Mining in Time Series Databases*. 2004.
11. Alvarez-Alvarez A, Sánchez-Valdes D, Triviño G, Sánchez A, Suárez PD. Automatic linguistic report of traffic evolution in roads. In: *Expert Systems with Applications.* 2012; **39**-12. p. 11293-11302.
12. Kacprzyk J, Zadrożny S. Linguistic summaries of time series: A powerful and prospective tool for discovering knowledge on time varying processes and systems. In: *Towards the Future of Fuzzy Logic.* 2015. p. 65-79.
13. Yager R, Ford K, Cañas A.J. An approach to the linguistic summarization of data. In: *Proc. 3rd Int. Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems* (*LNCS 'Uncertainty in Knowledge Bases'*). 1990; **521**. p. 456-468.
14. Kasprzyk J, Wilbik A, Zadrozny S. Towards human consistent linguistic summarization of time series via computing with words and perceptions. In: *Forging New Frontiers: Fuzzy Pioneers I.* 2007; **217**. p. 17-35.
15. Zadeh L.A. Fuzzy logic as the logic of natural languages. In: Melin P, Castillo O, Ramírez E, Kasprzyk J, Pedrycz W, editors. *Analysis and Design of Intelligent Systems Using Soft Computing Techniques*. 2007. p. 2-3.
16. Kacprzyk J, Zadrożny S. On interactive linguistic summarization of databases via a fuzzy-logic based querying add-on to Microsoft Access. *Lecture Notes in Computer Science (LNCS)*; **1625**. p. 462-472.
17. Kacprzyk J, Zadrożny S. Supporting decision making via verbalization of data analysis results using linguistic data summaries. In: Rakus-Andersson E, Yager R, et al., editors. *Recent Advances in Decision Making (SCI)*. 2009; **222**. p. 121-143.
18. Kacprzyk J, Yager R. Linguistic summaries of data using fuzzy logic. *Int. Journal of General Systems*. 2001; **30**-2. p. 133-154.
19. Kacprzyk J, Zadrożny S. Linguistic summarization of the contents of Web server logs via the ordered weighted averaging (OWA) operators. *Fuzzy Sets and Systems*. 2016; **285**. p. 182-198.
20. Donis-Diaz CA, Bello R, Kacprzyk J. Linguistic data summarization using an enhanced genetic algorithm. *Czasopismo Techniczne. Automatyka (Technical Transactions. Automatic Control)*. 2013; **110**-2-AC. p. 3-12.
21. Marín N., Sánchez D. On generating linguistic descriptions of time series. *Fuzzy Sets and Systems*. 2016; **285**. p. 6-30.
22. Castillo-Ortega R, Marín N, Sanchez D, Tettamanzi AGB. Linguistic summarization of time series data using genetic algorithms. *Proc. 7th Conf. of the European Society for Fuzzy Logic and Technology (EUSFLAT-LFA)*. 2011. pp. 416-423.
23. Batyrshin IZ, Sheremetov LB. Perception-based approach to time series data mining. *Applied Soft Computing*. 2008; 8. p. 1211-1221.
24. Prado R, West M. *Time series: Modeling, computation, and inference* (Texts in Statistical Science), Chapman&Hall/CRC. 2010.
25. Kasprzyk J, Wilbik A, Zadrożny S. Linguistic summarization of time series under different granulation of describing features. In: Kryszkiewicz M., Peters J.F., et al., editors. *Proc. Int. Conf. Rough Sets and Intelligent Systems Paradigms (LNCS)*. 2007; **4585**. p. 230-240.
26. Kasprzyk J, Wilbik A, Zadrożny S. On linguistic summarization of numerical time series using fuzzy logic with linguistic quantifiers. *Studies in Computational Intelligence (SCI)*. 2008; **109**. p. 169-184.
27. Yager R. Approximate reasoning as a basis for computing with words. In: Zadeh LA, Kacprzyk J., editors. *Computing with Words in Information/Intelligent Systems 1: Foundations*. 1999. p. 50-77.
28. Kitchenham B, Pfleeger SL, Fenton N. Towards a framework for software measurement validation. *IEEE Trans. SE*. 1995; **21**-2. p. 929-944.