Software and algorithmic decision support tools for real estate selection and quality assessment

I S Zelenskiy¹, D S Parygin¹,³, D Ather², I N Soplyakov¹, A Yu Antyufeev¹ and E A Prigarin¹

¹ Volgograd State Technical University, 28, Lenina Ave., Volgograd, 400005, Russia
² Teerthanker Mahaveer University, N.H. 24, Delhi Road, Moradabad, 244001, India
³ dparygin@gmail.com

Abstract. This paper describes the development of an algorithm for comprehensive assessment the quality level of real estate objects based on multicriteria estimation. Existing approaches to assessment and two main groups of parameters are noted. The module of pre-processing and structuring of records about real estate objects based on text processing is presented. In addition the task to indicate suspicious real estate ads is considered and an algorithm of the solution based on image processing is presented. Conclusions are made about the problems and prospects of applying the proposed approach to the assessment the quality of real estate objects.

1. Introduction
Searching for a real estate object suitable for living, business or other purposes is an essential and almost inevitable task in human life [1]. To choose the best and most suitable option is to consider and analyze a significant numbers of alternatives from various data sources, comparing them by a set of quality criteria (cost, comfort, infrastructure etc.). Ultimately, such choice is based on the assessment and comparison of the quality of objects among a set of alternatives, i.e. building a quality rating of these alternatives. At the same time, the process of determining the set of quality criteria, assessing objects and constructing a rating becomes more difficult for a person the more objects he/she gets on the list of alternatives. In this regard, the task of developing a software implementable methodology to assess the quality of real estate objects and its subsequent implementation as part of the integrated systems for processing data on urban areas arises [2].

2. Data preparation

2.1. Descriptions of structure formation
The most detailed and relevant information, among the existing open sources of real estate data, is provided by real estate ads which users post on specialized sites [3]. The key problem is the fact that information in such ads is most often presented in the form of textual descriptions of objects in a natural language, i.e. in an unstructured or, at best, in a partially structured form, unsuitable for machine processing without additional pre-processing [4]. As a result, the task of extracting the structured descriptions of real estate objects from the texts of ads arises.

Among the currently available technologies for processing of unstructured text in a natural language, the most popular freely distributed, such as GATE [5], NLTK [6], OpenNLP [7], DeepDive
etc., are designed for English text analysis while the ads which need to be structured are written in Russian.

The suitable solution is the formal grammar description language Tomita and the tool Tomita-parser, developed by Yandex for extracting named entities from Russian texts based on the method of Left-to-right (LR) parsing of natural languages [9].

To extract information from texts Tomita-parser uses keyword dictionaries and context-free (CF) grammars written in Tomita, which describe words and their chains to be searched in the text in a generalized form, and rules for their interpretation (output) [9].

Grammar rules are generally represented as chains of the following form:

$$S \rightarrow S_1S_2 \ldots S_n,$$  \hspace{1cm} (1)

$S$ in (1) is a non-terminal of the grammar and $S_1S_2\ldots S_n$ is the right part of the grammar rule.

For each non-terminal mentioned in the right part of any rule, there must be a rule in which this non-terminal is to the left from $\rightarrow$. The order of listing the rules in the grammar does not matter, i.e. does not affect the process of applying these rules to the text. If there is a chain of words in the input text that corresponds to the right side, the rule is “executed” and the grammar uses this chain as the value of the non-terminal on the left. The example is presented on figure 1.

![Figure 1. Tomita grammar example](image)

In this paper grammars were used to define the rules for searching various chains of words that represent the characteristics of real estate objects described in ads (address, area, rooms number, etc.).

As input, Tomita-parser receives the configuration files (keyword dictionaries and grammars, on the basis of which the text will be analyzed) as well as files with the analyzed texts. As an output, the parser provides an XML-tree with the results of parsing the texts submitted to the input, suitable for further machine processing [4].

A program shell in Python 3 was developed for Tomita-parser to the tasks of choosing a configuration according to the type of real estate in the ad (apartment, office etc.), as well as launching Tomita with the selected configuration and subsequent processing of the result received from the parser (figure 2). The final result is structured descriptions of the real estate objects from the ads (figure 3).

2.2. Ads filtering

It is necessary to take into account that realtor ads exist as well as deliberately false ads, often carrying false information about real estate objects or even describing the objects that do not exist at all. Such an unreliable data source is a very common thing in the sale or rental of residential real estate as, for example, apartments.

Such ads may be identified by filtering on the basis of a set of formal signs. Later this fact can be taken into account when calculating the rating of described objects, for example, by lowering the quality level of objects from such “suspicious” ads.

In the framework of this paper, the following formal signs of a “suspicious” apartment ad were defined:

- The ad contains known agent phone number.
- 5 or more ads contain the same phone number.
- The same phone number is indicated in ads with various sellers.
- 2 or more ads describing different objects contain 1 or more same pictures (photos).

The filtering should be carried out in the database every N hours. An algorithm is presented below (figure 4).
2.3. Image hashing

Insofar as the image comparing should be performed many times with many images during the ads filtering process, the solution of this task must be minimally time consuming [9]. Therefore the perceptual hash algorithms are the best option for the image comparing task in the framework of this paper.

This approach consists of construction of an individual (but not unique) “fingerprint” (Fig. 5), which describes the original image in the form of a chain of bits. Such a chain is called a hash. Comparison of images is then reduced to comparing their hashes. The comparison operation is performed by counting the Hamming distances, i.e. the numbers of non-similar bits, between image hashes [10].

Among the simplest hash algorithms are aHash, pHash and dHash [10].

aHash (Simple (Average) Hash) [10] reduces to displaying the average of low frequencies. Therefore, to obtain such a hash function, it is necessary to filter them out. This procedure is performed in the following steps:
1. The original image is reduced to 32x32 pixels (or even up to 8x8).
2. Image is transferred to the grayscale palette.
3. The average colour value of all pixels is calculated.
4. The resultant hash value is constructed by passing through all pixels of the image.
5. For pixels with a value greater than the average, the value of one is assumed, and for all others, zero.

This algorithm has a number of advantages, such as speed and stability of the hash to the operations of scaling, compression or stretching, changing brightness, contrast, and manipulating colours [11].

pHash (Discrete Cosine Transform Based Hash) [12] is based on a discrete cosine transform (DCT) that expresses a function or signal as a sum of cosine functions with different frequencies and amplitudes. The principle of operation and procedure for implementing this algorithm are similar to aHash in terms of performing discoloration, noise filtering and compression operations to a size of 32x32 pixels. Additionally:
1. The image undergoes a discrete cosine transformation.
2. For the subsequent construction of the hash, low-frequency coefficients are used which are in the left upper block 8x8 of the obtained matrix.

dHash [11] is similar to pHash and aHash in terms of reducing image size and discoloration. However, gradients are used instead of the average colour value. The direction of the gradient is determined by the difference of neighbouring pixels. Hash bits in this case are set depending on whether the pixel on the left is brighter than the pixel on the right.

To perform this comparison and obtain an 8-byte hash, it is necessary to reduce the image size to 9x8 pixels, rather than 8x8, as in other algorithms. The order of writing bits to the hash does not play a special role.

Among these algorithms dHash gives quite a few false positives, while having a significant advantage in speed over pHash and almost equal speed with aHash [13, 14] and so it was chosen to be implemented for image comparing.

For practical implementation of dHash algorithm the ImageHash library [15] written in Python was used. Image comparing process was implemented using Hamming distance equal to zero in order to filter only true duplicates.

To speed up the further processing it was decided to calculate every hash once and store it in the database with the image itself.

3. Real estate’s quality calculation

3.1. Calculating the quality level

To date, a large number of methods for assessing the quality of real estate objects, for example, unified multi-unit residential buildings classification [16] or office space classification [17], have been developed.

In this case, the most obvious indicator of the real estate object’s quality is its market price, because the better the properties of the object (whether it is an office, a house or an apartment), the higher the price.

As the main characteristics of the real estate object, affecting its final quality level and, accordingly, the final price, the following are usually defined:

- Properties of an object itself and the building in which it is located (number of rooms, total area, availability and number of elevators, etc.).
- Characteristics of the surrounding infrastructure (the availability of shops, pharmacies, public transport stops within walking distance, etc.) [18].

The quality level of a real estate object is calculated from assessments of many quality criteria. A modified direct multicriteria estimation by the weighted sum method, consisting of the following steps (figure 5), is used to calculate the rating of a sample of real estate objects of a given type.

The highest possible assessment is calculated using (2):

$$ O_{\text{max}} = \sum_{i=1}^{N} k_i \cdot O_{\text{imax}} $$

where $O_{\text{max}}$ is the highest possible assessment, $k_i$ is the weight of the i-th criterion ($k_i = [0, 1], k_i \in \mathbb{R}$), $O_{\text{imax}}$ is the highest possible integer grade on the i-th criterion’s scale, $N$ is the total number of criteria.

The total assessment of an object is calculated using (3).

$$ O_{\text{sum}} = \sum_{i=1}^{N} k_i \cdot O_i $$

where $O_{\text{sum}}$ is the total assessment of an object, $k_i$ is the weight of the i-th criterion ($k_i = [0, 1], k_i \in \mathbb{R}$), $O_i$ – is an integer grade for the i-th criterion, $N$ is the total number of criteria.

To normalize the total assessment of an object (4) is used.

$$ R_1 = \frac{O_{\text{sum}}}{O_{\text{max}}} \cdot 100 \% $$

where $R_1$ is normalized intermediate rating value of an object, $O_{\text{sum}}$ is the total assessment of an object, $O_{\text{max}}$ is the highest possible assessment.
In order to take into account the parameters the influence of which on the final quality level is objectively known and does not depend on any preferences, adjustments mechanism is introduced [19]. This mechanism is similar to the mechanism of the same name, which is used in the assessment of the real estate object’s market price [18].

The principle of both is to use correction coefficients, real numbers which indicates how much less the quality level of an object with some parameter is lower than its full counterpart without such.

For example, quality level of an apartment located on the first floor of the residential building will be equal to 0.93 of the quality level of its full counterpart located on the sixth floor yet the quality level of a full counterpart located on the fourth floor will be equal to 1.00 of it. It is also possible that an adjustment will raise the quality level, not lower it. Then the corresponding correction coefficient will have value more than 1.00.

Therefore, the final rating value of an object is calculated according to (5).

\[
R = R_1 \cdot \prod_{i=1}^{M} kp_i,
\]

where \( R \) is the final rating value of an object, \( R_1 \) is normalized intermediate rating value, \( kp_i \) is i-th correction coefficient (\( kp_i \in R \)), \( M \) is the number of adjustments to apply.

Then it is multiplied by 100 to present the result in the form of percent. In cases where adjustments raise the final value of the rating above 100%, this value is assumed to be 100%.

3.2. Calculating the weights of the criteria
First it is necessary to determine the weight of each criterion by which this object will be evaluated to calculate quality level of a real estate object. The Saati’s pairwise comparison model [20] is used to solve this task.
This method is based on a pairwise comparison of alternatives (in this paper alternatives is the criteria for assessing the quality of real estate objects). As a result, for each pair of alternatives the expert indicates which of them has a higher level of significance (importance).

Saatí’s pairwise comparison algorithm consists of the following steps:

1. The matrix A, a M×M real matrix, where m is the number of alternatives (criteria) considered, is defined.
2. Entries $a_{ii}$ on the main diagonal of the matrix A are filled with the value 1.
3. Each other entry $a_{ij}$ of the matrix A is filled according to Table 1 and (6) to represent the importance of the i-th criterion relative to the j-th criterion:
   \[ a_{ij} = \frac{1}{a_{ji}} \]  
   (6)

4. The final weight $k_i$ of each criterion is calculated by:
   \[ k_i = \frac{a_i}{\sum_{j=1}^{M} a_j} \]
   (7)

where values $a_i$ and $a_j$ are calculated according to (8):
   \[ a_i = \sqrt[\prod_{j=1}^{M} a_{ij}] \]
   (8)

To take the user’s preferences into account it is possible to pick the criteria which he/she considers as more important. Therefore the values $a_{ij}$ in the matrix A for these criteria is raised by 2 steps, i.e. 1 becomes 5 or 1/9 becomes 1/5 (see Table 1). The values $a_{ji}$ are then recalculated according to (6) and the weights $k_i$ are then recalculated as well.

### Table 1. Table of relative scores.

| $a_{ij}$ | Interpretation                      |
|----------|-------------------------------------|
| 1/9      | j is absolutely more important than i |
| 1/7      | j is strongly more important than i  |
| 1/5      | i is less important than j           |
| 1/3      | i is slightly less important than j  |
| 1        | i and j are equally important       |
| 3        | i is slightly more important than j  |
| 5        | i is more important than j           |
| 7        | i is strongly more important than j  |
| 9        | i is absolutely more important than j|

### 3.3. Program implementation and testing

A prototype implementation in Python 3 was created to test the methodology developed in this paper.

It was decided to set parameters like pairwise comparison matrices A, rating scales for each criterion, the lists of the criteria etc. through configuration files in order to make it flexible. This approach allows the prototype being easily changed and adapted almost on the run without changing the source code itself.

For a test run a sample of 10 randomly selected apartments for sale on sites avito.ru and irr.ru was used. The list of the criteria evaluated presented in Table 2 was defined.

### Table 2. List of criteria.

| Criterion       | Possible values                                                                 |
|-----------------|---------------------------------------------------------------------------------|
| Living area     | Numerical values in square meters. State standards [21] are taken into account when assessing |
| Rooms number    | Integer numbers                                                                 |
| Repair          | European-quality repair, redecoration, designer repair, etc.                    |
| Parking         | Is there or not                                                                  |
| Bathroom        | Separate, combined, absent                                                      |
| Balcony         | Is there or not                                                                  |
| Elevator        | Is there or not                                                                  |
| Conditioner     | Is there or not                                                                  |
| Building’s age  | The number of years between the current year and the building’s construction year |
The objects of a given type were considered at a distance of 1000 m (pharmacies, stops) and 2000 m (stores, schools, kindergartens, clinics) from the object of assessment when calculating infrastructure criteria. The floor of an apartment’s location and building’s walls material were taken into account using adjustments mechanism (see above).

All the apartments in the sample were assessed taking into account several sets user preferences, each representing one of the user groups (see table 3). Weights for the criteria defined as more important for their group were recalculated as described above. Part of the test run results is presented in table 4.

### Table 3. User groups.

| User group                              | Important criteria                                      |
|-----------------------------------------|---------------------------------------------------------|
| Group 1 – Student                       | Stops, stores                                           |
| Group 2 – Young family with a school child | Stores, schools, pharmacies, clinics, living area, rooms number, repair |
| Group 3 – Retiree                        | Pharmacies, stops, clinics, stores, elevator            |
| Group 4 – Young family without children and with a car | Living area, repair, parking |

### Table 4. Test results.

| Criterion                  | Apart. A | Apart. B | Apart. C | Apart. D |
|----------------------------|----------|----------|----------|----------|
| Living area                | 79.6     | 66       | 68       | 29.9     |
| Rooms number               | 3        | 2        | 3        | 1        |
| Repair                     |          | Redecoration | European-quality | —        |
| Floor                      | 2 of 9   | 1 of 9   | 6 of 12  | 2 of 10  |
| Parking                    |          | +        | —        |          |
| Bathroom                   | Separated | Separated | Separated | Combined |
| Balcony                    | +        | +        | +        | –        |
| Elevator                   |          | +        | +        | +        |
| Conditioner                | +        | —        | +        | –        |
| Construction year          | 1963     | 2009     | 1983     | 2010     |
| Walls material             | Panels   | Brick    | Panels   | Monolithic |
| Pharmacies                 | +        | +        | +        | +        |
| Stores                     | +        | +        | +        | +        |
| Clinics                    |          | +        | +        | +        |
| Stops                      | +        | +        | +        | +        |
| Schools                    |          | +        | +        | +        |
| Kindergartens              | –        | +        | –        | +        |
| Parks                      | –        | +        | –        | –        |

| User group | Rating value |
|------------|--------------|
| Group 1    | 40           |
| Group 2    | 37           |
| Group 3    | 38           |
| Group 4    | 42           |

### 4. Conclusion

The developed algorithms and software for analyzing data and assessing the quality of real estate objects allow the preparation of data and subsequent calculation of the rating of real estate objects in terms of quality and comfort, taking into account user preferences. The proposed solutions contribute to the determination of the aggregate quality indicator of a real estate object and form the basis for creating tools that will increase the objectivity of real estate valuation and the validity of decisions on various operations with objects.

Future research involves the development of a complex system of analytical processing of advertisement records to calculate the rating of the entire range of real estate available on the market. In addition, it is promising to solve the inverse problem associated with identifying factors that affect the assessment of objects for certain groups of stakeholders. The totality of the proposed solutions will allow a comprehensive assessment of the investment attractiveness of real estate for various purposes of operation.

### Acknowledgments

The reported study was funded by Russian Foundation for Basic Research according to the research project No. 18-37-20066_mol_a_ved. The authors express gratitude to colleagues from UCLab involved in the development of OS.UrbanBasis.com project.
References

[1] Shabalina O A, Davtian A G, Sadovnikova N P and Parygin D S 2019 Modeling dynamic goal-setting in socio-economic systems [Моделирование динамического целеполагания в социально-экономических системах] (Volgograd: IUNL VolgGTU)

[2] Parygin D, Kozlov D, Sadovnikova N, Kvetkin V, Sopyakov I and Malikov V 2019 Development the Online Operating System of Urban Infrastructure Data Communications in Computer and Information Science 1084 pp 203–216

[3] Donchenko D, Sadovnikova N, Parygin D and Shabalina O 2016 Promoting urban projects through social networks using analysis of users influence in social graph Advances in Computer Science Research 2017 pp 162–165

[4] Cherkesov V, Malikov V, Golubev A, Parygin D and Smykovskaya T 2017 Parsing of Data on Real Estate Objects from Network Resource Advances in Computer Science Research 72 pp 385–388

[5] Cunningham H, Maynard D, Bontcheva K and Tablan V 2002 GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications Proc. of the 40th Anniversary Meeting of the Association for Computational Linguistics pp 145–158

[6] Bird S, Klein E and Loper E 2009 Natural Language Processing with Python (Sebastopol: O'Reilly Media Inc)

[7] Apache OpenNLP library (URL: https://opennlp.apache.org/)

[8] DeepDive (URL: http://deepdive.stanford.edu/)

[9] Golubev A, Zelenskiy I, Parygin D, Cherkesov V, Finogeev A and Degtyarenko D 2018 Validation of Real Estate Ads based on the Identification of Identical Images Proc. of the 7th International Conference on System Modeling & Advancement in Research Trends pp 274–279

[10] Perceptual image hashes (URL: https://jenssegers.com/61/perceptual-image-hashes)

[11] Zauner C 2010 Implementation and Benchmarking of Perceptual Image Hash Functions

[12] Discrete Cosine Transform Locality-Sensitive Hashes for Face Retrieval (URL: https://www.researchgate.net/publication/261672303_Discrete_Cosine_Transform_Locality-Sensitive_Hashes_for_Face_Retrieval)

[13] Looks Like It (URL: https://www.hackerfactor.com/blog/index.php/?archives/432-Looks-Like-It.html)

[14] Kind of Like That (URL: https://www.hackerfactor.com/blog/index.php/?archives/529-Kind-of-Like-That.html)

[15] ImageHash 4.0 (URL: https://pypi.org/project/ImageHash/)

[16] Unified multi-unit residential buildings classification (URL: http://www.domchegov.ru/userfiles/docs/Edinaya_klassifikaciya1.pdf)

[17] Sternik G 2003 Office space classification (URL: http://realtymarket.org/docs/lib_58.htm)

[18] Leifer L and Krainikova T 2016 Real estate appraiser’s guide (Nizhny Novgorod: INFORM-Value)

[19] Parygin D S, Malikov V P, Golubev A V, Sadovnikova N P, Petrova T M and Finogeev A G 2018 Categorical data processing for real estate objects valuation using statistical analysis,” J. of Physics: Conference Series 1015 032102

[20] Saaty T L 2008 Relative Measurement and its Generalization in Decision Making: Why Pairwise Comparisons are Central in Mathematics for the Measurement of Intangible Factors – The Analytic Hierarchy/Network Process Review of the Royal Spanish Academy of Sciences (Series A, Mathematics) 102 no 2 pp 251–318

[21] Federal Law “On Social Guarantees for Employees of Some Federal Executive Bodies and Amending Certain Legislative Acts of the Russian Federation” 30.12.2012 no 283-FL sec 7