Framework for Evolutionary Modelling in Text Mining

Michael. Y. Bogatyrev
Tula State University
Tula, Russia
okkambo@mail.ru

Alexey P. Terekhov
Tula State University,
Tula, Russia
temp_@mail.ru

Abstract

Special framework for Evolutionary Modelling is presented. It is oriented on experimental investigations of schemes and properties of genetic algorithms. With the help of this framework Evolutionary Approach to Conceptual Graphs Clustering is investigated. Some experimental results of clustering scientific papers abstracts are presented.

1 Introduction

The problem of extracting natural language semantics is very complex and could not be solved by implementing a single approach. In the field of Text Mining solution of this problem is proposed by constructing semantic models of a text and applying these models to detect special patterns such as *clusters*, *associations*, *deviations* and *similarities* in text collections. The patterns mentioned, being abstract mathematical objects or numerical values, do not solve initial problem of extracting language semantics and often need to be *semantically interpreted*.

In this paper *Evolutionary Modelling* [2] as a possible way of such semantic interpretation is presented. It is based on the paradigm of Evolutionary Computation [24] which in turn engenders the area of Genetic Algorithms [14], a powerful and domain independent search and optimization technique. As domain independent tool genetic algorithms have been used to solve various Text Mining problems: information and documents retrieval [1], [12], text segmentation [15], extracting key phrases from text [23], etc. Having a semantic text model and a Text Mining problem, Evolutionary Modelling is the way to find the best solution of a problem by *evolving* parameters of a model. The mechanism of evolution of a model is specified by appropriate genetic algorithm. Genetic algorithm produces populations of various solutions. Analyzing the evolution of such populations one can construct possible semantic interpretations of them.

The method which is proposed here is mainly experimental and is supported by special framework for evolutionary modelling. This framework is a part of EVOLIB digital library research project [3]. Among user functions of EVOLIB there is the function of fact extraction from abstracts of scientific papers [3]. To realize this function we use Conceptual Graphs as semantic models of abstract sentences. Then Evolutionary Modelling is applied to conceptual graphs clustering. Various similarity measures are investigated with the help of the framework. As result abstracts of *declarative* and *narrative* types were detected in the library.

The paper is organized as follows. Section 2 contains quite general description of the principle of Evolutionary Computation to show its fundamental nature. Some peculiarities of application of Evolutionary computation in Text Mining are also presented. In section 3

evolutionary approach is applied to conceptual graphs clustering problem. Section 4 is devoted to the Framework for Evolutionary Modelling as a part of EVOLIB digital library research project. Illustrative examples of conceptual graphs clustering are shown in this section. Finally, conclusion and further work plan constitute the section 5.

2 Evolutionary Computations in Text Mining

Consider the principle of Evolutionary Computation in general to outline some its features feasible to possible implementations in Text Mining.

2.1. Principle of Evolutionary Computation

Let $\boldsymbol{X}$ is a set of solutions of a Text Mining problem. Every solution $x \in \boldsymbol{X}$ can be characterized by a quality measure named as *fitness function* or *fitness*. Actually very often this measure is not a function in mathematical sense because x may be not a numerical value, being for example configuration of clusters. So in general instead of denoting $y = f(x)$ we have to use a *mapping* $\boldsymbol{f}: \boldsymbol{X} \rightarrow \boldsymbol{Y}$ where $\boldsymbol{Y} \subseteq \boldsymbol{R}^{+}$ is the subset of a set of positive real numbers. Nevertheless, following the tradition we will use "$f(\,)$" notation and call it as fitness function.

Let solutions of a Text Mining problem depend on a set of parameters $\boldsymbol{P}$ of a model which is used in a problem. Most of Text Mining problems which have been solved by using genetic algorithms can be formulated as the following optimization problem: it is required to find optimal values of parameters p^{*} which deliver maximum fitness value $y^{*} \in \boldsymbol{Y}$, so the following is true:

$$p^{*} = \underset{p^{*} \in \boldsymbol{P}}{\operatorname{argmax}} \boldsymbol{f}(x), \qquad (1)$$

Evolutionary approach to solving this problem consists in the following.

2.1. Building encoding scheme. *Encoding scheme* is the mapping $\boldsymbol{\varphi}: \boldsymbol{P} \rightarrow \boldsymbol{S}$ where set $\boldsymbol{S}$ contains objects which *encode* parameters from $\boldsymbol{P}$. Most of genetic algorithms use binary encoding and every value of $p \in \boldsymbol{P}$ is represented as binary string. Then these strings are randomly manipulated by genetic algorithm and all variety of existing manipulation methods (genetic operators) [11] can be treated as permutations of strings. Thereupon one can conclude that genetic algorithms only constitute a sort of random search methods. Actually encoding is very important and represents the essence of evolutionary approach. There is *an atomic* principle of encoding which claims that encoding scheme has to be such that it generates minimal elements which influence on the values of elements of $\boldsymbol{Y}$. As in biology, heredity theory claims that gene (strictly gene combinations) is the minimal element which really determines individual characteristics, as here, in Evolutionary computation, atomic encoding principle plays the same role. Therefore genetic algorithms were justly named as *genetic*. Encoding scheme is not necessarily binary (as it is not binary in Nature): every string position contains a symbol (*gene*) from *encoding alphabet*, and there are variants of alphabets applied in encoding schemata [1], [4], [7], [23]. But necessarily there exists an inverse mapping $\boldsymbol{\varphi}^{-1}: \boldsymbol{S} \rightarrow \boldsymbol{P}$, so for every $s \in \boldsymbol{S}$ there exists $p \in \boldsymbol{P}$.

2.2. Evolutionary algorithm. For given encoding scheme the following algorithm solves the problem (1).

A. Randomly generate an initial set (population) $\boldsymbol{S}_0$ of objects from $\boldsymbol{S}$.
B. Start *evolution* of the populations by applying a set of operators $\boldsymbol{A}$ to population $\boldsymbol{S}_0$ and further iteratively so that for every $\boldsymbol{S}_{k+1} = \boldsymbol{A}(\boldsymbol{S}_k)$ exists at least one

$$\boldsymbol{f}[\boldsymbol{\varphi}^{-1}(s_{k+1})] \geq \boldsymbol{f}[\boldsymbol{\varphi}^{-1}(s_k)], \qquad (2)$$

where $s_k \in \boldsymbol{S}_k$ and $s_{k+1} \in \boldsymbol{S}_{k+1}$.

C. Stop evolution of populations when p^* is found in a population.

If the set of operators $\boldsymbol{A}$ consists of genetic operators of *selection*, *mutation* and *recombination* (crossover) then evolutionary algorithm is named as *genetic algorithm*.

Selection works so that condition (2) is supported by the following "biological" principle: good parents produce good offspring (that is not true in Nature). So the higher fitness chromosomes have more opportunity to be selected than the lower ones and good solution is always alive in the next generation.

Crossover is the genetic operator that mixes two chromosomes together to form a new offspring. It does mixing by replacing fragments of chromosome's code divided in certain one or several randomly selected points.

Mutation involves modification of the gene values by randomly selecting new value from the alphabet at random point in the strings of genes.

Being realized, the algorithm (A. – C.) provides fast and quite exact solution of the problem (1).

Quite exact means that genetic algorithm stops in a neighbourhood of global extreme of fitness function $\boldsymbol{f}$ [11]. The size of a neighbourhood around extreme depends on the fitness function and parameters of genetic operators. When genetic algorithm works too fast it may stop at local extreme. This feature is traditionally considered as the lack of the algorithm but it may be useful for Text Mining since local extreme of quality measure may be semantically "better" than global extreme. In our experiments we have observed just that situation [4].

Operating speed of genetic algorithms could not be high because they have to manage not one but a whole set of possible solutions and evaluate fitness function N times on every step of evolution, where N is the size of population. Nevertheless, they are fast as compared to other algorithms for solving the problem (1) due to the following properties of genetic algorithms:

- their feature of *implicit parallelism* [11] provides a quasi parallel way of computations, and by using properly constructed genetic operators genetic algorithms have an ability to span simultaneously a large subsets of search space;
- a population of chromosomes evolves successfully to an extreme because specific "good" gene combinations known as *building blocks* [11] appear and breed other, "better" gene combinations which form final solution.

The last property of appearing building blocks during evolution is the reason for applying strings as working objects in evolutionary algorithms because building blocks act on strings. It is also needed to control building blocks in evolutionary algorithms and this is realized in our framework.

2.2. Applications in Text Mining

Genetic algorithms have been applied to solve many problems in Information Retrieval and Text Mining. Following the review in [1] we briefly outline distinctive directions of their implementations and comment them to illustrate properties of evolutionary algorithm described in previous subsection.

Documents Indexing and Retrieval [1], [8]. The main problem in this area is to adapt documents descriptions in the library with the requirements of queries. That adaptation allows constructing special models (indexes) of stored documents which facilitate search for documents being relevant to the query. Encoding scheme here represents documents indexes as chromosomes with binary and non binary alphabets. Using those indexes it is easy to construct fitness function which is based on calculating the similarity between the current document and each of the queries. According to our model of evolutionary algorithm from

previous subsection here $\boldsymbol{X}$ is a set of documents, encoding scheme is the mapping $\boldsymbol{\psi: X \rightarrow S}$ and fitness function is calculated as $f(s, q)$, where s is a chromosome and q is a query. For some practical reasons mutations are not allowed on chromosomes. So the evolutionary algorithms in this area are realized as not classical genetic algorithms.

Learning of Matching Functions and Queries. [9], [20]. Another model of stored documents which is used in Text Mining applications is vector space model. The content of library documents is described by n terms which constitute n-dimensional vector space. Each document is a point in this space which coordinates are weights of corresponding terms occurred in the document. A query is also treated in the same way as a vector constructed from the terms and weights distinguished from user request. Document retrieval is based on the measurement of the similarity between the query and the documents. Here the learning also means adapting objects - matching functions and queries – to provide for certain fitness function to have an extreme. Evolutionary algorithms have been applied to different matching functions - from traditional form of linear combination of existing similarity functions to tree form. In the last case *Genetic Programming*, the branch of Evolutionary computation where evolution is performed by operations on graphs, is applied. In Genetic Programming encoding scheme is graph and application of genetic operators on that scheme has some peculiarities. Here mutations are not allowed or restricted by specificity of applied alphabet, crossover on graphs is realized as exchange between two graphs by their sub graphs and selection needs the measure of quality (semantics) of graphs.

Clustering of Documents and Terms [7], [12], [19], [21]. In spite of the fact that clustering problem is well investigated and many clustering algorithms have been proposed, its new solutions are still appearing, also with using evolutionary approach. There are two crucial parameters of clustering problem: a measure of similarity of clustering objects and number of clusters – is it given or not before clustering. In Text Mining, clusters of documents have to be additionally interpreted, including semantic interpretation. Evolutionary algorithms have advantage over traditional clustering methods when:

- measure of similarity of clustering objects is not traditional (Euclidian norm),
- number of clusters is not given and
- number of clusters is great.

All these conditions present in Text Mining problems. For example semantic measure of texts similarity is not traditional, number of clusters is not known in many Text Mining problems and number of clusters is great when they for long texts.

Irrespectively of the nature of Text Mining problem in all three marked above applications areas the mapping $\boldsymbol{f: X \rightarrow Y}$ could not be represented as simple analytical function. It is often represented as multimodal function which has finite breaks [4]. Evolutionary and genetic algorithms are good just for optimization problems with multimodal fitness functions which have finite breaks [11]. That fact is the main motivation for application of evolutionary approach in Text Mining.

3. Evolutionary Approach to Conceptual Graphs Clustering

Consider Conceptual Graphs Clustering problem as an example of application Evolutionary Approach in Text Mining. The formulation of clustering problem actual for Text Mining is known for a long time [16]: for a given set of objects and their associate descriptions it is needed to find clustering that groups these objects into concepts, then find an intentional definition for each concept, and a hierarchical organization for these concepts. That intentional definition may be realized as semantic interpretation of clusters. For Conceptual Graphs [22], the clustering problem urgency depends on how measure of similarity and number of clusters are specified in the problem.

In all known approaches to conceptual clustering [5], [17], [19] classical clustering algorithms of *k* – means and hierarchical clustering have been applied.

3.1. Measures of similarity of conceptual graphs

The Dice coefficients known as standard similarity measure for text documents serve as a base for creating specific measures in concrete problems. This is done in [18] for comparison of conceptual graphs and in [19] for conceptual graphs clustering problem. We also used Dice coefficients and their modifications.

For the two given graphs G_1 and G_2 the similarity measure depends on two values – conceptual similarity s_c and relative similarity s_r.

$$s_c = \frac{2n(G_c)}{n(G_1)+n(G_2)} \tag{3}$$

where $G_c = G_1 \cap G_2$, $n(G)$ is the number of concepts – conceptual nodes of graph G.

$$s_r = \frac{2m(G_c)}{m_{G_c}(G_1)+m_{G_c}(G_2)} \tag{4}$$

where $m(G_c)$ is the number of relations – relative nodes of conceptual graph G_c, $m_{G_c}(G)$ is the number of relations – relative nodes of conceptual graph G, at least one of which belongs to graph G_c.

The measures (3), (4) are imperfect. Using these similarities we can get graphs that have many common concepts and relations but different meanings. This property of standard similarity measures is well known. We get incorrect similarity value due to the presence of identical words in sentences. These words don't carry useful information but affect the calculated similarity value. These words (called clichés, stock phrases and set expressions) exist in all languages. It is necessary to use some kind of filtering to remove this "noise".

If some lexical restrictions are introduced– for example only texts of scientific articles are analyzed – it has become possible to create a set of concepts of general usage in the given research area. For example they are "article", "this", "document", "observe", "describe", etc. These concepts can be safely excluded when analyzing similarity between conceptual graphs.

Taking into account in general usage concepts the value $n(G)$ in (3) is assumed to be the number of concepts that are not in general usage:

$$n(G) = a_1 + a_2 + ... + a_i \ , \ i = 1,...,N \ , \ a_i \in \{0\,|1\},$$

where *N* is the number of all concepts in graph G, a_i (general validity factor) possesses the value 0 or 1 subject to whether *i*-th concept is in general usage or not.

When calculating similarity between conceptual graphs it is also necessary to take into account sizes of the graphs compared. It is also possible to modify the formula for similarity (3), (4) in order to take into account the sizes of the graphs:

$$s_c = \frac{2n(G_c)l}{n(G_1)+n(G_2)}, \tag{5}$$

where

$$l = \begin{cases} k\dfrac{n(G_1)}{n(G_2)}, if\ n(G_1) \geq n(G_2) \\ k\dfrac{n(G_2)}{n(G_1)}, if\ n(G_1) < n(G_2) \end{cases},$$

and k is a scaling factor.

In order to calculate relative similarity between conceptual graphs using (4), one has to take into account the number of relations. Taking the relations into account, the value of $m_{G_c}(G)$ in formula (4) is calculated as follows:

$$m_{G_c}(G) = m_{both} + b_1 + b_2 + ... + b_i,\ i = 1,...,m - m_{both},\ b_i \in \{0,1\},$$

where m is the number of all the relations of graph G, m_{both} is the number of relations of graph G both nodes of which belong to graph G_c, b_i (relevance factor) possesses values in the range of 0 and 1 subject to relation type.

Both measures – conceptual and relative similarity – were applied in clustering experiments separately.

Semantic measure. The measure proposed in [25] is appropriate and well grounded model of semantic measure of similarity between conceptual graphs. The similarity between two concepts is obtained by the distance between them. The distance between two concepts is calculated by their respective positions in the concept hierarchy. This hierarchy can be obtained from WordNet system. In the framework for evolutionary modelling we use WordNet as the source of ontology segments needed to find a closest common parent for two concepts.

3.2. Encoding scheme

Every solution of clustering problem is represented as a string of length n called a chromosome. There exist several ways to represent clustering problem solution in the encoding scheme [7]. Figure 1 illustrates variants of encoding schemes. All of them use long chromosomes which length is equal to the number of objects to be clustered.

The drawback of encodings (a) – (e) on the Figure 1 is that it is necessary to specify a number of clusters which is usually unknown a priori.

We propose modified encoding in which a_i denotes ordinal number of an object that belongs to the same cluster as i-th object. As a result, the number of clusters depends on the distribution of such links between objects. If object A points to object B and the latter points to object C, it means that they are all in the same cluster. This encoding doesn't bind the objects to some particular cluster – it just contains information about links between the objects. Due to that kind of "chain" relations between objects it is enough to add a link between any object from one set and any object from another set in order to join these sets.

This encoding is atomic for clustering problem.

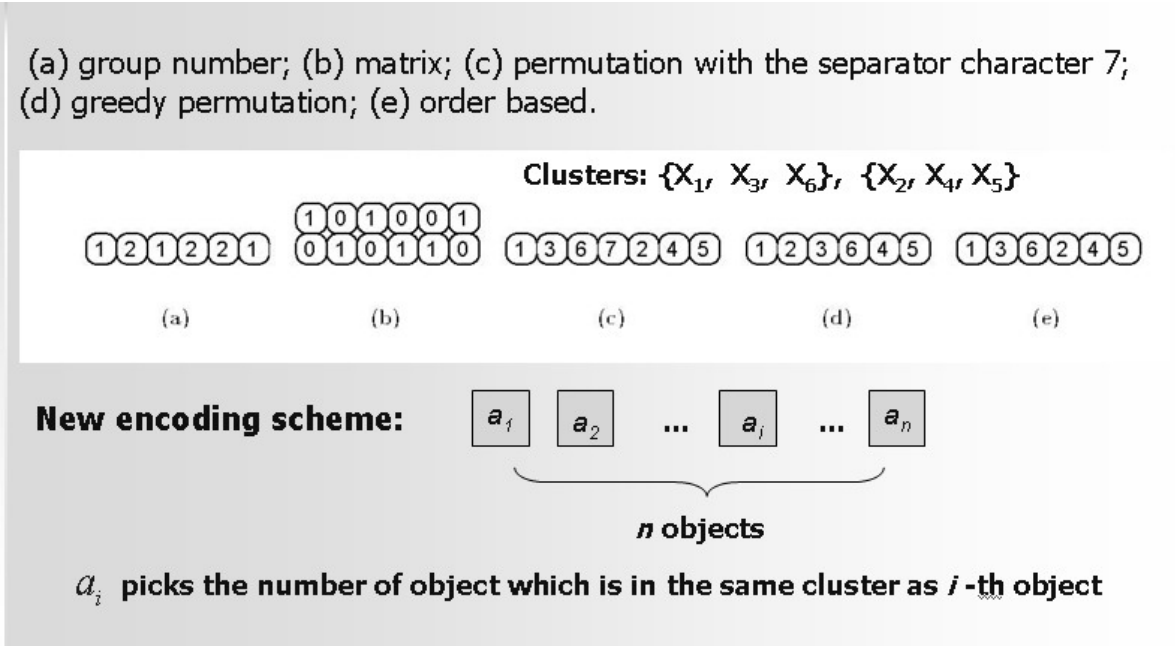


Figure 1: Variants of encoding schemes for conceptual graphs clustering.

Another important advantage of the proposed encoding is that it provides quite fast work of the algorithm even on long chromosomes due to quasi parallelization of calculations: several genes in the chromosome may point to the same cluster simultaneously, so clusters are formed in a quasi parallel way.

4. The EVOLIB Framework

The EVOLIB is a digital library research project [3] that includes a subsystem of Evolutionary computation as a solver. The structure of EVOLIB system is shown in Figure 2.

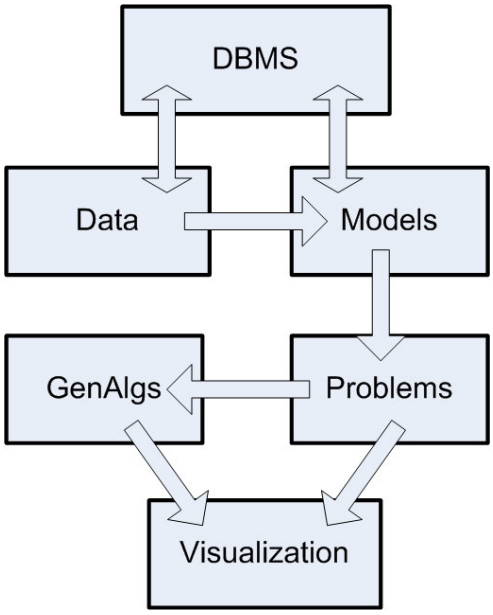


Figure 2: The general structure of EVOLIB system.

The EVOLIB contains the library of scientific papers as a content of Data subsystem. There are over 2500 items in the library. Every paper has an abstract which is an object of processing by the system instead of paper's text. This is based on assumption that an abstract is short and clear essence of a paper. Abstracts form input data for Models subsystem where conceptual graphs have been created.

The problem of building conceptual graphs is solved by using existing approaches [6], [10], [13]: we apply Semantic Roles Labelling as the main instrument for building relations and also WordNet system for English texts. Since there are many Russian papers in the library some special decisions were made concerning the Russian language. The XML database for storing conceptual graphs and corresponding DBMS are parts of the system.

The Problems subsystem realizes the software for conceptual graphs clustering.

The GenAlgs subsystem is devoted to exploration of genetic algorithms by varying their schemes and parameters of genetic operators. The visualization subsystem plays an important role in the experiments.

The EVOLIB system is created as an open framework. That means that new models, new problems and new genetic algorithms can be added to the system without changing its kernel architecture.

4.1. Evolutionary Modelling in Conceptual Graphs Clustering

Evolutionary modelling is appropriate instrument for *learning abstracts semantics*. This is general problem which may be separated on some specific tasks including conceptual graphs clustering.

As it is mentioned above Evolutionary approach is preferable tool for clustering when the number of clustering objects is great and similarity measure is not usual. The first condition leads to possibly great number of clusters which have to be interpreted for further implementations. The second condition, i.e. measures based on Dice coefficients and semantic measure cause the fitness function to be multimodal. There are several variants of clustering for multimodal fitness function [7] because genetic algorithm may quickly find local extreme and then stop. Modelling framework allows adjusting parameters of genetic operators – *mutation probability*, *number of crossover points* and *type of selection* - to control genetic algorithm convergence.

The next important tool of modeling in EVOLIB is visualization. It is applied to show two types of structures of clusters: clustering dendrograms and maps of clusters and their parameters. The last tool of visualization is actual when semantic measure is applied. Visualization also helps to find out what kind of extreme is achieved when genetic algorithm stops. This is achieved by finding and showing building block elements in chromosomes.

Consider some illustrative examples of applying evolutionary modelling in conceptual graphs clustering.

Clustering helps to investigate structures of papers abstracts. It is observed that there are two types of abstracts – abstracts of *declarative* and *narrative* types. As a rule, declarative abstracts are short and contain weakly connected sentences. Oppositely, narrative abstracts are quite long and contain sequences of closely connected sentences. Declarative abstracts are common and may contain new facts as new terms and definitions. Narrative abstracts are not usual and may contain *new ideas* narrated in the abstract. To detect narrative abstracts we tested all three measures of similarity of conceptual graphs. Relative similarity was found as the most appropriate for such detection.

Figure 3 illustrates the most meaningful such result. This is result of clustering two abstracts. One of them is long containing 11 sentences (numbered from 0 to 10). Another abstract consists of a single sentence number 11. In Figure 3 one can see "narrativeness" of the first abstract that is expressed as "nesting of senses" on the clusters structure. At the same time graph number 11 constitutes its own cluster.

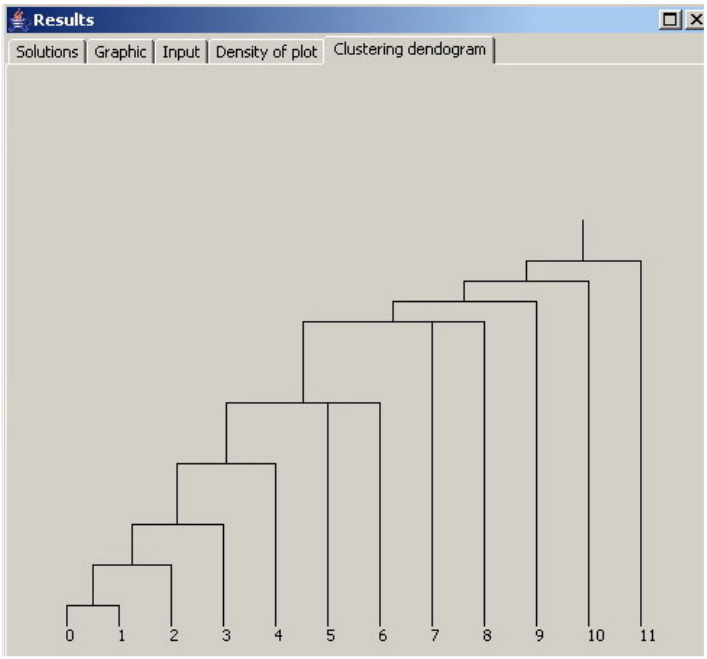


Figure 3: Example of clustering with the help of the relative similarity measure.

Figure 4 illustrates clustering the same two abstracts when semantic measure is applied. It is shown the list of hyperonims (common parents) for concepts in the cluster.

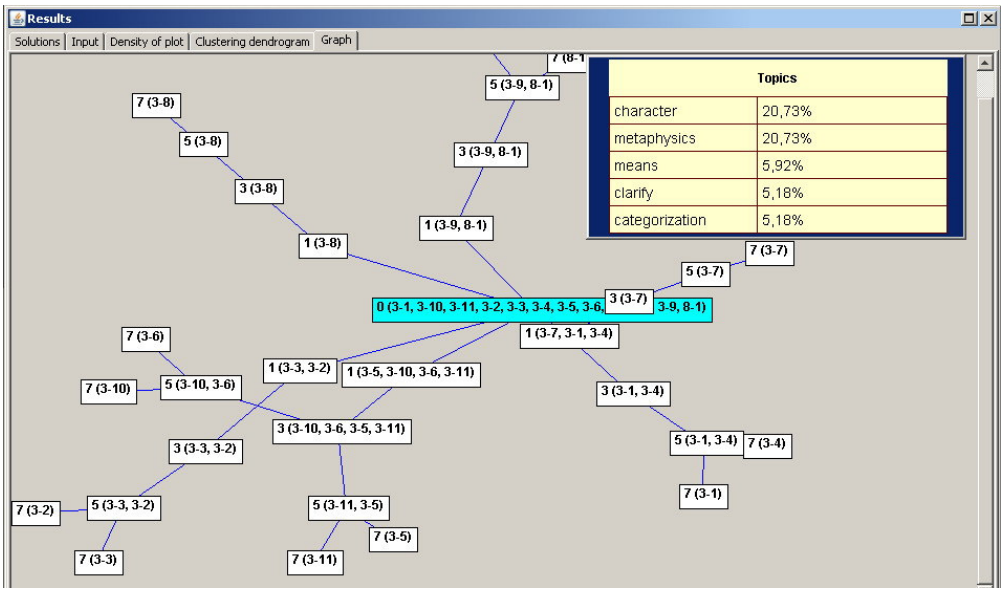
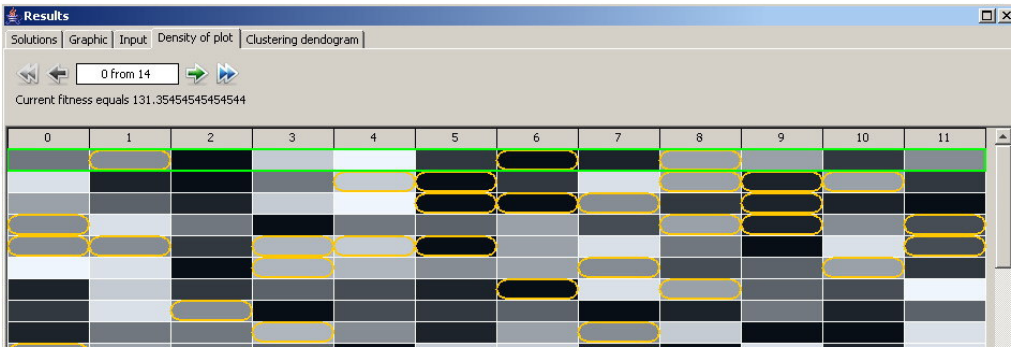


Figure 4: Example of the map of clusters and their parameters. Notation 7(8-1) means that on the 7 th step of clustering the cluster contains first sentence from 8 th abstract.

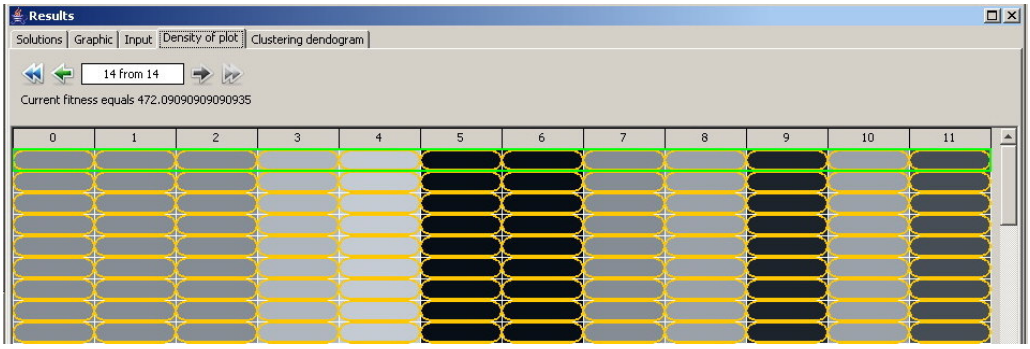
This is the cluster for the single sentence in second abstract numbered as 7(8-1). The sentence is about ontology so “metaphysics” has appeared as hyperonim for “ontology”.

During experiments of optimization the whole evolution of chromosome populations is stored. This information is further used to reveal building blocks (see Section 2). Building block elements are visualized with the help of genotype density plot which is shown in Figure 5. This allows relating good solutions observed during the process of evolution to properties of the system under optimization.

Genotype density plot is a function $P \times L \rightarrow A$, where A is an alphabet, P is a set of population species and L is a set of positions in a chromosome. The function possesses values from a set of alphabet values which are currently shown in gray on the application plots. In Figure 6 the number of tints is not enough to map to all 11 alphabet values. Building block elements are shown as chamfered rectangles. These blocks appear several times in a population. They are further used to "breed" parts of final population chromosomes concurrently.



(a)



(b)

Figure 5: Genotype density plot. Chromosome elements (genes) are arranged horizontally; different chromosomes in a population are arranged vertically.

Final population genotype density plot is shown in Figure 5 (b). One can see that the population genotype has been unified. That means that a stopping criterion has been satisfied in global extreme.

Checking the genotype to be unified is carried out not only visually but also with the help of the framework software.

5. Conclusion and Further Work

Summarizing the material above, we resolve that Evolutionary Modelling is a perspective experimental method for solving problems in Text Mining which can be treated as optimization problems. It can be also applied as a tool for semantic interpretations of modeling results.

At the same time the framework for Evolutionary Modelling considered in this paper needs further development. That development is planned in following directions:

- applying corpora technologies in Data subsystem;
- including Genetic Programming as a tool for working with graph models;
- developing online versions of EVOLIB subsystems.

We hope that this Evolutionary Modelling framework will be valuable for *evolution* of the area of Text Mining.

References

[1] Ahmed A. A. Radwan, Bahgat A. Abdel Latef, Abdel Mgeid A. Ali, Osman A. Sadek. (2006) Using Genetic Algorithm to Improve Information Retrieval Systems. Proc. of World Academy of Science, Engineering and Technology Vol. 17.

[2] Birchenhall, C.R., N. Kastrinos, and S. Metcalfe (1997). Genetic algorithms in evolutionary modelling, Journal of Evolutionary Economics, 7, 375-393.

[3] Bogatyrev, M. Latov, V. Stolbovskaya, I. (2007) Application of Conceptual Graphs in Digital Libraries. In: Digital libraries: Advanced Methods and Technologies, Digital Collections. Proceedings of the Ninth Russian National Research Conference - Pereslavl-Zalesskij: Pereslavl University. P.p. 109-115 (in Russian)

[4] Bogatyrev, M.Y. Tuhtin, V.V. (2008). Solving Some Text Mining Problems with Conceptual Graphs. In: Digital libraries: Advanced Methods And Technologies, Digital Collections. Proceedings of the Tenth Russian National Research Conference RCDL'2008 Dubna, JINR p.p. 31-36. (in Russian).

[5] Bournaud, I., Ganascia, J.-G. (1995) Conceptual Clustering of Complex Objects: A Generalization Space based Approach. Lecture Notes in Artificial Intelligence 954, Springer, P.p. 173—187.

[6] Boytcheva, S. Dobrev, P. Angelova, G. (2001). CGExtract: Towards Extraction of Conceptual Graphs from Controlled English. Lecture Notes in Computer Science № 2120, Springer Verlag.

[7] Cole, R. M. (1998) Clustering With Genetic Algorithms. University of Western Australia. 110 p.

[8] Fan, W. Gordon, M.D., Pathak, P. (2000) Personalization of search engine services for effective retrieval and knowledge management. In: Proc. International Conference on Information Systems (ICIS), Brisbane, Australia, 2000.

[9] Fan, W. Gordon, M.D., Pathak, P. (2004). Discovery of context-specific ranking functions for effective information retrieval using genetic programming, IEEE Transactions on knowledge and Data Engineering. Vol. 16, issue 4, p.p. 523 – 527.

[10] Gildea D., Jurafsky D. (2002) Automatic labeling of semantic roles. Computational Linguistics, 2002, v. 28, p.p. 245-288.

[11] Goldberg D.E. Genetic Algorithms in Search Optimization and Machine Learning. Addison-Wesley, Reading, MA, USA, 1989.

[12] Gordon, M.D. (1988). Probabilistic and genetic algorithms for document retrieval, Communications of the ACM, 31(10), pp: 1208-1218.

[13] Hensman, S, Dunnion, J. (2004). Automatically building conceptual graphs using VerbNet and WordNet. In: Proceedings of the 3rd International Symposium on

Information and Communication Technologies (ISICT), Las Vegas, June 16-18, 2004, pp.115-120.
[14] Holland, J.H. (1975). Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, MI, USA.
[15] Lamprier, S., Amghar, T., Levrat, B., Saubion, F. (2007). SegGen: a Genetic Algorithm for Linear Text Segmentation. In: Proceedings of the Twentieth International Joint Conference on Artificial Intelligence. AAAI Press, Menlo Park, California, 2007. p.p. 1647-1652
[16] Michalski, R. S., Stepp, R. E. (1983). Learning from observation: Conceptual clustering. Michalski, R. S.; Carbonell, J. G.; Mitchell, T. M. (Eds.) Machine Learning: An Artificial Intelligence Approach: 331–363, Palo Alto, CA: Tioga.
[17] Mineau, G. W., Godin R. (1995) Automatic Structuring of Knowledge Bases by Conceptual Clustering. IEEE Transactions on Knowledge and Data Engineering. Vol.7 , Issue 5. P.p. 824 – 829.
[18] Montes-y-Gómez, M. Gelbukh, A. López-López, A.R. Baeza-Yates (2001). Flexible comparison of conceptual graphs. In: Proceedings of the 12th International Conference and Workshop on Database and Expert Systems Applications
[19] Montes-y-Gomez, M., Gelbukh, A., Lopez-Lopez, M. (2002) Text Mining at Detail Level Using Conceptual Graphs. Lecture Notes In Computer Science; Vol. 2393. P. 122 – 136.
[20] Pathak, P. Gordon, M. Fan, W. (2000) Effective information retrieval using genetic algorithms based matching functions adaption, in: Proc. 33rd Hawaii International Conference on Science (HICS), Hawaii, USA.
[21] Robertson, A.M. Willet, P. Generation of equifrequent groups of words using a genetic algorithm, Journal of Documentation 50 (3), 1994, pp. 213–232.
[22] Sowa J. (1999) Conceptual Graphs: Draft Proposed American National Standard, International Conference on Conceptual Structures ICCS-99, Lecture Notes in Artificial Intelligence 1640, Springer 1999.
[23] Turney, P.D. (1999), Learning to Extract Keyphrases from Text, National Research Council, Institute for Information Technology, Technical Report ERB-1057. (NRC #41622)
[24] Vose, M. D. (1999). Random heuristic search. Theoretical Computer Science Vol. 229, Issue 1-2 P.p. 103 – 142.
[25] Zhong, J., Zhu H., Li J., Yu Y. (2002) Conceptual Graph Matching for Semantic Search. In: Proceedings of the 10 th International Conference on Conceptual Structures: Integration and Interfaces, Springer -Verlag p.p. 92–106.