Dmitry I. Ignatov, Sergei O. Kuznetsov, Jonas Poelmans (Eds.)

CDUD 2012 – Concept Discovery in Unstructured Data

Workshop co-located with the 10th International Conference on Formal Concept Analysis (ICFCA 2012)
May 2012, Leuven, Belgium

Volume Editors

Dmitry I. Ignatov
School of Applied Mathematics and Information Science
National Research University Higher School of Economics, Moscow, Russia

Sergei O. Kuznetsov
School of Applied Mathematics and Information Science
National Research University Higher School of Economics, Moscow, Russia

Jonas Poelmans
Faculty of Business and Economics
Katholieke Universiteit Leuven, Belgium

Printed in Belgium by the Katholieke Universiteit Leuven with ISBN 978-9-08-140991-9.

The proceedings are also published online on the CEUR-Workshop website in volume Vol-871 of a series with ISSN 1613-0073.

Preface

Concept discovery is a subarea of Knowledge Discovery in Databases (KDD) where concept models, such as Formal Concept Analysis (FCA), multimodal clustering, conceptual graphs and other, are used for gaining insight into the underlying conceptual structure of data. Traditional machine learning techniques are mainly focusing on structured data given by object-attribute tables, whereas most data available nowadays are given in unstructured, often textual, form. As compared to traditional data mining techniques, human-centered instruments of concept discovery actively engage domain experts in the discovery process.

This volume contains the papers presented at the 2nd International Workshop on Concept Discovery in Unstructured Data (CDUD 2012) held on May 10, 2012 at the Katholieke Universiteit Leuven, Belgium. This workshop welcomes papers describing innovative research on data discovery in complex data. Moreover, this workshop provides a forum for researchers and developers of data mining instruments, working on issues associated with analyzing unstructured data. This year the committee decided to accept 11 papers for publication in the proceedings. Each submission was reviewed by on average 3 program committee members.

A. Mestrovic presents an application of concept lattices to semantic matching in Croatian language. A. Chepovskiy et al. propose a method for automatic language identification for transliterated texts. X. Naidenova describes a novel neural network based data structure for inferring classification tests. A. Kravchenko et al. introduce an approach for expert search which is based on analyzing e-mail communication patterns. D. Ustalov et al. propose an ontology-based approach for text-to-picture synthesis. A. Skabin presents a computerized recognition system for hand-written historical manuscripts. A. Panchenko et al. extract semantic relations between concepts from Wikipedia using KNN algorithms. D. Fedyanin uses parameter identification methods for Markov models and applies them to influence analysis in social networks. S. Milyaev et al. discuss a new method for self-tuning semantic image segmentation. A. Vorobev proposes a probabilistic model for evaluating the quality level of projects, authors and experts in collaborative innovation platforms. D. Gnatyshak et al. present a novel pseudo-triclustering algorithm and applied it to online social network data. A. Bozhenyuk et al. discuss methods for maximum flow and minimum cost flow finding in fuzzy setting.

We would like to express our gratitude to all contributing authors and reviewers. We also want to thank our sponsors Amsterdam-Amstelland police, IBM Belgium, Research Foundation Flanders, Vlerick Management School, OpenConnect Systems and Higher School of Economics (Moscow, Russia). Finally, we should thank the authors of the EasyChair system which helped us to manage the reviewing process.

May 10, 2012
Leuven

Dmitry I. Ignatov
Sergei O. Kuznetsov
Jonas Poelmans

Organization

The 2nd International Workshop on Concept Discovery in Unstructured Data (CDUD 2012) was held on May 10, 2012 at the Katholieke Universiteit Leuven, Belgium. The workshop was co-located with the 10th International Conference on Formal Concept Analysis (ICFCA-2012). The inaugural edition of CDUD was held on June 25, 2011 at the Higher School of Economics in Moscow, Russia.

Program Chairs

Dmitry I. Ignatov	National Research University Higher School of Economics, Russia
Sergei O. Kuznetsov	National Research University Higher School of Economics, Russia
Jonas Poelmans	Katholieke Universiteit Leuven, Belgium

Program Committee

Simon Andrews	Sheffield Hallam University, United Kingdom
Guido Dedene	Katholieke Universteit Leuven, Belgium
Florent Domenach	University of Nicosia, Cyprus
Irina Efimenko	National Research University Higher School of Economics, Russia
Paul Elzinga	Amsterdam-Amstelland Police, The Netherlands
Boris Galitsky	University of Girona, Spain
Bernhard Ganter	Technische Universität Dresden, Germany
Yury Katkov	National Research University of Information Technologies, Mechanics and Optics, Russia
Natalia Loukachevitch	Moscow State University, Russia
Dmitry Mouromtsev	National Research University of Information Technologies, Mechanics and Optics, Russia
Xenia Naidenova	Military Medical Academy, Russia
Alexey A. Neznanov	National Research University Higher School of Economics, Russia
Sergei A. Obiedkov	National Research University Higher School of Economics, Russia
Simon Polovina	Sheffield Hallam University, United Kingdom
Uta Priss	Edinburgh Napier University, United Kingdom
Dominik Slezak	University of Warsaw and Infobright, Poland
Rustam Tagiew	Technische Universität Freiberg, Germany
Stijn Viaene	Katholieke Universiteit Leuven, Belgium

Johanna Voelker	University of Mannheim, Germany
Rostislav Yavorsky	Witology, Russia

Additional Reviewers

Ekaterina Cherniak, National Research University of Higher School of Economics, Russia
Alexandr Vorobev, Moscow State University and Witology, Russia

Sponsoring Institutions

Amsterdam-Amstelland police, The Netherlands
IBM, Belgium
OpenConnect Systems, USA
Research Foundation Flanders, Belgium
Vlerick Management School, Belgium
National Research University Higher of School Economics, Russia

Table of Contents

The Methods of Maximum Flow and Minimum Cost Flow Finding in Fuzzy Network

Alexandr Bozhenyuk[1], Evgeniya Gerasimenko[1], and Igor Rozenberg[2]

[1]Southern Federal University, Taganrog, Russia
AVB002@yandex.ru, e.rogushina@gmail.com
[2]Public Corporation "Research and Development Institute of Railway Engineers", Moscow, Russia
I.rozenberg@gismps.ru

Abstract. This article considers the problems of maximum flow and minimum cost flow determining in fuzzy network. Parameters of fuzzy network are fuzzy arc capacities and transmission costs of one flow unit represented as fuzzy triangular numbers. Conventional rules of operating with fuzzy triangular numbers lead to a strong "blurring" of their borders, resulting in loss of self-descriptiveness of calculations with them. The following technique of addition and subtraction of fuzzy triangular numbers is proposed in the presented paper: the centers are added (subtracted) by the conventional methods, and the borders of the deviations are calculated using linear combinations of the borders of adjacent values. The fact that the limits of uncertainty of fuzzy triangular numbers should increase with the increasing of central values is taken into account. To illustrate the proposed method numerical examples are presented.

Keywords: Fuzzy arc capacity, linear combination of borders, fuzzy triangular number, fuzzy flow.

1 Introduction

This article deals with flow problems arising in networks. The network corresponds to a directed graph $G = (X, A)$, where X – the set of nodes, A – the set of arcs with distinguished initial (source) and final (sink) nodes. Each arc $(x_i, x_j) \in A$ has capacity determining the maximum number of flow units, which can pass along the arc. The relevance of the tasks of maximum and minimum cost flow determining lies in the fact that the researcher can effectively manage the traffic, taking into account the loaded parts of roads, redirect the traffic, and choose the cheapest route. Suppose a network, which arcs have capacities (q_{ij}). Formulation of the problem of maximum flow finding in the network is reduced to maximum flow determining that can be passed along arcs of the network in view of their capacities [1]:

$$\begin{gathered}
\nu = \sum_{x_j \in \Gamma(s)} \xi_{sj} = \sum_{x_k \in \Gamma^{-1}(t)} \xi_{kt} \to \max, \\
\sum_{x_j \in \Gamma(x_i)} \xi_{ij} - \sum_{x_k \in \Gamma^{-1}(x_i)} \xi_{ki} = \begin{cases} \nu, & x_i = s, \\ -\nu, & x_i = t, \\ 0, & x_i \neq s,t, \end{cases} \\
0 \le \xi_{ij} \le q_{ij}, \ \forall (x_i, x_j) \in A.
\end{gathered} \tag{1}$$

In (1) ξ_{ij} – the amount of flow, passing along the arc (x_i, x_j); ν – the maximum flow value in the network; s – initial node (source); t – final node (sink); q_{ij} – arc capacity, $\Gamma(x_i)$ – the set of nodes, arcs from the node $x_i \in X$ go to, $\Gamma^{-1}(x_i)$ – the set of nodes, arcs to the node $x_i \in X$ go from. ξ_{ij} represents, for example, the amount of cars, going from the node $x_i \in X$ to the node $x_j \in X$. The first equation of (1) defines that we should maximize the total number of flow units emanating from the source (ν), which is equal to the total number of flow units entering the sink (ν). The second equation of (1) is a flow conservation constraint, which means that the total number of flow units emanating from the source (ν) must be equal to the total number of flow units entering the sink (ν) and the total number of flow units emanating from any node $x_i \neq s,t$ must be equal to the total number of flow units entering the node $x_i \neq s,t$. The third inequality of (1) is a bound constraint, which indicates that the flow of value ξ_{ij}, passing along any arc (x_i, x_j) must not exceed its arc capacity.

The task of minimum cost flow determining in a network can be formulated as follows: suppose we have a network, which arcs have two associated numbers: the arc capacity (q_{ij}) and transmission cost (c_{ij}) of one flow unit passing from the node $x_i \in X$ to the node $x_j \in X$. The essence of this problem is to find a flow of the given value ω from the source to the sink, which doesn't exceed the maximum flow in the graph ν and has minimal transmission cost. In mathematical terms the problem of minimum cost flow determining [2] in the network can be represented as follows:

$$\begin{gathered}
\sum_{(x_i, x_j) \in A} c_{ij} \cdot \xi_{ij} \to \min, \\
\sum_{x_j \in \Gamma(x_i)} \xi_{ij} - \sum_{x_k \in \Gamma^{-1}(x_i)} \xi_{ki} = \begin{cases} \omega, & x_i = s, \\ -\omega, & x_i = t, \\ 0, & x_i \neq s,t, \end{cases} \\
0 \le \xi_{ij} \le q_{ij}, \ \forall (x_i, x_j) \in A.
\end{gathered} \tag{2}$$

In (2) c_{ij} – transmission cost of one flow unit along the arc (x_i, x_j), ω – given flow value, that doesn't exceed the maximum flow ν in the network.

In practice, the arc capacities, transmission costs, the values of the flow entering the node and emanating from the node cannot be accurately measured according to their nature. Weather conditions, emergencies on the roads, traffic congestions, and repairs influence arc capacities. Variations in petrol prices, tolls can either influence transmission costs. Therefore, these parameters should be presented in a fuzzy form, such as fuzzy triangular numbers [3]. Thus, we obtain a problem statement of maximum and minimum cost flow problems in fuzzy conditions.

2 Literature Review of the Maximum and Minimum Cost Flow Determining Tasks

The problem of the maximum flow finding in a general form was formulated by T. Harris and F. Ross [4]. L. Ford and D. Fulkerson developed famous algorithm for solving this problem, called "augmented path" algorithm [5]. Maximum flow problem was considered in [1, 6].

There are different versions of the Ford-Fulkerson's algorithm. Among them there is the shortest path algorithm, proposed by J. Edmonds and R. Karp in 1972 [7], in which one can choose the shortest supplementary path from the source to the sink at each step in the residual network (assuming that each arc has unit length). The shortest path is found according to the breadth-first search.

Determining the maximum flow in the transportation network in terms of uncertainty has been studied less. In [8] a solution taking into account the interval capacities of arcs was proposed. S. Chanas [9] proposed to solve this problem by using so-called "fuzzy graphs". There are contemporary articles which solve the problem by the simplex method of linear programming [10].

Many researchers have examined the task of minimum cost flow finding in crisp conditions in the literature. Methods of its solution can be divided into graph techniques and the methods of linear programming. In particular, solutions by the graph methods are considered in [1, 6]. The advantages of this approach are great visualization and less cumbersome. The minimum cost flow is proposed to find by Busacker-Gowen and M. Klein's algorithms in [2]. In [2, 6] a task of minimum flow determining is considered as a task of linear programming. This approach is cumbersome.

The methods of minimum cost flow finding in networks in fuzzy conditions can be divided into two classes. The first class involves the use of conventional flow algorithms for determining the minimum cost flow, which operate with fuzzy data instead of crisp values and require cumbersome routines with fuzzy numbers. The second class of problems implies the use of "fuzzy linear programming", which was widely reported in the literature [11, 12].

Authors [13] consider the tasks of "fully fuzzy linear programming". These tasks are cumbersome and can not lead to optimal solutions in the minimum cost flow determination. The solution of fuzzy linear programming tasks by the comparison of fuzzy numbers based on ranking functions is examined in [14].

3 Presented Method of Operating with Fuzzy Triangular Numbers

Researcher is faced with the problem of fuzziness in the network, when considering the problems of maximum and minimum cost flow finding. Arc capacities, flow values, passing along the arcs, transmission costs per unit of goods cannot be accurately measured, so we will represent these parameters as fuzzy triangular numbers.

We will represent the triangular fuzzy numbers as follows: (a, γ, δ), where a – the center of the triangular number, γ – deviation to the left of the center, δ – deviation to the right of center, as shown in Fig. 1.

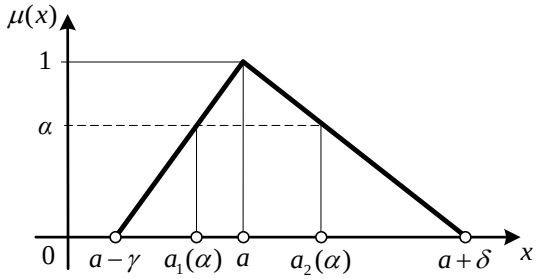


Fig. 1. Fuzzy triangular number.

Conventional operations of addition and subtraction of fuzzy triangular numbers are as follows: let $\tilde{A}_1$ and $\tilde{A}_2$ be fuzzy triangular numbers, such as $\tilde{A}_1 = (a_1, \gamma_1, \delta_1)$ and $\tilde{A}_2 = (a_2, \gamma_2, \delta_2)$. Therefore, the sum of triangular numbers can be written as: $\tilde{A}_1 + \tilde{A}_2 = (a_1 + a_2, \gamma_1 + \gamma_2, \delta_1 + \delta_2)$ and the difference represented as: $\tilde{A}_1 - \tilde{A}_2 = (a_1 - a_2, \gamma_1 + \delta_2, \delta_1 + \gamma_2)$ [3]. The disadvantage of the conventional methods of addition and subtraction of fuzzy triangular numbers is a strong "blurring" of the resulting number and, consequently, the loss of its self-descriptiveness. For example, when adding the same triangular number with itself, the borders of its uncertainty increase: (2,1,1) + (2,1,1) = (4, 2, 2) and (2,1,1) + (2,1,1) + (2,1,1) = (6, 3, 3). Generally, it is not true, because the center of the triangular number should increase, while its borders must remain constant. The fact that the degree of borders "blurring" of fuzzy number depends on the size of its center is not usually considered, when specifying the triangular fuzzy numbers. Therefore, the more the center, the more "blurred" the borders should be (while measuring 1 kg of material, we are talking "about 1 kg", implying the number "from 900 to 1100 g", but while measuring 1 t. of material, imply that "about 1 t." is the number "from 990 kg to 1110 kg").

Comparison of fuzzy triangular numbers according to various criteria is also very difficult and time-consuming. Consequently, following method is proposed to use

when operating with triangular fuzzy numbers. Suppose there are the values of arc capacities, flows or transmission costs in a form of fuzzy triangular numbers on the number axis. Then when adding (subtracting) the two original triangular fuzzy numbers their centers will be added (subtracted), and to calculate the deviations it is necessary to define required value by adjacent values. Let the fuzzy arc capacity (flow or transmission cost) "near $\tilde{x}'$" is between two adjacent values "near $\tilde{x}_1$" and "near $\tilde{x}_2$", $(x_1 \le x' \le x_2)$ which membership functions $\mu_{\tilde{x}_1}(x_1)$ and $\mu_{\tilde{x}_2}(x_2)$ have a triangular form, as shown in Fig. 2.

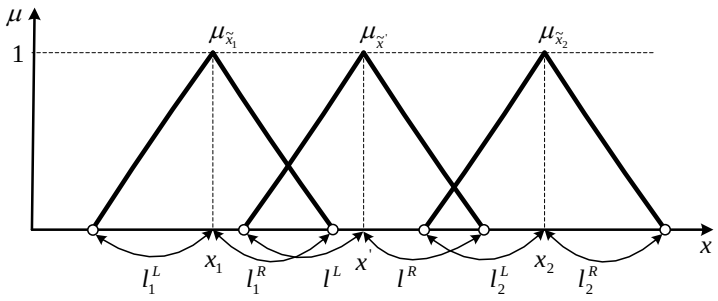


Fig. 2. Given values of arc capacities (flows or transmission costs).

Thus, one can set the borders of membership function of fuzzy arc capacity (flow or transmission cost) "near $\tilde{x}'$" by the linear combination of the left and right borders of adjacent values:

$$l^L = \frac{(x_2 - x)}{(x_2 - x_1)} \times l_1^L + \left(1 - \frac{(x_2 - x)}{(x_2 - x_1)}\right) \times l_2^L,$$
$$l^R = \frac{(x_2 - x)}{(x_2 - x_1)} \times l_1^R + \left(1 - \frac{(x_2 - x)}{(x_2 - x_1)}\right) \times l_2^R. \quad (3)$$

In (3) l^L is the left deviation border of required fuzzy number, l^R is the right deviation border. In the case when the central value of triangular number resulting by adding (subtracting) repeats the already marked value on the number axis, its deviation borders coincide with the deviation borders of the number marked on the number axis. If required central value is not between two numbers, but precedes the first marked value on the number axis, its deviation borders coincide with those of the first marked on the axis. The same applies to the case when the required central value follows the last marked value on the axis.

4 Solving the Task of Maximum Flow Finding in Fuzzy Network

The task of maximum flow finding in fuzzy network can be formulated as follows:

$$\begin{gathered}\tilde{\nu} = \sum_{x_j \in \Gamma(s)} \tilde{\xi}_{sj} = \sum_{x_k \in \Gamma^{-1}(t)} \tilde{\xi}_{kt} \to \max, \\ \sum_{x_j \in \Gamma(x_i)} \tilde{\xi}_{ij} - \sum_{x_k \in \Gamma^{-1}(x_i)} \tilde{\xi}_{ki} = \begin{cases} \tilde{\nu}, & x_i = s, \\ -\tilde{\nu}, & x_i = t, \\ \tilde{0}, & x_i \neq s,t, \end{cases} \\ \tilde{0} \le \tilde{\xi}_{ij} \le \tilde{q}_{ij}, \forall (x_i, x_j) \in A.\end{gathered} \tag{4}$$

In (4) $\tilde{\nu}$ is required maximum fuzzy flow value in the network; $\tilde{\xi}_{ij}$ – fuzzy amount of flow, passing along the arc (x_i, x_j); $\tilde{q}_{ij}$ – fuzzy capacity of the arc (x_i, x_j); $\tilde{0}$ is fuzzy number of the form (0, 0, 0), as it reflects the absence of the flow.

Let's consider an example, illustrating the solution of this problem, represented in Fig. 3. Let network, representing the part of the railway map, is given in a form of fuzzy directed graph, obtained from GIS "Object Land" [15, 16]. Let the node x_1 is a source, node x_{12} is a sink. The values of arc capacities in the form of fuzzy triangular numbers are defined above the arcs. It is necessary to calculate the maximum flow value between stations "Kemerovo" (x_1) and "Novosibirsk-Gl." (x_{12}) according to Edmonds-Karp's algorithm [7] and the method, described for operations with fuzzy triangular numbers. Determining of maximum flow is based on sending flows along the arcs of the network until one cannot send any additional unit of flow from the source to the sink. Edmonds-Karp's algorithm represents the choice of the shortest supplementary path from the source to the sink at each step in the residual network (assuming that each arc has unit length). Fuzzy residual network contains the arcs of the form (x_i, x_j) with the fuzzy residual arc capacity $\tilde{q}_{ij} - \tilde{\xi}_{ij}$, if the arcs (x_i, x_j) have the flow value $\tilde{\xi}_{ij} < \tilde{q}_{ij}$ in the initial network; and the arcs of the form (x_j, x_i) with the residual arc capacity $\tilde{\xi}_{ij}$, if the arcs (x_i, x_j) have the flow value $\tilde{\xi}_{ij} > \tilde{0}$. The arc (x_i, x_j) is called "saturated" when the flow, passing along it, equals to arc capacity $\tilde{q}_{ij}$. Other words, residual arc capacity defines how many flow units can be sent along the arc (x_i, x_j) to reach arc capacity. Residual arc capacity of arc saturated arc (x_i, x_j) is zero.

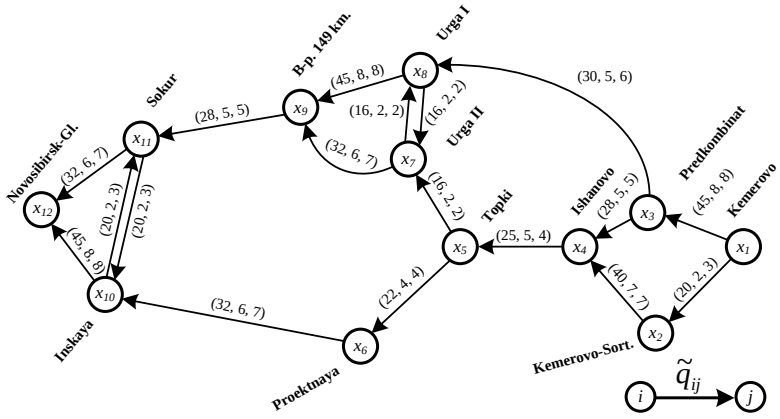


Fig. 3. Initial network.

Therefore, the algorithm proceeds as follows: the first iteration of the algorithm performs an augmenting chain $x_1x_3x_8x_9x_{11}x_{12}$. Push the flow, equals to (28, 5, 5) units along it. The arc (x_9, x_{11}) becomes saturated, consequently, fuzzy residual capacity of the arc (x_9, x_{11}) equals to (0, 0, 0). Let's define the fuzzy residual capacities of the remaining arcs of augmenting chain. The arc (x_1, x_3) has fuzzy residual capacity equals to (45, 8, 8) – (28, 5, 5). Thus, the central value of the resulting number is 17. It is located between adjacent arc capacities: (16, 2, 2) and (20, 2, 3) of the original graph as shown in Fig. 4.

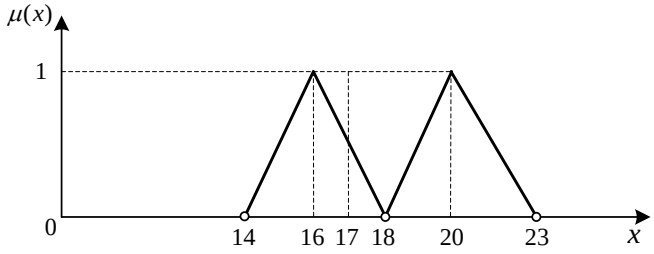


Fig. 4. Fuzzy triangular number with a center equals to 17 and its adjacent numbers.

Compute the left and the right deviation borders of the fuzzy triangular number with a center of 17 according to (3). Thus, we obtain a fuzzy triangular number of the form (17, 2, 2.25) units.

Fuzzy residual capacity of the arc (x_3, x_8) is (30, 5, 6) – (28, 5, 5). Consequently, we obtain a fuzzy triangular number with a center of 2, located to the left of fuzzy triangular number of the form (16, 2, 2). Deviation borders of the required number coincide with deviation borders of the number (16, 2, 2). Thus, we obtain a fuzzy triangular number of a type (2, 2, 2) units.

Fuzzy residual capacity of the arc (x_8, x_9) equals to (17, 2, 2.25) units, similarly with the arc (x_1, x_3).

Finally, fuzzy residual capacity of the arc (x_{11}, x_{12}) is equal to (32, 6, 7) – (28, 5, 5), i.e. we obtain fuzzy number (4, 2, 2) units and fuzzy residual capacity of the arc (x_{12}, x_{11}) equals to (28, 5, 5) units. Fuzzy residual capacities of the arcs $(x_3, x_1), (x_8, x_3), (x_9, x_8), (x_{11}, x_9), (x_{12}, x_{11})$ are (28, 5, 5) units.

The second iteration of the algorithm gives the augmenting chain $x_1 x_2 x_4 x_5 x_6 x_{10} x_{12}$. Push the flow equals to (20, 2, 3) units along it. The arc (x_1, x_2) becomes saturated, consequently, fuzzy residual capacity of the arc (x_1, x_2) equals to (0, 0, 0). Fuzzy residual capacity of the arc (x_2, x_4) is (40, 7, 7) – (20, 2, 3), i.e. we obtain a fuzzy triangular number (20, 2, 3) units. Fuzzy residual capacity of the arc (x_4, x_5) is the difference between the numbers (25, 5, 4) and (20, 2, 3). Thus, we get a number with a center of 5, located to the left of the number (16, 2, 2), i.e. (5, 2, 2) units. Fuzzy residual capacity of the arc (x_5, x_6) is equal to (22, 4, 4) – (20, 2, 3), i.e., (2, 2, 2) units. Fuzzy residual capacity of the arc (x_6, x_{10}) is equal to (32, 6, 7) – (20, 2, 3), i.e. (12, 2, 2) units. Fuzzy residual capacity of the arc (x_{10}, x_{12}) is (45, 8, 8) – (20, 2, 3), i.e., (25, 5, 4) units. Fuzzy residual capacities of the arcs $(x_2, x_1), (x_4, x_2), (x_5, x_4), (x_6, x_5), (x_{10}, x_6), (x_{10}, x_{12})$ are (20, 2, 3) units.

The third iteration of the algorithm performs the augmenting chain $x_1 x_3 x_4 x_5 x_6 x_{10} x_{12}$. Push the flow equals to (2, 2, 2) units along. The arc (x_5, x_6) becomes saturated. Let's define fuzzy residual capacities of the remaining arcs of the augmenting chain. Fuzzy residual capacity of the arc (x_1, x_3) is (17, 2, 2.25) – (2, 2, 2), i.e. (15, 2, 2) units. Fuzzy residual capacity of the arc (x_3, x_1) is (30, 5, 6) units. Fuzzy residual capacity of the arc (x_3, x_4) is equal to (28, 5, 5) – (2, 2, 2). We get the number with a center of 26, located between adjacent values (25, 5, 4) and (28, 5, 5).

Compute the left and the right deviation borders of the fuzzy triangular number with a center of 26 according to (3). Thus, we obtain a fuzzy triangular number of the form (26, 5, 4.33) units.

Fuzzy residual capacity of the arc (x_4, x_5) is equal to (5, 2, 2) – (2, 2, 2), i.e. (3, 2, 2) units. Fuzzy residual capacity of the arc (x_6, x_{10}) is equal to (12, 2, 2) – (2, 2, 2), i.e. (10, 2, 2) units. Fuzzy residual capacity of the arc (x_{10}, x_{12}) is equal to (25, 5, 4) – (2, 2, 2), i.e. we obtain a fuzzy number with a center of 23, located between adjacent values (22, 4, 4) and (25, 5, 4), therefore, the left deviation border of the number with a center of 23 equals to 4.33, the right deviation border is 4. We obtain fizzy triangu-

lar number (23, 4.33, 4) units. Fuzzy residual capacities of the arcs $(x_5, x_4), (x_6, x_5), (x_{10}, x_6), (x_{10}, x_{12})$ are (22, 4, 4) units.

After execution of three iterations of the algorithm it is impossible to pass any single additional flow unit. The total flow is (28, 5, 5) + (20, 2, 3) + (2, 2, 2) units. Therefore, we obtain a fuzzy triangular number with a center of 50, located to the right of the number (45, 8, 8) with the borders, repeated deviations of the number 45: (50, 8, 8) units.

Thus, the maximum flow value between the stations "Kemerovo" and "Novosibirsk-Gl." is (50, 8, 8) units. Let us carry out an interpretation of the results: the maximum flow between the given stations can not be less than 42 and more than 58 units, with the highest degree of confidence it will be equal to 50 units. But with changes in the environment, repairs on the roads, traffic congestions the flow is guaranteed to lie in the range from 42 to 58 units. Fuzzy optimal flow distribution along the arcs and labels of the nodes is shown in Fig. 5. Saturated arcs are bold.

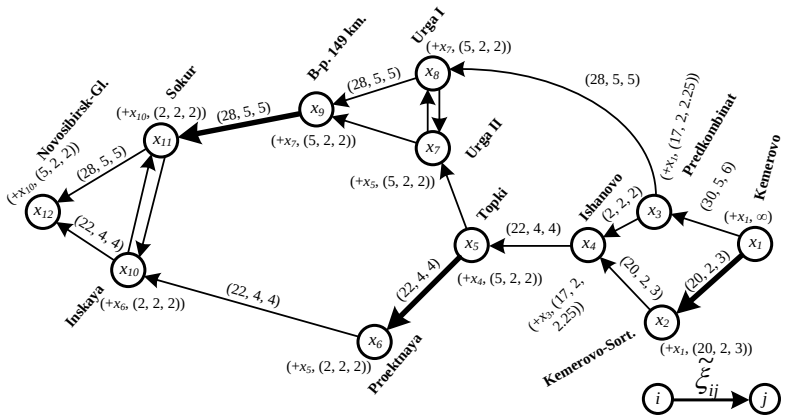


Fig. 5. Network with maximum flow of (50, 8, 8) units.

5 Solving the Task of Minimum Cost Flow Determining in Fuzzy Network

Consider the problem of minimum cost flow finding in a network according to fuzzy values of arc capacities, flows and transmission costs of one flow unit.

$$\sum_{(x_i,x_j)\in A} \tilde{c}_{ij}\cdot\tilde{\xi}_{ij} \to \min,$$

$$\sum_{x_j\in\Gamma(x_i)} \tilde{\xi}_{ij} - \sum_{x_k\in\Gamma^{-1}(x_i)} \tilde{\xi}_{ki} = \begin{cases} \tilde{\omega}, & x_i = s, \\ -\tilde{\omega}, & x_i = t, \\ \tilde{0}, & x_i \neq s,t, \end{cases} \tag{5}$$

$$\tilde{0} \le \tilde{\xi}_{ij} \le \tilde{q}_{ij},\ \forall (x_i,x_j)\in A.$$

In (5) $\tilde{c}_{ij}$ – fuzzy transmission cost of one flow unit along the arc (x_i, x_j), $\tilde{\omega}$ – given fuzzy flow value, that doesn't exceed the maximum flow $\tilde{v}$ in the network.

Let us turn to the graph, shown in Fig. 3. Fuzzy values of transmission costs in addition to fuzzy arc capacities are given in this task:

$\tilde{c}_{x_1x_2} = (12,3,3);\ \tilde{c}_{x_1x_3} = (6,1,2);\ \tilde{c}_{x_3x_4} = (10,2,3);\ \tilde{c}_{x_2x_4} = (18,4,5);\tilde{c}_{x_3x_8} = (4,1,1);$

$\tilde{c}_{x_4x_5} = (12,3,3);\ \tilde{c}_{x_5x_7} = (20,5,6);\ \tilde{c}_{x_8x_7} = (15,4,4);\tilde{c}_{x_7x_8} = (15,4,4);\ \tilde{c}_{x_8x_9} = (21,6,7);$

$\tilde{c}_{x_7x_9} = (10,2,3);\ \tilde{c}_{x_5x_6} = (30,8,9);\tilde{c}_{x_6x_{10}} = (8,2,2);\ \tilde{c}_{x_9x_{11}} = (19,5,5);$

$\tilde{c}_{x_{11}x_{10}} = (32,7,12);\ \tilde{c}_{x_{10}x_{11}} = (32,7,12);\tilde{c}_{x_{11}x_{12}} = (25,7,8);\ \tilde{c}_{x_{10}x_{12}} = (20,5,6).$

It is necessary to find a flow value $\tilde{\omega}$ of (45, 8, 8) units from the source to the sink, which has a minimal cost. Consider the Busacker-Gowen's [2] algorithm, taking into account the fuzzy capacities and costs to solve this problem:

Step 1. Assign all arc flows and the flow rate equal to zero.

Step 2. Determine the modified arc costs $\tilde{c}^*_{ij}$ that depend on the value of the already found flow as follows:

$$\tilde{c}^*_{ij} = \begin{cases} \tilde{c}_{ij},\ if\ \tilde{0} \le \tilde{\xi}_{ij} \le \tilde{q}_{ij}, \\ \infty,\ if\ \tilde{\xi}_{ij} = \tilde{q}_{ij}, \\ -\tilde{c}_{ji},\ if\ \tilde{\xi}_{ij} > 0. \end{cases}$$

Step 3. Find the shortest chain (in our case – the chain of minimal cost) [2] from the source to the sink taking into account arc costs $\tilde{c}^*_{ij}$, found in the step 1. Push the flow along this chain until it ceases to be the shortest. Receive the new flow value by adding the new flow value, passing along the considered chain, to the previous one. If the new flow value equals to $\tilde{\omega}$, then the end. Otherwise, go to the step 2.

Solve this problem, taking into account fuzzy arc capacities costs.

Step 1. Assign all $\tilde{\xi}_{ij} = 0$.

Step 2. Determine $\tilde{c}^*_{ij} = \tilde{c}_{ij}$.

Step 3. Find the shortest path by the Ford's algorithm [1]: $x_1x_3x_8x_9x_{11}x_{12}$ of the total cost of (75, 8, 8) standard units. Push the flow, equals to (28, 5, 5) units along this chain.

Step 2. Define the new modified fuzzy arc costs: $\tilde{c}^*_{x_1x_3} = (6,1,2)$; $\tilde{c}^*_{x_3x_1} = -(6,1,2)$; $\tilde{c}^*_{x_3x_8} = (4,1,1)$; $\tilde{c}^*_{x_8x_3} = -(4,1,1)$; $\tilde{c}^*_{x_8x_9} = (21,6,7)$; $\tilde{c}^*_{x_9x_8} = -(21,6,7)$; $\tilde{c}^*_{x_9x_{11}} = \infty$; $\tilde{c}^*_{x_{11}x_9} = -(19,5,5)$; $\tilde{c}^*_{x_{11}x_{12}} = (25,7,8)$; $\tilde{c}^*_{x_{12}x_{11}} = -(25,7,8)$.

Step 3. Find the shortest path using the obtained modified costs: $x_1x_3x_4x_5x_6x_{10}x_{12}$ of the total cost of (86, 8, 8) standard units. Push the flow, equals to (17, 2, 2.25) units along this chain. As a result, we obtain the total flow equals to (45, 8, 8) units, having a total transmission cost along the network, equals to (28, 5, 5) × ((75, 8, 8) + (86, 8, 8)) = (3562, 8, 8) standard units. There are fuzzy flow values $\tilde{\xi}_{ij}$ under the arcs and fuzzy transmission costs $\tilde{c}_{ij}$ of optimal fuzzy flow values $\tilde{\xi}_{ij}$ above the arcs of the graph, saturated arcs are bold as shown in Fig. 6.

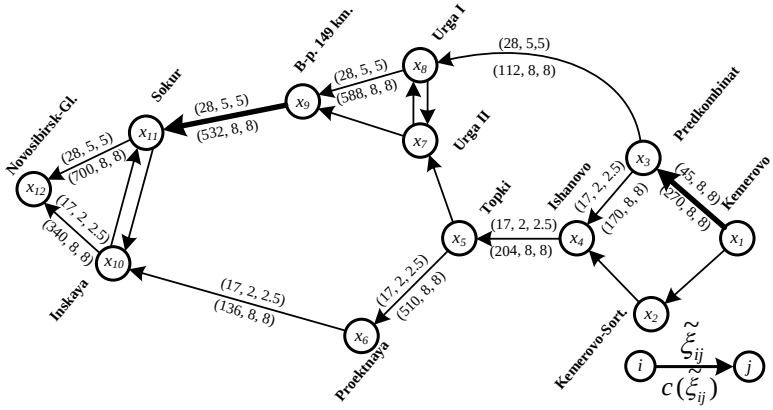


Fig. 6. Network with the flow of (45, 8, 8) units and transmission costs of each arc of the total cost (3562, 8, 8) standard units.

6 Conclusion

This paper examines the problems of maximum and minimum cost flow determining in networks in terms of uncertainty, in particular, the arc capacities, as well as the transmission costs of one flow unit are represented as fuzzy triangular numbers. The technique of addition and subtraction of triangular numbers is considered. Presented technique suggests calculating the deviation borders of fuzzy triangular numbers based on the linear combinations of the deviation borders of the adjacent values. The fact that the limits of uncertainty of fuzzy triangular numbers should increase with the increasing of central values is taken into account. Advantage of the proposed method lies in the fact that operations with fuzzy triangular numbers don't lead to a strong

"blurring" of their deviation borders, it makes calculations with such numbers more effective.

Acknowledgments. This work has been supported by the Russian Research project № 11-01-00011a.

References

1. Christofides, N.: Graph Theory: An Algorithmic Approach. Academic Press, New York, London, San Francisco (2006)
2. Hu, T.C.: Integer Programming and Network Flows. Addison-Wesley Publishing Company, Ontario (1970)
3. Dubois, D., Prade, H.: Operations on Fuzzy Numbers. J. Systems Sci. 9 (6), 613--626 (1978)
4. Harris T. E., Ross F. S.: Fundamentals of a method for Evaluating rail net capacities. (U) RAND Corporation, Research Memorandum RM-1573 (1956)
5. Ford, L.R., Fulkerson, D.R.: Flows in Networks. Princeton University Press, Princeton (1962)
6. Minieka, E.: Optimization Algorithms for Networks and Graphs. Marcel Dekker, Inc., New York and Basel (1978)
7. Edmonds, J., Karp, R.M.: Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems. Journal of the Association for Computing Machinery, vol. 19 (2), 248--264 (1972)
8. Rozenberg, I., Starostina, T.: Minimal Cost Flow Problem in a Graph with Flow Intensification and Fuzzy Parameters. In: 13th Zittau East West Fuzzy Colloquium, pp. 176--184. Hochschule Zittau Goerlitz, Zittau (2006)
9. Chanas, S., Delgado, M., Verdegay, J.L., Vila, M.: Fuzzy Optimal Flow on Imprecise Structures. European Journal of Operational Research, vol. 83, 568--580 (1995)
10. Kumar, A., Kaur, M.: A Fuzzy Linear Programming Approach to Solve Fuzzy Maximal Flow Problems. International Journal of Physical and Mathematical Sciences, vol. 1 (1), 6--12 (2010)
11. Thakre, P. A., Shelar, D. S., Thakre, S. P.: Solving Fuzzy Linear Programming Flow Problem as Multi Objective Linear Programming Problem. In: Proc. of the World Congress on Engineering, vol. II, London, U.K. (2009)
12. Ganesan K., Veeramani P.: Fuzzy Linear Programs with Trapezoidal Fuzzy Numbers. Ann Oper Res., 305--315 (2006)
13. Kumar A., Kaur J., Singh, P.: Fuzzy Optimal Solution of Fully Fuzzy Linear Programming Problems with Inequality Constraints. International Journal of Mathematical and Computer Sciences 6:1, 37--41 (2010)
14. Yoon, K. P.: A Probabilistic Approach to Rank Complex Fuzzy Numbers. Fuzzy Sets and Systems 80, 167--176 (1996)
15. Rozenberg, I. N., Gittis, C. A., Svyatov, D. C.: Geoinformation System Object Land. In: Proc. IPI RAN "Systems and Means of Informatics". Science, Moscow (2000)
16. Object Land, http://www.objectland.ru/

Language Identification for Texts Written in Transliteration

Andrey Chepovskiy, Sergey Gusev, Margarita Kurbatova

Higher School of Economics,
Data Analysis and Artificial Intelligence Department,
Pokrovskiy boulevard 11, 109028 Moscow, Russia
achepovskiy@hse.ru, unk379@mail.ru, rityastik@gmail.com

Abstract. The problem of identification of natural languages for the texts written in transliteration is considered. We consider a method of identification of five Slavic languages for texts written with use of a Latin transliteration. We use two ways of creation models for such texts and compare results of those application.

Keywords: statistical text model; natural language identification; transliteration.

1 Introduction

In various type of information systems intended for automatic processing of large amounts of texts in natural languages various data recognition problems are actual. The requirement of automating textual data processing brings specific importance to the language identification problem of a text or a part of a text.

At present time the sufficiently accurate language recognition methods for long texts consisting of tens of sentences are known [1]. Models based on frequencies of letter combinations are widely used to identify language of a text [2, 3]. It was noted in [3] that it is possible to use rank methods for language identification of a text, but they are not suitable for short texts. Also in [3] it is concluded that the language identification problem for short texts segments is still actual, and higher accuracy is achieved at the expense of larger model size and slower processing.

In [4] a language identification method for texts in natural language was studied. This method was applied to texts written in native alphabets for corresponding languages and good results were achieved. In current work we consider using the same method for language identification of a text written in Latin transliteration.

We use a set of five Slavic languages: Russian, Ukrainian, Byelorussian, Bulgarian and Macedonian. All of these languages use Cyrillic alphabet as native, and transliteration must be used to write in Latin.

The main problems related to language identification process are:

- A wide variety of conflicting transliteration tables;

- A frequent use of transliteration rules which do not meet any of standard transliteration tables.
- Amount of texts in transliteration is not enough for construction of training sets that are used in language identification algorithms.

To solve the first problem several transliteration tables should be used for each of the languages. Training text sets were created with an automatic text generation process from Cyrillic texts by using transliteration tables.

2 Statistical model for a natural language text

The text language identification problem is a pattern recognition problem, and its solution can be based on a probabilistic model. A Bayesian classifier can be applied to a string of characters assuming that we know statistical characteristics of characters for texts in specific language or texts belonging to a given class.

Let's consider a string s which consists of N characters c_n $(n = 1,\ldots, N)$ that belong to alphabet Σ. Further we will use the following notation: $s = c_1c_2\ldots c_N$ - a specific value of the string, $s_i = c_i$ – a value of a character which is at i-th position in the string. For solving the problem this string must be assigned to one of the classes $Y_l (l = 1, \ldots, K)$, where Y_l denotes one of K languages.

We assume that every class defines some kind of probability distribution on the set of all possible strings. In that case it is possible to apply a statistical criterion of maximum likelihood to determine a class which contains the string being classified.

The probability of a fact that string s will appear in some language equals to product of probabilities that each character of this string will appear in this language provided that all preceding characters will appear in this language too:

$$\begin{aligned} &P(s = c_1 \ldots c_N) = P(s_N = c_N \mid s_1 = c_1, \ldots, s_{N-1} = c_{N-1}) * \\ &P(s_{N-1} = c_{N-1} \mid s_1 = c_1, \ldots, s_{N-2} = c_{N-2}) * \ldots * P(s_1 = c_1) \end{aligned} \quad (1)$$

Let's assume that the probability distribution for a character at i-th position depends on probability distribution of not more than k preceding characters. In this case equation (1) can be written as following:

$$P(s_i = c_i \mid s_1 = c_1, \ldots, s_{i-1} = c_{i-1}) = P(s_i = c_i \mid s_{i-k} = c_{i-k}, \ldots, s_{i-1} = c_{i-1}) \quad (2)$$

An estimation of the conditional probabilities is performed on the training set. For this purpose, the frequencies of all substrings of lengths less than $k+2$ are calculated, and the estimated value of conditional probability for the next character is a ratio of the frequencies of the corresponding substrings:

$$P(Y_l, s_i = c_i \mid s_{i-m} = c_{i-m}, \ldots, s_{i-1} = c_{i-1}) = \frac{f(c_{i-m} \ldots c_i)}{f(c_{i-m} \ldots c_{i-1})}, \forall m \le k, \quad (3)$$

f(X) – the frequency of substring ***X*** in the training set.

The estimated value of probability of string ***s*** appearance in class Y_l is defined as follows:

$$P(Y_l, s) = P(Y_l, s_N = c_N \mid s_{N-k} = c_{N-k}, ..., s_{N-1} = c_{N-1}) * \\ P(Y_l, s_{N-1} = c_N \mid s_{N-k-1} = c_{n-k-1}, ..., s_{N-2} = c_{N-2}) * ... * P(Y_l, s_1 = c_1) \quad (4)$$

The classified string is assigned to the class with the highest probability estimate.

3 Algorithm implementation

First, each natural language text is converted to a set of words consisting of lower cased characters belonging to the native alphabet of the language. It forms a frequency dictionary of substrings having lengths in the range ***[1, k+1]*** taken into account the number of word occurrences in the text. This process is executed during the construction of a string model for a given language. This construction is based on training set of texts. The model is represented as a finite state machine with states marked with preceding character sequences and transitions marked with the next character and corresponding conditional probability.

A single space is appended to the end of each word of the text, then the word is passed as input to the finite state machine. The initial state of the machine corresponds to character sequence consisting of ***k*** spaces. According the formulas (1) – (4) a probability of transition to the next state from the current one by each of the characters is being calculated. A probability of appearance of the given word is a product of probabilities of all transitions that occurred during the machine operation. The probability of a text is a product of probabilities of all its words.

For the language identification a probability of the text appearance for models of every natural language is estimated. The language of the model with the highest probability is assigned to the text.

4 Quality of the text language identification

For the language identification of a text fragment a numeric estimate of its correspondence to a natural language text model can be calculated. Let the text fragment s consist of ***N*** characters. A probability of its appearance in the text written in ***l***-th language can be estimated with the formula (4). Then an estimation of this text fragment correspondence to the ***l***-th language will be calculated as:

$$E_l(s) = \frac{\ln(P(Y_l, s))}{N} + const, \quad (5)$$

$P(Y_l, s)$ – the probability of the string ***s*** appearance in the language Y_l;

N – the number of characters in the string ***s***;
const – a normalizing constant.
The expected value of this estimation doesn't depend on the length of text fragment. We choose the language with a maximum value of the estimation $E_l(s)$.

5 Initial data

Transliteration tables for all of five languages were constructed. Several different tables were used for each language – from 4 to 10, depending on language.

Cyrillic text sets consisting of at least 500 thousands characters were made for each of the five languages. These sets were automatically transliterated into Latin alphabet with each of the transliteration tables. Some tables contain ambiguous translation rules (i.e. there are several versions of Latin character combinations for one Cyrillic character). When several different rules were possible then only one of the rules was chosen randomly. These texts subsequently were used as training texts for creation of text models.

Test sets consisting of at least 50 thousands characters were constructed for each language. Test sets are real texts written in transliteration taken from various sources. We will call texts written in transliteration as transliterated texts, and texts written in native alphabets of corresponding languages as native texts.

To compare language identification quality for transliterated texts with language identification quality for native texts the following training and text sets of the same size were made:

- Cyrillic sets for the five languages being considered;
- Sets for 31 languages which use Latin alphabet as native.

6 Experimental results

We evaluate the quality of our language identification method by calculating precision and recall for individual languages. When we identify the language of a text sample of known language we can determine whether it was identified correctly or not. For a set of test samples we know the number of correctly identified samples as well as the number of identification errors for each language. For a given language precision can be calculated as a ratio of the number of correctly identified text samples in this language to the overall number of samples identified to this language. Recall is a ratio of correctly identified text samples in this language to the number of all samples in this language in the test set. We use F-measure to combine precision and recall to a single value, which is defined as harmonic mean of precision and recall.

To evaluate the quality of the language identification method we created a set of language models. Each model was trained on a text from the training set and marked with the language of that text. Then several sets of text samples were created from the texts belonging to the test set. Each set included 1000 text samples for each language

of the same length. The samples were generated as text fragments starting from a randomly selected position inside text with the restriction that the position must be a beginning of a word. Every text sample was evaluated with every model and the language of the model with the highest estimate was chosen as the identified language of the sample. Then the values of F-measure were calculated for every language.

The set of models included 5 models for transliterated texts of Slavic languages as well as 31 models for native texts with languages which use Latin as their native alphabet. The test set contained 36 texts for the same languages.

We considered two methods of language identification for transliterated texts. The first method uses several text models for each language. Each model corresponds to a particular transliteration table for the language. The model is trained on a text which was generated using that transliteration table. So the set of models contains several different models for each language. If a text sample receives the highest estimate for any of the models for a particular language then the sample is identified as belonging to that language. The second method uses exactly one text model for each language. The model is trained on a text which is a concatenation of all texts generated for the language by using all its transliteration tables.

Figures 1 and 2 show dependence of the language identification F-measure for transliterated texts on the text sample length. Using several models for each language instead of one model does not lead to significant improvement of the text language identification.

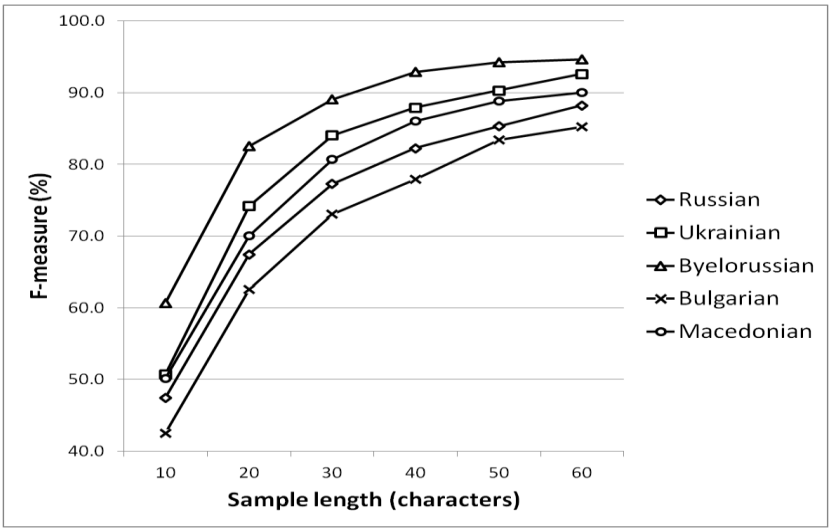


Fig. 1. Dependence of the F-measure on text sample length when using several models for one language.

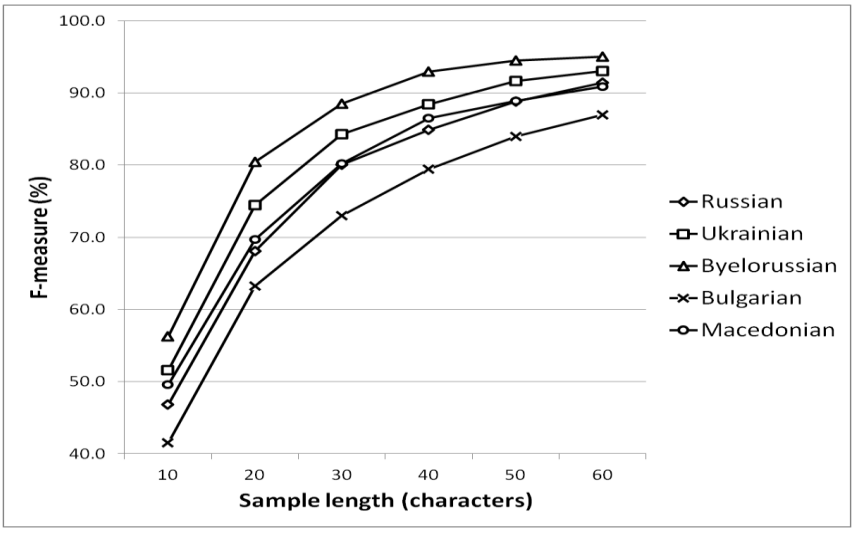


Fig. 2. Dependence of the F-measure on text sample length when using one model for one language.

For comparison figure 3 shows the language identification quality for native texts. Results were obtained by training and testing of five language models for the Cyrillic sets for the same five Slavic languages.

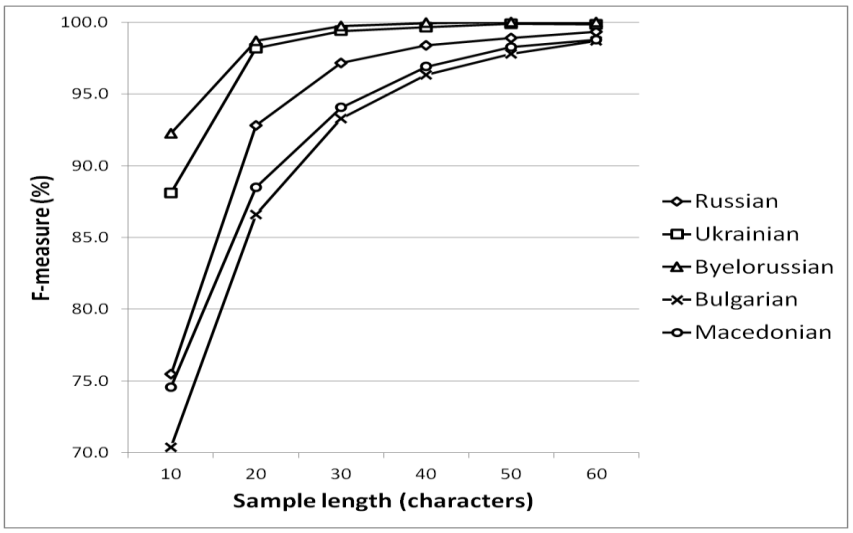


Fig. 3. Dependence of the F-measure on text sample length for language identification of native texts.

From figure 3 one can see that language identification quality for transliterated texts is significantly lower than for native texts. The most substantial part of accuracy loss occurs when transliterated text incorrectly classifies to model representing transliterated text of another language. There are much fewer errors when transliterated text classifies to a language that uses Latin alphabet.

To illustrate this fact, we can measure the quality of separation of transliterated texts from native texts written in Latin. It is much higher than language identification quality for transliterated text. Figure 4 shows F-measure values for separation of transliterated texts for various text sample lengths.

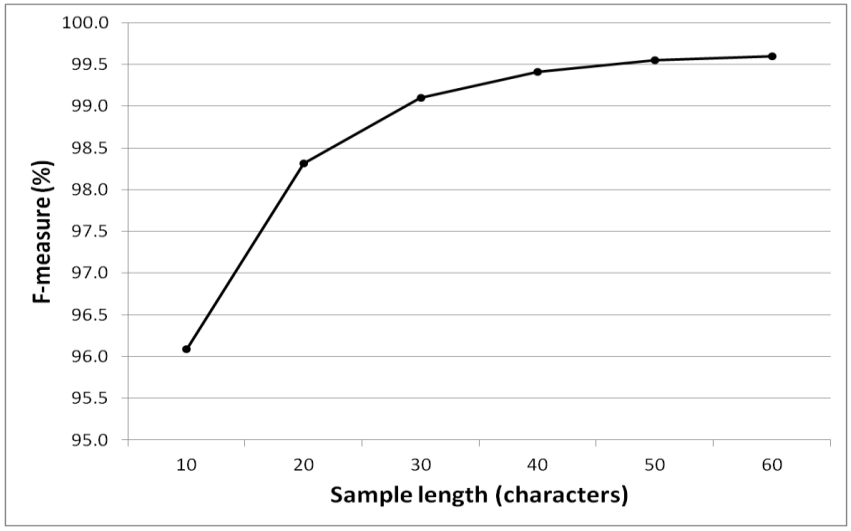


Fig. 4. F-measure values for separation of transliterated texts for various text sample lengths.

7 Conclusion

The text model considered in this article allows to successfully separate texts written in languages with Latin-based alphabets from texts written in Latin transliteration for languages with Cyrillic alphabets. The F-measure of such separation reaches 98% for short texts consisting of only 20 characters.

Identification accuracy of languages using Cyrillic alphabet for transliterated texts reaches more than 80% for texts consisting of 40 characters.

It was observed that using several different text models for a single language does not give a significant advantage over using a single model for a language.

References

1. McNamee, B.P.: Language identification: a solved problem suitable for undergraduate instructionl Journal of Computing Sciences in Colleges, 20(3), P. 94–101 (2005)
2. Cavnar, W. B., Trenkle, J. M.: N-gram-based text categorization. In.: Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, US, P. 161–175 (1994)
3. Vatanen, T, Väyrynen, J.J., Virpioja, S.: Language identification of short text segments with n-gram models. In.: Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10), P. 3423–3430 (2010)
4. Gusev, G., Chepovskiy, A.: The model for identification of a natural language of the text. Business Informatics, 3(17), P. 31–35 (2011)

On Parameter Identification Methods for Markov Models Applied to Social Networks

Denis Fedyanin

V.A. Trapeznikov Institute of Control Sciences, Russian Academy of Sciences
Russia, Moscow, 117997, Profsoyuznaya ulitsa, 65

dfedyanin@inbox.ru

Abstract. In this paper we investigate the mutual influence of participants (agents) of a social network on each other using the framework of Markov models. The main objective of this study was to check several hypotheses concerning dependencies between the influence of agents and their impact on several computational models.

Keywords: social network, dissemination of information, Markov model, influence.

1 Introduction

In this paper, we investigate the mutual influence of participants of a social network (we will call them agents in accordance with the terminology used in [1]). We used a general approach taking its roots in previous work [2]. The initial project was divided into several subprojects. The same data analysis methods were used but the input datasets were different. Work [2] used an on-line community consisting of 964 members. In this paper we use well-known methods but apply them to a different on-line community consisting of 2960 members.

Influence is understood as a process of changes in a subject caused by the behavior of other entities, their settings, intentions, views, assessments and their actions during cooperation with them [3]. Observations of psychologists show [4] that agents in a social network often do not have sufficient information for decision-making or are unable to handle available information, which causes that their decisions can be based on the decisions and/or views of other agents (social influence) [5].

Our analysis is based on data of three communities extracted from the Live Journal website. Live Journal (http://www.livejournal.com) consists of blogs, which contain sequences of messages called posts. Additionally we have available event logs, online diaries and other website content including images, multimedia, texts, etc.

The differences between a blog and a traditional diary are caused by the environment: blogs are usually public and involve third-party readers, who may enter into a public debate with the author, by commenting on blogs.

Authors of posts are called bloggers. The majority of posts are available for reading and commenting by other bloggers. Live Journal also provides an opportunity to bloggers to unite in a community, and subscribe to a community to read their blogs. In this case all the new posts in the selected blogs are displayed in a special news feed. The blogger can belong to several communities at the same time.

Information about communities, subscriptions and records themselves in most cases, are open and accessible to any Internet user. For each of the communities anyone can get the list of participants and the list of friends for each participant. Data consists of three tables: the list of communities, the list of bloggers and the list of links between bloggers. Further in this paper the terms "blogger" and "agent "will be used as synonyms. The number of entries in the list of participants who are members of one of the three communities of Live Journal are 964, 2960, 6587, and the number of links are 6359, 49504 and 190427 respectively.

2 Motivation

There are numerous works about properties of Markov models [16], which describe a social network. Traditional, analysis is mostly theoretical [1, 16], but for the successful application of the obtained theoretical results it is necessary to have well-proven algorithms for identification of the model from the observed data. There are some works where these methods are described, for example [1,16].

Despite the high effectiveness of existing computational algorithms, the main disadvantage of a Markov model is the need to build the initial matrix of influence in the infinite or a high degree. In addition there is some uncertainty in the determination of the initial matrix of mutual trust agents and their relationships with other agents.

In this paper preliminary comparison of different methods for determining the influence of agents of the social network without taking into consideration the data of the messages exchanged between the agents, was conducted. The basis for the identification of the network was data of the agents about whose blogs they read. The format of available data and the sample data is presented in table 1.

Table 1. Data fragment

The ID of the connection between agents	The ID of the reading agent	The ID of the agent, a blog which is being read	The ID of the community, to which belong both agents
1	1	2	1
2	3	2	1
3	4	1	2

In future it is worth to use messages exchanged between the agents of the social network.

3 Review of existing mathematical models

In literature, several approaches have been proposed to describe the interaction between participants in a social network: a Markov model or model of De Groot [6], a Linear Threshold Model [7], Independent Cascade Model [8], a filtering and intrusion model, Ising model, cellular automata model, etc. [16]. The models have been investigated from several perspectives: the conditions of convergence of opinions of members of the social network (see [9]), the dynamic of changes of power, the speed of convergence, the condition of the uniqueness of the final opinion (see [10]). In this work, we will use a model, described in detail in the book [1].

In some models, ranking of agents is used, for example, by means of power indices, index of Houde-Bakker [11], calculation of impact-factor of journals, ranking of web pages, PageRank algorithms, as well as the ordering of parameters "betweenness" [13] , "centrality"[14], "clustering" etc. [5,12,15].

4 Abbreviations and definitions

Because of its wide popularity, the description of Markov models in this work for the sake of brevity was not given. Details can be found, for example, in [1]. Note that transitive influence of the *i*-th agent is defined by

$$w_j = \sum_i a_{ij}^{\infty} , \tag{1}$$

where a_{ij}^{∞} is an element of the transitive closure of the matrix of direct influence, can also be computed for the original stochastic matrix of direct influence. In this case we will call it direct influence of *i*-th agent. The common method of agent identification is based on the direct influence matrix which is derived from the adjacency matrix by the formula where a_{ij} is a weight in the matrix of direct influence and b_{ij} is an element of the adjacency matrix.

$$a_{ij} = \frac{b_{ij}}{\sum_i b_{ij}} \tag{2}$$

In some cases, one can try to take into account the impact of the authority of the agent on the strength of the influence. We consider the case when the impact of authority is proportional to the number of friends of the agent, where f_j represents the credibility of the *i*-th agent.

$$a_{ij} = \frac{f_i\left(b_{ij}\right)}{\sum_i f_i\left(b_{ij}\right)} \tag{3}$$

$$f_i(x) = \left(\sum_i b_{ij} \right)^{\beta} \tag{4}$$

5 Hypotheses

1. Direct influence depends on the number of friends of an agent.
2. The number of friends is not correlated with transitive influence.
3. There is a correlation between transitive influences of agents, calculated by different methods taking into account the authority of agents.
4. The direct influence of the agent does not correlate with its transitive influence.
5. Implementation of hypotheses does not depend on the size of the network.

6 Data analysis results

Testing hypothesis 1 reveals that direct influence depends on the number of friends of an agent, and the relationship between them is close to a power-law function as shown in figure 1. The coefficient of correlation is 0.85.

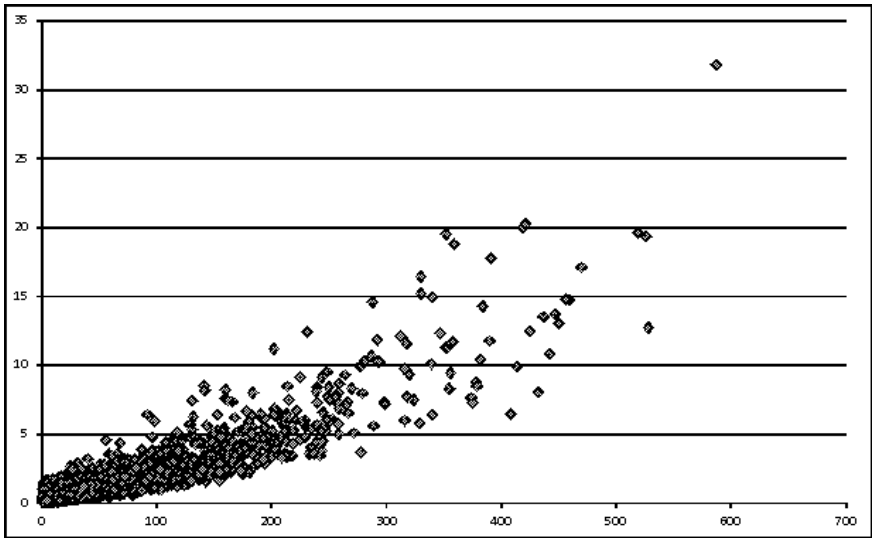


Fig. 1. The dependency between direct influence of agents (vertical) and the number of the agents' friends (horizontal).

Testing hypothesis 2 revealed that the number of friends is not correlated with transitive influence. This is shown in figure 2. The coefficient of correlation is 0.72. In addition to a linear dependence we observe the almost vertical "tail". Its presence

means that there are several agents who have a small number of friends, but a substantial influence. In particular, there are three agents for whom the transitive influence exceeds the transitive influence of the agent with the highest number of friends (whose influence can be assumed). The existence of this phenomenon has been theoretically predicted, but validation on real experimental data had not been performed yet. Note that we do not yet have an explanation for the presence of only two main lines in diagram and this issue should be investigated more thoroughly in the future.

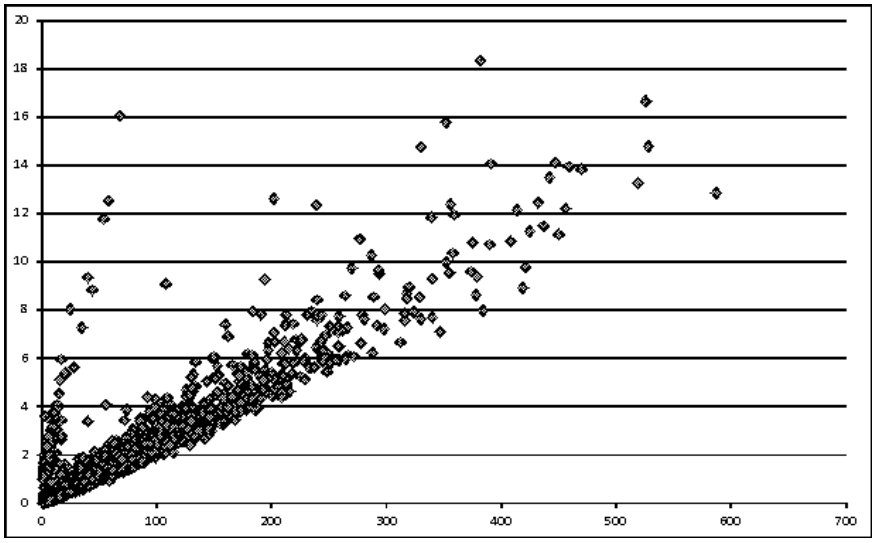


Fig. 2. The dependence of the transitive influence of agents (vertical axis) on the number of agents' friends (horizontal axis).

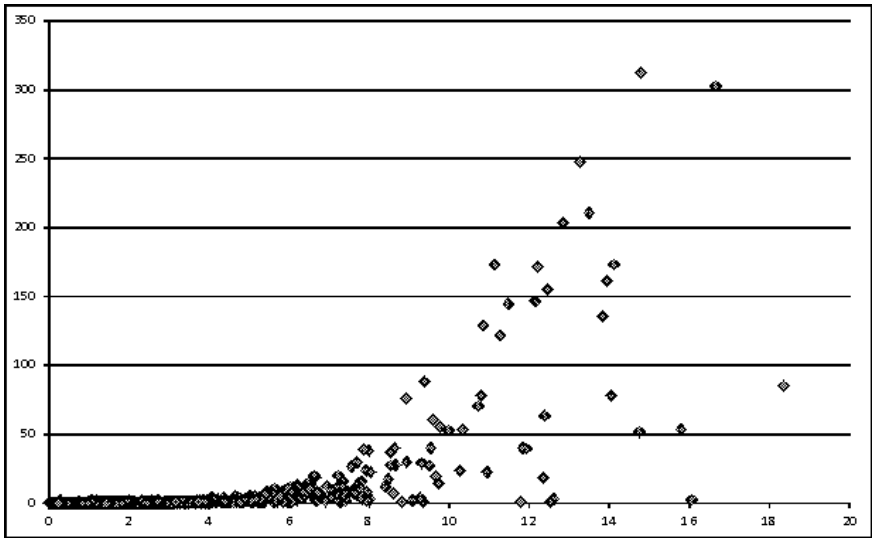


Fig. 3. The transitive influence of agents, calculated without taking into account their authority (horizontal axis) and calculated while taking authority into account (β=4) (vertical axis).

Testing hypothesis 3 revealed that there is no correlation between transitive influences of agents, neither without taking into account the authority, nor when taking authority into account. In figure 3 we see that there is no correlation. The coefficient of the correlation is 0.31. This is an important observation because by making assumptions about the impact of the number of friends of an agent on his credibility, you can get, generally speaking, different results. If we ignore some of the outlier observations, we can once again identify two "tails". The main tail shows a linear dependency, which is not equal to the constant β, and the second tail indicates a non-increasing transitive influence of the agents, despite of the increase in their transitive influence in the case of not taking into account their credibility. Moreover, the figure shows that there are a number of influential agents with low authority. This is consistent with the result that we received in the process of verification of hypothesis 2. So it can be argued that the correlation between transitive influences is complex in nature, and thus, hypothesis 3, cannot be affirmed without additional clarifications.

Testing hypothesis 4 revealed that the direct influence of an agent does not correlate with its transitive visibility. In figure 4 you can see that the linear correlation between direct and transitive influences is not clear. The coefficient of correlation is 0.78. This is also interesting, since we believe that not all agents can make decisions based on the computation of transitive influence, and therefore are forced to use direct influence measures.

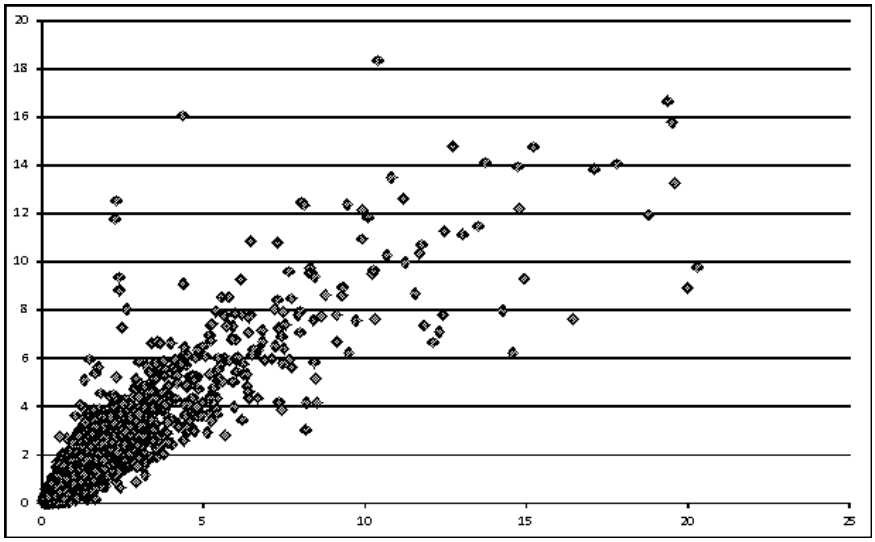


Fig. 4. The dependence of the transitive influence of the agent (vertical) from its direct influence (horizontal), which was calculated without taking into account the authority of agents.

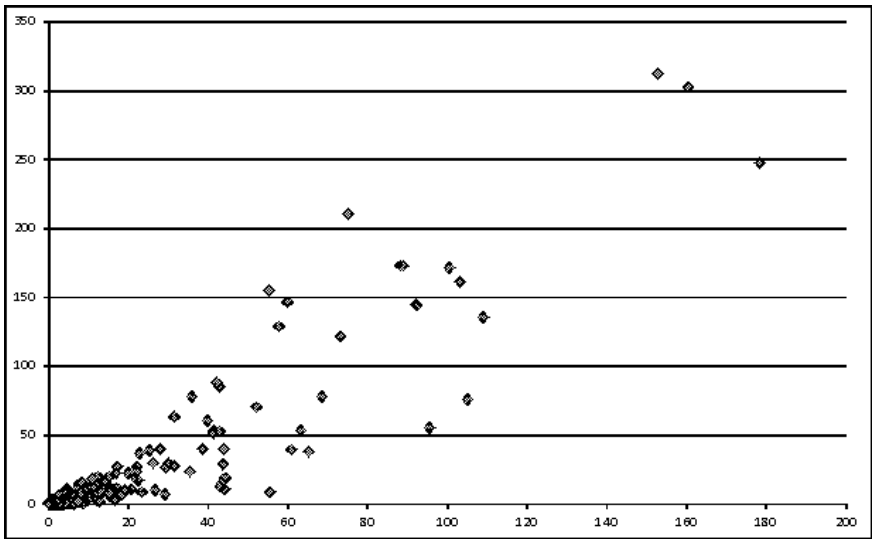


Fig. 5. The dependency of the transitive influence of the agent (vertical) from its direct influence (horizontal), calculated taking into account the authority of agents

Then we come to a rather obvious conclusion, that in real social networks such agents can be mistaken. However, we conclude that there is no ground for hypothesis 4. In

the case shown in figure 5, the linear correlation is noticeable. However, it is different for small values of direct influence than for higher values, where you can also identify the correlation. The coefficient of the correlation is 0.92.

Hypothesis 5 states the assumption that the validity of the hypotheses does not depend on the size of the network. This has not yet been verified and is a possible direction for future research.

7 Conclusions and future work

The study showed the presence of a certain number of anomalies and effects that need to be taken into account while identifying optimal Markov model parameters for experimental data. It was shown that the credibility of agents has a significant impact on the influence of agents. It was shown that there is a specific dependence between transitive influence and direct influence. We identified an abnormal cluster of agents, which have a small number of friends, but which have a great transitive influence.

It may be interesting to continue our study by verifying hypothesis 5, as well as including in the analysis the possibility of taking into account the exchange of messages between agents. We also intend to investigate the ranking of agents using methods such as alpha-centrality, the PageRank algorithm, as well as other widely used methods based on direct and transitive influences of agents.

Acknowledgement. The research is supported by the grant 10-07-00129 of Russian Foundation for Basic Research. We would like to express our gratitude to Jonas Poelmans and Dmitry Ignatov for improving the language quality.

References

1. Gubanov, D.A., Novikov, D.A., Chkhartishvili, A. G.: Social network: a model of information influence, control and confrontation. Fizmatlit, 228 p., Moscow (2010) (in Russian)
2. Fedyanin, D.N.: Application of Markov models for the analysis of influence of the participants of the Internet-community. In: Lecture Notes of the all-Russian scientific-practical conference "Analysis of Images, Networks and Texts" (AIST 2012), pp. 132-143. The national Open University "INTUIT", Yekaterinburg (2012)
3. Glossary on Control Theory and its Applications, http://glossary.ru (in Russian)
4. Deutsch, M., Gerard, H.: Study of Normative and Informational Social Influence upon Individual Judgment. In: Journal of Abnormal and Social Psychology. no.51, pp. 629-636. (1995)
5. Zuyev, A.S., Fedyanin, N.A.: Model of management of views agents in co-social networks. In: the Problems of management. № 1. pp. 37-45. ICP RAS, Moscow (2011). (in Russian)
6. DeGroot, M.H.: Reaching a Consensus. In: Journal of American Statistical Association. №69, pp.118-121 (1974)
7. Ganovetter, M.: Threshold Models of Collective Behavior. In: American Journal of Sociology, vol. 83. №6, pp.1420-1443 (1978)

8. Goldberg, J., Libai, B., Muller, E.: Talk of the Networks: A Complex Systems looks at the Underlying Process of Word-of-Mouth. In: Marketing Letters, №2, pp.11-34 (2001)
9. Berger, R.L.: Necessary and Sufficient Conditions for Reaching a Consensus using DeGroot's method. In: Journal of American Statistical Association, vol. 76, pp. 415 – 419 (1981)
10. Golub, B., Jackson, M.O.: Naive Learning in Social Networks: Convergence, Influence and Wisdom of Crowds. Technical Report 64 (2007)
11. Hoede, C., Bakker, R.: A Theory of Decisional Power. In: Journal of Mathematical Sociology, №8, pp. 309-322. (1982)
12. Rusinowska, A., Swart, H.: Generalizing and Modifying the Hoede-Bakker Index. In: Theory and Applications of Rational Structures as Knowledge Instruments. №2. Springer's Lecture Notes in Artificial Intelligence 4342, pp. 60-88. Springer (2007)
13. Freeman, L.: A set of measures of centrality based upon betweenness. In: Sociometry №40, pp. 35–41. (1977)
14. Borgatti, S, Everett, M.: A Graph-Theoretic Perspective on Centrality. In: Social Networks, 28. pp. 466–484. Elsevier (2005)
15. Wasserman, S., Faust, K.: Social Network Analysis: Methods and Applications. Cambridge: Cambridge University Press (1994)
16. Jackson, M.: Social and Economic Networks. Princeton: Princeton University Press (2008)

Analysing Online Social Network Data with Biclustering and Triclustering

Dmitry Gnatyshak[1], Dmitry I. Ignatov[1], Alexander Semenov[1], and Jonas Poelmans[1,2]

[1] National Research University Higher School of Economics, Russia
dignatov@hse.ru
http://www.hse.ru
[2] Katholieke Universiteit Leuven, Belgium

Abstract. In this paper we propose two novel methods for analysing data collected from online social networks. In particular we will do analyses on Vkontake data (Russian online social network). Using biclustering we extract groups of users with similar interests and find communities of users which belong to similar groups. With triclustering we reveal users' interests as tags and use them to describe Vkontakte groups. After this social tagging process we can recommend to a particular user relevant groups to join or new friends from interesting groups which have a similar taste. We present some preliminary results and explain how we are going to apply these methods on massive data repositories.

Keywords: Formal Concept Analysis, Biclustering and Triclustering, Online Social Networks, Web 2.0 and social computing

1 Introduction

Recently the focus of social network analysis shifted from 1-mode networks, like friend-to-friend, to 2-mode [1,2,3], 3-mode [4,5,6] and even multimodal dynamic networks [7,8,9].

This interest is not only pure academic but caused by modern business requirements. Thus, every user of a social networking website has not only friends, but he also has specific profile features, e.g. he can belong to some groups of users, indicate his tastes or books he read etc. These profile attributes are able to describe the user's tastes, preferences, attitudes, which is highly relevant for business oriented social networking web sites owners. Finding bicommunities and tricommunities can help the networking site owners to analyze large groups of their users and adjust their services according to users' needs which may in the end result in financial or other benefits.

There is a large amount of network data that can be represented as bipartite or tripartite graphs. Standard techniques like "maximal bicliques search" return a huge number of patterns (in the worst case exponential w.r.t. the input size). Therefore we need some relaxation of the biclique notion and good interestingness measures for mining biclique communities.

Applied lattice theory provides us with a notion of formal concept [10] which is the same thing as a biclique; it is widely known in the social network analysis community (see, e.g. [11,12,13,14,15,16]).

A concept-based bicluster [17] is a scalable approximation of a formal concept (biclique). The advantages of concept-based biclustering are:

1. Less number of patterns to analyze;
2. Less computational time (polynomial vs exponential);
3. Manual tuning of bicluster (community) density threshold;
4. Tolerance to missing (object, attribute) pairs.

For analyzing three-mode network data like folksonomies [18] we also proposed a triclustering technique [19]. In this paper we describe a new pseudo-triclustering technique for tagging groups of users by their common interest. This approach differs from traditional triclustering methods because it relies on the extraction of biclusters from two separate object-attribute tables. Biclusters which are similar with respect to their extents are merged by taking the intersection of the extents. The intent of the first bicluster and the intent of the second bicluster become the intent and modus respectively of the newly obtained tricluster. Our approach was empirically validated on online social network data obtained from Vkontakte (`http://vk.com`).

The remainder of the paper is organized as follows. In section 2 we describe some key notions from Formal Concept Analysis. In section 3 we introduce a model for our new pseudo-triclustering approach. In section 4 we describe a dataset which is a sample of users, their groups and interests from the Vkontakte (`http://vk.com`) social networking web site. We present the results obtained during experiments on this dataset in Section 5. Section 6 concludes our paper and describes some interesting directions for future research.

2 Basic definitions

The formal context in FCA [10] is a triple $\mathbb{K} = (G, M, I)$, where G is a *set of objects*, M is a *set of attributes*, and the relation $I \subseteq G \times M$ shows which object which attribute possesses. For any $A \subseteq G$ and $B \subseteq M$ one can define *Galois operators*:

$$\begin{aligned} A' &= \{m \in M \mid gIm \text{ for all } g \in A\}, \\ B' &= \{g \in G \mid gIm \text{ for all } m \in B\}. \end{aligned} \tag{1}$$

The operator $''$ (applying the operator $'$ twice) is a *closure operator*: it is idempotent ($A'''' = A''$), monotonous ($A \subseteq B$ implies $A'' \subseteq B''$) and extensive ($A \subseteq A''$). The set of objects $A \subseteq G$ such that $A'' = A$ is called closed. The same is for closed attributes sets, subsets of a set M. A couple (A, B) such that $A \subset G$, $B \subset M$, $A' = B$ and $B' = A$, is called *formal concept* of a context K. The sets A and B are closed and called *extent* and *intent* of a formal concept (A, B)

correspondingly. For the set of objects A the set of their common attributes A' describes the similarity of objects of the set A, and the closed set A'' is a cluster of similar objects (with the set of common attributes A'). The relation "to be a more general concept" is defined as follows: $(A, B) \geq (C, D)$ iff $A \subseteq C$. The concepts of a formal context $\mathbb{K} = (G, M, I)$ ordered by extensions inclusion form a lattice, which is called *concept lattice*. For its visualization the *line diagrams* (Hasse diagrams) can be used, i.e. cover graph of the relation "to be a more general concept". In the worst case (Boolean lattice) the number of concepts is equal to $2^{\{\min|G|,|M|\}}$, thus, for large contexts, FCA can be used only if the data is sparse. Moreover, one can use different ways of reducing the number of formal concepts (choosing concepts by their stability index or extent size). The alternative approach is a relaxation of the definition of formal concept as a maximal rectangle in an object-attribute matrix which elements belong to the incidence relation. One of such relaxations is the notion of an object-attribute bicluster [17]. If $(g, m) \in I$, then (m', g') is called object-attribute bicluster with the density $\rho(m', g') = |I \cap (m' \times g')|/(|m'| \cdot |g'|)$.

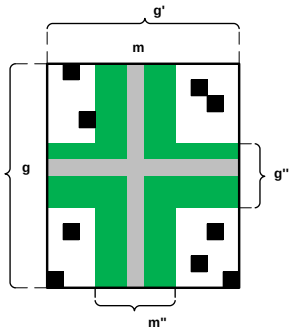


Fig. 1. OA-bicluster.

The main features of OA-biclusters are listed below:

1. For any bicluster $(A, B) \subseteq 2^G \times 2^M$ it is true that $0 \leq \rho(A, B) \leq 1$.
2. OA-bicluster (m', g') is a formal concept iff $\rho = 1$.
3. If (m', g') is a bicluster, then $(g'', g') \leq (m', m'')$.

Let $(A, B) \subseteq 2^G \times 2^M$ be a bicluster and ρ_{min} be a non-negative real number such that $0 \leq \rho_{\min} \leq 1$, then (A, B) is called *dense*, if it fits the constraint $\rho(A, B) \geq \rho_{\min}$. The above mentioned properties show that OA-biclusters differ from formal concepts by the fact that they do not necessarily have unit density. Graphically it means that not all the cells of a bicluster must be filled by a cross (see fig. 1).

3 Model and algorithm description

Let $\mathbb{K}_{UI} = (U, I, X \subseteq U \times I)$ be a formal context which describes what interest $i \in I$ a particular user $u \in U$ has. Similarly, let $\mathbb{K}_{UG} = (U, G, Y \subseteq U \times G)$ be a formal context which indicates what group $g \in G$ user $u \in U$ belongs to.

We can find dense biclusters as $(users, interesets)$ pairs in $\mathbb{K}_{UI}$ using the OA-biclustering algorithm which is described in [17]. These biclusters are groups of users who have similar interests. In the same way we can find communities of users, who belong to similar groups on the Vkontakte social network, as dense biclusters $(users, groups)$.

By means of triclustering we can also reveal users' interests as tags which describe similar Vkontakte groups. So, by doing this we can solve the task of social tagging and recommend to a particular user relevant groups to join or interests to indicate on the page or new friends from interesting groups with similar tastes to follow.

To this end we need to mine a (formal) tricontext $\mathbb{K}_{UIG} = (U, I, G, Z \subseteq U \times I \times G)$, where (u, i, g) is in Z iff $(u, i) \in X$ and $(u, g) \in Y$. A particular tricluster has a form $T_k = (i^X \cap g^Y, u^X, u^Y)$ for every $(u, g, i) \in Z$ with $\frac{|i^X \cap g^Y|}{|i^X \cup g^Y|} \geq \Theta$, where Θ is a predefined threshold between 0 and 1. We can calculate the density of T_k directly, but it takes $O(|U||I||G|)$ time in the worst case, so we prefer to define the quality of such tricluster by density of biclusters (g^Y, u^Y) and (i^X, u^X). We propose to calculate this estimator as $\widehat{\rho}(T_k) = \frac{\rho(g^Y, u^Y) + \rho(i^X, u^X)}{2}$; it's obvious that $0 \leq \widehat{\rho} \leq 1$. We have to note that the third component of a (pseudo)tricluster or triadic formal concept usually is called *modus*.

The algorithm scheme is displayed in Fig. 2

4 Data

For our experiments we collected a dataset from the Russian social networking site Vkontakte. Each entry consisted of the following fields: id, userid, gender, family status, birthdate, country, city, institute, interests, groups. This set was divided into 4 subsets based on the values of the institute field, namely students of two major technical universities and two universities focusing on humanities and sociology were considered: The Bauman Moscow State Technical University, Moscow Institute of Physics and Technology (MIPT), the Russian State University for Humanities (RSUH) and the Russian State Social University (RSSU). Then 2 formal contexts, users-interests and users-groups were created for each of these new subsets.

5 Experiments

We performed our experiments under the following setting: Intel Core i7-2600 system with 3.4 GHz and 8 GB RAM. For each of the created datasets the following experiment was conducted: first of all, two sets of biclusters using

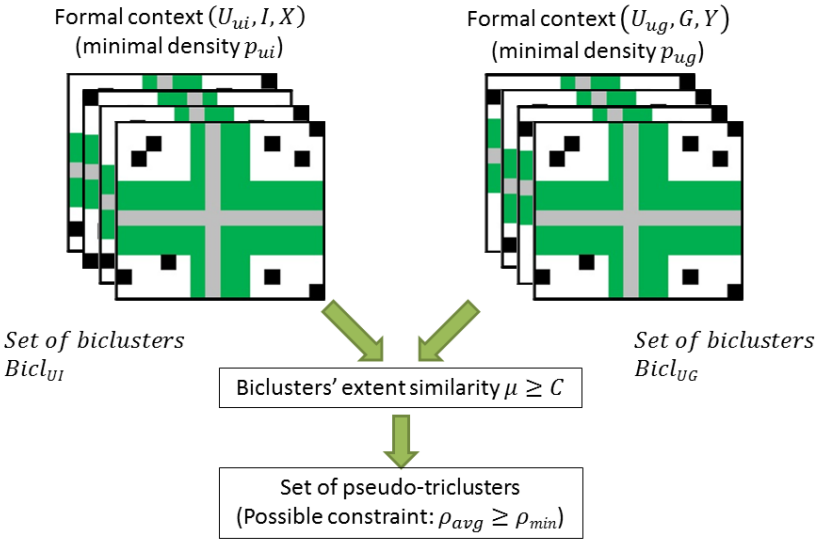


Fig. 2. Pseudo-triclustering algorithm scheme

Table 1. Basic description of four data sets of large Russian universities.

	Bauman	MIPT	RSUH	RSSU
number of users	18542	4786	10266	12281
number of interests	8118	2593	5892	3733
number of groups	153985	46312	95619	102046

various minimal density constraints were generated, one for each formal context. Then the sets fulfilling the minimal density constraint of 0.5 were chosen, each pair of their biclusters was enumerated and the pairs with sufficient extents intersection (μ) were added to the corresponding pseudo-tricluster sets. This process was repeated for various values of μ.

As it can be seen from the graphs and the tables, the majority of pseudo-triclusters had μ value of 0.3 (or, to be more precise, 0.33). In this series of experiments we didn't reveal manually any interests which are particular for certain universities, but the number of biclusters and pseudo-triclusters was relatively higher for Bauman State University. This is a direct consequence of the higher users' number and the diversity of their groups.

Some examples of obtained biclusters and triclusters with high values of density and similarity are presented below.

Example 1. Biclusters in the form $(Users, Intersts)$.

Table 2. Bicluster density distribution and elapsed time for different ρ_{min} thresholds (Bauman and MIPT universities).

ρ	Bauman				MIPT			
	UI		UG		UI		UG	
	Time, s	Number	Time, s	Number	Time, s	Number	Time, s	Number
0.0	9.188	8863	1874.458	248077	0.863	2492	109.012	46873
0.1	8.882	8331	1296.056	173786	0.827	2401	91.187	38226
0.2	8.497	6960	966.000	120075	0.780	2015	74.498	28391
0.3	8.006	5513	788.008	85227	0.761	1600	63.888	21152
0.4	7.700	4308	676.733	59179	0.705	1270	56.365	15306
0.5	7.536	3777	654.047	53877	0.668	1091	54.868	13828
0.6	7.324	2718	522.110	18586	0.670	775	44.850	5279
0.7	7.250	2409	511.711	15577	0.743	676	43.854	4399
0.8	7.217	2326	508.368	14855	0.663	654	43.526	4215
0.9	7.246	2314	507.983	14691	0.669	647	43.216	4157
1.0	7.236	2309	511.466	14654	0.669	647	43.434	4148

Table 3. Bicluster density distribution and elapsed time for different ρ_{min} thresholds (RSUH and RSSU universities).

ρ	RSUH				RSSU			
	UI		UG		UI		UG	
	Time, s	number	Time, s	number	Time, s	number	Time, s	number
0.0	3,958	5293	519.772	116882	2.588	4014	693.658	145086
0.1	3.763	4925	419.145	93219	2.450	3785	527.135	110964
0.2	3.656	4003	330.371	68709	2.369	3220	402.159	79802
0.3	3.361	3123	275.394	50650	2.284	2612	332.523	58321
0.4	3.252	2399	232.154	35434	2.184	2037	281.164	40657
0.5	3.189	2087	224.808	32578	2.179	1782	270.605	37244
0.6	3.075	1367	174.657	10877	2.159	1264	211.897	12908
0.7	3.007	1224	171.554	9171	2.084	1109	208.632	10957
0.8	3.032	1188	170.984	8742	2.121	1081	209.084	10503
0.9	2.985	1180	174.781	8649	2.096	1072	206.902	10422
1.0	3.057	1177	173.240	8635	2.086	1068	207.198	10408

Table 4. Number of similar biclusters and elapsed time for different μ thresholds (four universities).

μ	Bauman		MIPT		RSUH		RSSU	
	Time, s	Count	Time, s	Count	Time, s	Count	Time, s	Count
0.0	3353.426	230161	77.562	24852	256.801	35275	183.595	55338
0.1	76.758	10928	35.137	5969	62.736	5679	18.725	5582
0.2	80.647	8539	31.231	4908	58.695	5089	16.466	3641
0.3	77.956	6107	27.859	3770	53.789	3865	17.448	2772
0.4	60.929	31	2.060	12	9.890	14	13.585	12
0.5	66.709	24	2.327	10	9.353	14	12.776	10
0.6	57.803	22	2.147	8	11.352	14	12.268	10
0.7	68.361	18	2.333	8	10.778	12	13.819	4
0.8	70.948	18	2.256	8	9.489	12	13.725	4
0.9	65.527	18	1.942	8	10.769	12	11.705	4
1.0	65.991	18	1.971	8	10.763	12	13.263	4

- $\rho = 83,33\%$, generator pair: $\{3609, home\}$,
  bicluster: $(\{3609, 4566\}, \{family, work, home\})$
- $\rho = 83,33\%$, generator pair: $\{30568, orthodox\ church\}$,
  bicluster: $(\{25092, 30568\}, \{music, monastery, orthodox\ church\})$
- $\rho = 100\%$, generator pair: $\{4220, beauty\}$,
  bicluster: $(\{1269, 4220, 5337, 20787\}, \{love, beauty\})$

E.g., the second bicuster can be read as users 25092 and 30568 have almost all "music", "monastery", "orthodox church" as common interests. The pair generator shows which pair $(user, interest)$ was used to build a particular bicluster.

Example 2. Pseudo-triclusters in the form $(Users, Intersts, Groups)$.

Bicluster similarity $\mu = 100\%$, average density $\widehat{\rho} = 54,92\%$.

Users: $\{16313, 24835\}$,

Interests: $\{sleeping, painting, walking, tattoo, hamster, impressions\}$,

Groups: $\{365, 457, 624, \ldots, 17357688, 17365092\}$

This tricluster can be interpreted as a set of two users who have on average 55% of common interests and groups. The two corresponding biclusters have the same extents, i.e. people with almost all interests from the intent of this tricluster and people with almost all groups from the tricluster modus coincide.

6 Conclusions

The approach needs some improvements and fine tuning in order to increase the scalability and quality of the community finding process. We consider several directions for improvements: Strategies for approximate density calculation; Choosing good thresholds for n-clusters density and communities similarity; More sophisticated quality measures like recall and precision in Information

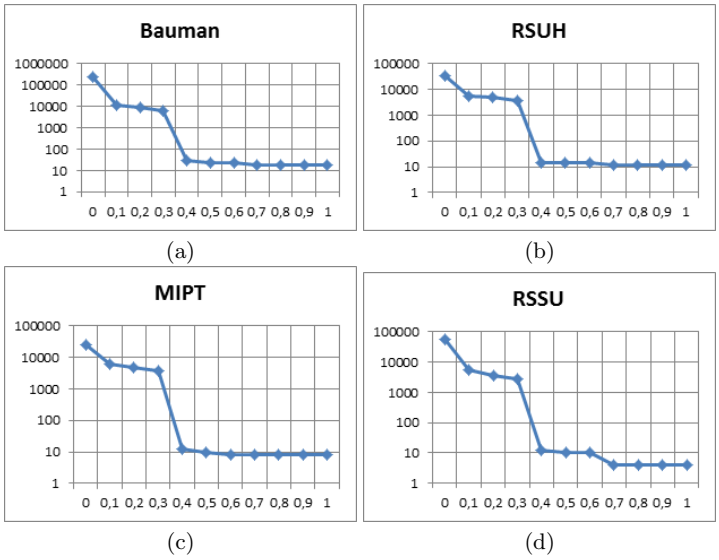


Fig. 3. Density bicluster distribution for the empirical data sets of four Russian universities. (a) Bauman State University (b) Russian State University for Humanities (c) Moscow Physical University (d) Russian State Social University

Retrieval; The proposed technique also needs comparison with other approaches like iceberg lattices ([20]), stable concepts ([21]), fault-tolerant concepts ([22]) and different n-clustering techniques from bioinformatics ([23], [24], etc.). We also claim that it is possible to obtain more dense pseudo-triclusters based on conventional formal concepts (even though it is expensive from a computational point of view). To validate the relevance of the exctracted tricommunities expert feedback (e.g., validation by sociologist) is needed.

Finally, we conclude that it is possible to use our pseudo-triclustering method for tagging groups by interests in social networking sites and finding tricommunities. E.g., if we have found a dense pseudo-trciluster ($Users$, $Groups$, $Interests$) we can mark $Groups$ by user interests from $Interests$. It also makes sense to use biclusters and triclusters for making recommendations. Missing pairs and triples seem to be good candidates to recommend the target user other potentially interesting users, groups and interests.

Acknowledgments. We would like to thank our colleagues Vincent Duquenne, Sergei Kuznetsov, Sergei Obiedkov, Camille Roth and Leonid Zhukov for their inspirational discussions, which directly or implicitly influenced this study.

References

1. Latapy, M., Magnien, C., Vecchio, N.D.: Basic notions for the analysis of large two-mode networks. Social Networks **30**(1) (2008) 31 – 48
2. Liu, X., Murata, T.: Evaluating community structure in bipartite networks. In Elmagarmid, A.K., Agrawal, D., eds.: SocialCom/PASSAT, IEEE Computer Society (2010) 576–581
3. Opsahl, T.: Triadic closure in two-mode networks: Redefining the global and local clustering coefficients. Social Networks **34** (2011) – (in press).
4. Jäschke, R., Hotho, A., Schmitz, C., Ganter, B., Stumme, G.: TRIAS–An Algorithm for Mining Iceberg Tri-Lattices. In: Proceedings of the Sixth International Conference on Data Mining. ICDM '06, Washington, DC, USA, IEEE Computer Society (2006) 907–911
5. Murata, T.: Detecting communities from tripartite networks. In Rappa, M., Jones, P., Freire, J., Chakrabarti, S., eds.: WWW, ACM (2010) 1159–1160
6. Ignatov, D.I., Kuznetsov, S.O., Magizov, R.A., Zhukov, L.E.: From Triconcepts to Triclusters. In: Proceedings of the 13th international conference on Rough sets, fuzzy sets, data mining and granular computing. RSFDGrC'11, Berlin, Heidelberg, Springer-Verlag (2011) 257–264
7. Roth, C.: Generalized preferential attachment: Towards realistic socio-semantic network models. In: ISWC 4th Intl Semantic Web Conference, Workshop on Semantic Network Analysis, Galway, Ireland,. Volume 171 of CEUR-WS Series (ISSN 1613-0073). (2005) 29–42
8. Roth, C., Cointet, J.P.: Social and semantic coevolution in knowledge networks. Social Networks **32** (2010) 16–29
9. Yavorsky, R.: Research Challenges of Dynamic Socio-Semantic Networks. In Ignatov, D., Poelmans, J., Kuznetsov, S., eds.: CEUR Workshop proceedings Vol-757, CDUD'11 - Concept Discovery in Unstructured Data. (2011) 119–122
10. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. 1st edn. Springer-Verlag New York, Inc., Secaucus, NJ, USA (1999)
11. Freeman, L.C., White, D.R.: Using galois lattices to represent network data. Sociological Methodology **23** (1993) 127–146
12. Freeman, L.C.: Cliques, galois lattices, and the structure of human social groups. Social Networks **18** (1996) 173–187
13. Duquenne, V.: Lattice analysis and the representation of handicap associations. Social Networks **18**(3) (1996) 217 – 230
14. White, D.R.: Statistical entailments and the galois lattice. Social Networks **18**(3) (1996) 201 – 215
15. Mohr, J.W., Duquenne, V.: The Duality of Culture and Practice: Poverty Relief in New York City, 1888-1917. Theory and Society, Special Double Issue on New Directions in Formalization and Historical Analysis **26**(2/3) (April-June 1997) 305–356
16. Roth, C., Obiedkov, S.A., Kourie, D.G.: Towards concise representation for taxonomies of epistemic communities. In Yahia, S.B., Nguifo, E.M., Belohlávek, R., eds.: CLA. Volume 4923 of Lecture Notes in Computer Science., Springer (2006) 240–255
17. Igantov, D.I., Kaminskaya, A.Y., Kuznetsov, S., Magizov, R.A.: Method of Biclusterzation Based on Object and Attribute Closures. In: Proc. of 8-th international Conference on Intellectualization of Information Processing (IIP 2011). Cyprus, Paphos, October 17–24, MAKS Press (2010) 140 – 143 (in Russian).

18. Vander Wal, T.: Folksonomy Coinage and Definition (2007) http://vanderwal.net/folksonomy.html (accessed on 12.03.2012).
19. Ignatov, D.I., Kuznetsov, S.O., Magizov, R.A., Zhukov, L.E.: From triconcepts to triclusters. In: Proceedings of the 13th international conference on Rough sets, fuzzy sets, data mining and granular computing. RSFDGrC'11, Berlin, Heidelberg, Springer-Verlag (2011) 257–264
20. Stumme, G., Taouil, R., Bastide, Y., Pasquier, N., Lakhal, L.: Computing iceberg concept lattices with titanic. Data & Knowledge Engineering **42**(2) (August 2002) 189–222
21. Kuznetsov, S.O.: On stability of a formal concept. Ann. Math. Artif. Intell. **49**(1-4) (2007) 101–115
22. Besson, J., Robardet, C., Boulicaut, J.F.: Mining a new fault-tolerant pattern type as an alternative to formal concept discovery. In Scharfe, H., Hitzler, P., Ohrstrom, P., eds.: Conceptual Structures: Inspiration and Application. Volume 4068 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2006) 144–157
23. Zhao, L., Zaki, M.J.: Tricluster: an effective algorithm for mining coherent clusters in 3d microarray data. In: Proceedings of the 2005 ACM SIGMOD international conference on Management of data. SIGMOD '05, New York, NY, USA, ACM (2005) 694–705
24. Mirkin, B.G., Kramarenko, A.V.: Approximate bicluster and tricluster boxes in the analysis of binary data. In: Proceedings of the 13th international conference on Rough sets, fuzzy sets, data mining and granular computing. RSFDGrC'11, Berlin, Heidelberg, Springer-Verlag (2011) 248–256

Term Weighting in Expert Search Task: Analyzing Communication Patterns*

Anna Kravchenko and Dmitry Romanov

Higher School of Economics,
Research and Educational Center of Information Management Technologies
Kirpichnaya ul. 33/4, 105679 Moscow, Russia

Abstract. The goal of the expert search task is finding knowledgeable persons within the enterprise. In this paper we focus on its distinctions from the other information retrieval tasks. We review the existing approaches and propose a new term weighting scheme which is based on analysis of communication patterns between people.
The effectiveness of the proposed approach is evaluated on a collection of e-mails from an organization of approximately 1500 people. Results show that it is possible to take into account communication structure in the process of term weighting, effectively combining communication-based and document-based approaches to expert finding.

Keywords: term weighting, expert finding, expert search task, graph analysis, referral systems

1 Introduction

The problem of locating desired information or source of knowlege has been faced by almost everyone, and sometimes finding a right person may be even more valuable than finding the right documents. In large enterprises such as companies and goverment agencies it may become a task of major importance.

For example, according to Yimam-Seid&Kobsa [14] people may search/tend to seek for an expert as a source of information for the following reasons:

- Access to non-documented information. Not all information in organizations can possibly be explicitly documented or information may be deliberately not made publicly available for economic, social and political reasons.
- Specification need. Userw may be unable to formulate a plan to solve a problem, or to pick a direction of research and resorts to seeking experts.

* This work was conducted with financial support from the Government of the Russian Federation (Russian Ministry of Science and Education) under contract 13.G25.31.0096 on "Creating high-tech production of cross-platform systems for processing unstructured information based on open source software to improve management innovation in companies in modern Russia".

- Leveraging on others expertise (group efficiency). E.g. finding a piece of information that a relevant expert would know/find with less effort than the seeker or filtering reliable information from the huge mass of information available.
- Interpretation need. Userw may need help understanding the information even if he/she manages to find it from documents.
- Socialization need. Users may prefer the human dimension that is involved in asking an expert as opposed to interacting with documents and computers.

It may be also important to identify users or groups of users who are more knowlegeable than the others. In a big enterprise it may help discover new ideas, the most valuable employees and sources of innovations.

An expert search system is designed to help with these tasks. In contrast with classical information retrieval where documents are retrieved, given a query, the system has to return a ranked list of person names in response.

Most approaches to expert finding tend to copy standard information retrieval methods, such as Vector Space Model [15], representing user profiles as documents. We argue that it may not be the best way, since the need to search for experts mostly emerges in business environments (like corporate mail, for example), which implies a completely different data organization. Authors are usually connected to each other and data mostly consists of messages, not isolated documents.

Existing methods can be divided into communication-based, that take into account the communication patterns between users, and content-based, which focus strictly on the documents content. An essential notion in content-based methods is term significance. It is used to estimate the generality or specificity of a term and evaluate how important is a word to a document, collection or corpus. The process of calculating term significance is called term weighting and the most common approach to term weighting is the tf-idf measure and it's variations [8].

The problem of existing term weighting schemes as well as other content-based methods is that they don't take into account communication patterns between users. They may be combined with communication-based approaches sequentially, but there is no existing method that allows communication structure to affect the term's weight. Developing such method could be a significant contribution to the domain.

In this paper we analyse the specifics of the task, review the existing approaches to expert finding and term weighting and propose a new weighting scheme developed specifically for expertise evaluation.

2 Task specifics

As it has been mentioned earlier, the expert search task has different properties from the standart information retrieval task. The standard IR task was developed for analysing web pages and text collections such as online libraries. An expert

search task needs to deal with forums, social networks, blogs and corporate e-mails.

We consider the following properties of the expert search task the most important:

- In business environment size of the available corpora is limited by the company lifespan, therefore number of documents is relatively small. Same counts for social networks.
- Corpora may contain information that normal text collections don't, like authorship or communication patterns.
- Communications between authors are not the same as hyperlinks. The possibility to relate every document to its author creates a hierarchical structure, which is absent in the standard IR task.

Naturally, different data organization demands alternative approaches to its processing, that take into account additional features and information they provide. Existing but rare approaches to standart IR such as time marks may also prove useful. Further investigation of the impact of some of this information to extracting data from business corpora is required.

3 Existing Approaches

3.1 Content-based schemes

The most direct approach to expert finding is adapting Vector Space Model for the task. VSM, one of the most successful solutions to IR, has proved to be highly effective for many types of documents. VSM represents queries and documents as vectors in an n-dimensional information space. Then it systematically compares each document vector with the query vector to find the documents nearest to the query. To adapt VSM for referral systems, each person is represented with a vector based on a "document" consisting of all messages they sent to the user. A person vector contains weights for each term in the query based on how frequently the term was used by the sender in his or her email messages. An example of this approach has been described in [15].

Let P_i be the person vector for colleague p_i. This vector contains the computed weights for all terms in the sender's email that also occurs in the query vector. Adaptation of TF-IDF metrics and variations are the most common for calculating the weights.

$$w_i k = \frac{tf_{ik} \log(N/n_k)}{\sqrt{\sum_{t=1}^{n} [(tf_{ik}) \log(N/n_k]^2}} \ . \tag{1}$$

Here, N = number of colleagues of the user, n_k = number of colleagues who have used term k, tf_{ik} = term frequency; number of times colleague p_i has used term k.

It can be separated into the term frequency factor (tf) and the inverse document frequency factor (idf, collection factor).

The first gives a measure of the importance of the term t within the particular context, document or user profile. The simplest case is the occurrence count of a term in a document. The more a term is encountered in a certain context, the more it contributes to the meaning of the context

Inverse document frequency factor diminishes the weight of terms that occur very frequently in the collection and increases the weight of terms that occur rarely. It is a measure of the general importance of the term (obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient).

Variations of this scheme include term relevance weight (i.e. probabilistic term inverse frequency) and inverse term frequency, a good overview can be found in [8].

The calculation of the query vector, Q_j , can be taken directly from information retrieval methods with computed weights for each term in the query vector $< w_{j1}, w_{j2}, ...w_{jn} >$. Several different query vectors has been proposed, for example, by Salton and Buckley [12]:

$$w_{ik} = [0.5 + \frac{0.5tf_{ik}}{tf_{max}}] \log N/n_k \ . \tag{2}$$

Here tf_{max} = maximum frequency of all terms in query j.

After person vectors have been calculated for each sender, the similarity between the query vector and a person vector can be calculated as the cosine of the angle between those vectors. Using this sort of similarity measure, the agent will return to the user a list of potential experts ranked in decreasing order of computed similarity values upon a query. The term weights can be normalized, so the similarity is between 0 and 1. It is convenient to think of this value as the likelihood or con dence that a person will be able to answer the query of the user.

3.2 Communication-based schemes

The ranking scheme and it's variations described above don't take into account link structure or communication patterns. A rather more popular approach for expert finding are schemes that focus specifically on those properties, which has proven to be more effective [2]. An overview of graph-based algorithms is given in [4].

One of the most famous is PageRank. In this approach the obtained ranking is the principal eigenvector of the adjacency matrix of the digraph in which edges correspond to links (messages) between users. This approach turns out to be virtually idenctical to the well-known PageRank algorithm for ranking web pages, the only difference being that in PageRank low-weight edges are added between all nodes in both directions.

The simplest is the successor model, where messages are viewed as directed edges pointed from greater to lesser expertise. All the people 'downstream' from a given node are considered of lesser expertise. This one measure of expertise

is simply the count of such people (nodes) Another is positional power function (PPF), where the ranks$(ri|i = 1..n)$ satisfy the following system of equations:

$$r_i = \sum_{j \in S_i} \frac{1}{n}(r_j + 1) \ . \tag{3}$$

An adaptation of HITS algorithm, a precursor to PageRank was also proposed in [2].

Neither of those pays attention to term weighting, focusing only on communication patterns.

4 Proposed approach

The goal of our work was is to investigate the impact of communication pattern analysis to expertise evaluation in business corpora and the possibility of using this information during term weighting.

In case of expert finding the main task of term weighting is to divide generic terms from professional vocabulary. Selecting rare terms doesn't always work, as well as the TF-IDF scheme, since rare terms may be generic as well. However, significance of the term may correlate with the patterns of communicaion between people who use it. Generic terms tend to come up randomly in documents of any domain created by authors of all kinds of occupation and level of professional expertise, while professional terminology is mostly used between experts and people asking for a consultation, and experts are normally few and tend to interact with each other. Those people should form a tight cluster, while generic term users would be scattered around the enterprise with relatively small amount of connections.

The main idea of our approach is to calculate the number of clusters that people form for every term in the corpora and to compare it to the number of clusters that would be formed if the links between users were distributed randomly.

The proposed method was analyzed on a collection of e-mails from an organization of approximately 1500 people consisting of 1270000 messages.

5 Algorithm realisation

For every term, information about the number of occurrences, number of people using the term and messages id was included in the index. For every message senders and receivers were known.

For every $n = 3..50$ all terms that were used by n people was selected. Then for every term the number of clusters k its users were forming was computed, and finally for every pair (n, k) the number of terms was measured.

Then a random distribution of messages has been modelled.

For every $n = 3..50$ 200 terms were randomly selected from the table, then, according to number of term users n_1 and number of term encounters n_2, n_1

people were randomly selected from the list of users and liked ("sending messages") to n_2 any other users on the list. Uniform distribution was used for the probability of creating link with another user. As a result an alternative index was created, then numbers of all (n, k) pairs from this index were computed according to the algorithm described above.

6 Results

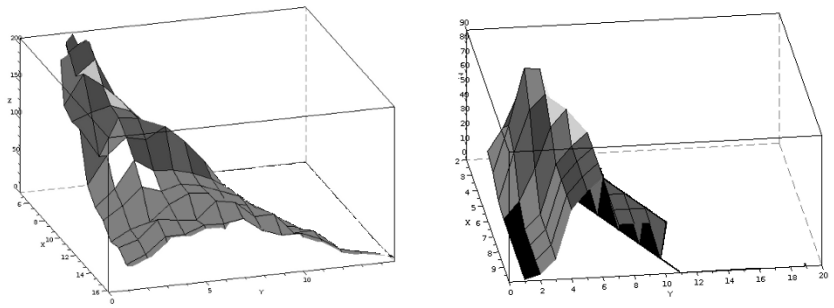


Fig. 1. Distribution pattern for the factual communication structure (on the left) and the randomly generated one (on the right).

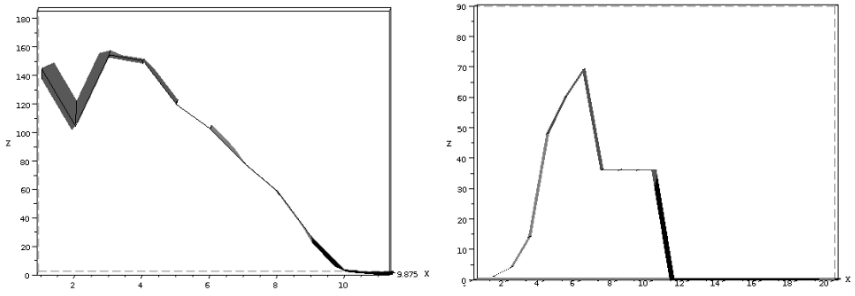


Fig. 2. Distribution for n=10. The graph on the right shows randomly generated communications

Fig. 1 and Fig. 2 depict the results of the experiment.

As it can be seen, random distribution differs prominently from the factual one.

The random distribution pictures show that the number of terms, users of which form a single cluster, is vanishingly small. It is highly improbable for such terms to occur by chance.

At the same time, pictures of the factual distribution contain a much higher number of "single cluster" terms, and the distribution graph itself is flatter. Therefore it can be concluded that those terms do not appear at random and may share some common properties.

It also appeared that probability of term being professional instead of generic corresponds with the number of clusters. Terms with smaller k values tend to be more significant.

Taking all these things into account, we find it reasonable to assume that number of clusters can be used as an important indicator of term significance.

This leads to a following weighting scheme:

$$w_{ij} = \frac{p}{p_r} \ . \tag{4}$$

$p = \frac{N_{nk}}{N_n}$, $p_r = \frac{N_{nk,r}}{N_{n,r}}$, where p is the probability of n users forming k clusters for factual distributuion, p_r the probability of it for random distributuion, N_{nk} and $N_{nk,r}$ are the number of terms, users of which form k clusters, in factual and random distribution accordingly, N_n and $N_{n,r}$ are the total number of terms that are used by n people.

If $w_{ij} > 1$ term is considered a professional one and if $w_{ij} < 1$ the term is considered general.

A simpler, though less accurate scheme is

$$w_{ij} = \frac{k}{k_{mostprobable}} \ . \tag{5}$$

Here if the number of users is n and they form k clusters, $k_{mostprobable}$ is the value that has the highest number of (n, k) pairs in random distribution.

Those schemes can be used by itself or in a combination with TF-IDF as an additional factor.

Precise evaluation is currently unachievable due to the small amount of annotated data available - there are no corpora focusing on the term level. The direction of our future work is building a full expert search system based on this scheme. It will allow to evaluate the approach using standart expert search corpora, annotated on text level, such as, for example, the TREC 2005 task.

7 Conclusion

In this paper we have shown that it is possible to make analysis of communication patterns a part of a term weighting scheme, and have suggested an example of such scheme. We have also focused on the distinctions between expert finding and standard IR task.

The proposed approach allows to take into account the specifics of the expert search task. It also allows to adapt the existing approaches (such as vector space models, for example) for expert finding more effectively.

For further investigation, we are focusing on proper evaluation of the scheme, combining it with other term weighting schemes. and improving it by exploring the impact of other properties like graph density.

References

1. Balog, K., De Rijke, M.: Determining Expert Profiles (with an Application to Expert Finding). In IJCAI07: Proc. 20th Intern. Joint Conf. on Artificial Intelligence, pp. 2657–2662 (2007)
2. Campbell, C.S., Maglio, P.P., Cozzi, A., Dom, B.: Expertise Identification Using Email Communications. In Proceedings of the Twelfth International Conference on Information and Knowledge Management, pp. 528–531 (2003)
3. Chen, J., Geyer, W., Dugan, C., Muller, M., Guy, I.: Make New Friends, but Keep the Old: Recommending People on Social Networking Sites. In Proceedings of the 27th International Conference on Human Factors in Computing Systems, pp. 201–210 (2009)
4. Dom B., Eiron I., Cozzi A., Zhang Y.: Graph-Based Ranking Algorithms for E-mail Expertise Analysis. In proceedings of 8th AGM SIGMOD Workshop on Research issues in Data Mining and Knowledge Discovery (2003)
5. Feldman, R., M. Fresko, Y. Kinar, Y. Lindell, O. Liphstat, M. Rajman, Y. Schler, and O. Zamir: Text Mining at the Term Level. Principles of Data Mining and Knowledge Discovery: pp. 6573. (1998)
6. Golbeck, J., and J. Hendler: Reputation Network Analysis for Email Filtering. In Proceedings of the First Conference on Email and Anti-Spam, 44:5458. (2004)
7. Kolari, P., Finin, T., Lyons, K., Yesha, Y.:Expert Search Using Internal Corporate Blogs. In Workshop on Future Challenges in Expertise Retrieval, SIGIR 2008, pp. 2–5. (2008)
8. Lan, M., Sung, S.Y., Low, H.B., Tan. C.L.: A Comparative Study on Term Weighting Schemes for Text Categorization. (2005)
9. Macdonald, C., Ounis, I.:Voting for Candidates: Adapting Data Fusion Techniques for an Expert Search Task. In Proceedings of the 15th ACM International Conference on Information and Knowledge Management, pp. 387–396. (2006)
10. Mesnage, CS, Carman, M.: Piloted Search and Recommendation with Social Tag Cloud-based Navigation. In 1st Workshop On Music Recommendation And Discovery (WOMRAD), ACM RecSys, 2010, Barcelona, Spain. (2010)
11. Petkova, D., Croft, W.B.: Hierarchical Language Models for Expert Finding in Enterprise Corpora. In Tools with Artificial Intelligence, 2006. ICTAI06. 18th IEEE International Conference On, pp. 599608. (2006)
12. Salton. G., Buckley C.: Term Weighting Approaches in Automatic Text Retrieval Gerard Salton. Proceedings. 2005 IEEE International Joint Conference (2005)
13. Serdyukov, P., Rode, H., Hiemstra, D.: Modeling Relevance Propagation for the Expert Search Task. Technical report, TREC 2007 Enterprise Track (2007)Dom
14. Yimam, S., Kobsa, A.: Expert Finding Systems for Organizations: Problem and Domain Analysis and the DEMOIR Approach. (2003)
15. Yu, B., Singh, M.P.: A Multiagent Referral System for Expertise Location. In Working Notes of the AAAI Workshop on Intelligent Information Systems, pp. 66–69 (1999)

Semantic Matching Using Concept Lattice

Ana Meštrović

University of Rijeka, Department of Informatics,
Omladinska 14, 51000 Rijeka, Croatia
amestrovic@inf.uniri.hr

Abstract. This paper describes how a concept lattice that represents semantic relations (synonymy, hyponymy, hypernymy) in a set of words can be used for semantic matching. This kind of concept lattice is the result of the formal concept analysis technique used for determining semantic relations in a set of words extracted from a monolingual dictionary. It is shown how relations between concepts can be mapped into semantic matching relations (equivalence, disjointness, more specific, less specific). The results of using semantic matching with concept lattice in a spoken dialog manager system are shown. The models are represented in F-logic language and implemented in FLORA-2 system.

Keywords: Concept Lattice, Semantic Matching, Synonym Extraction, F-logic, Natural Language Dialog System

1 Introduction

This paper describes how formal concept analysis can be applied in the domain of natural language processing. An approach of semantic analysis using concept lattice as background knowledge is proposed. An important problem of semantic analysis is semantic heterogeneity that includes managing the diversity in knowledge. Therefore a process of semantic matching is defined. Semantic matching in this paper denotes an operation of matching two lexical units that have equal or similar meaning. Semantic matching may have a more general definition, such as matching operation that takes two graph-like structures (e.g., conceptual hierarchies, database schemas or ontologies) and produces mappings among the nodes of two graphs that correspond semantically to each other [8]. However, the main idea presented in this paper is one of semantic matching as an operation that matches different concepts that are semantically close.

Formal concept analysis is a technique that includes lattices and order theory as a tool for data analysis. It is based on sound mathematical theory, introduced to information science by Ganter and Wille [6,7]. The idea of using FCA in the domain of natural language processing has been already discussed in [5,3]. Further, the idea of using concept lattices for semantic relations capturing and in linguistic applications is presented in [12,14,18,19,20,21,17]. In [21] it is described how WordNet can be formalised using FCA. A similar approach has been used in [14] for capturing semantic relations between words given in a Croatian

monolingual dictionary using formal concept analysis in FLORA-2 system. We presented how semantic relations that are automatically extracted from a dictionary can be formalised and visualised using concept lattice. These results can be further used for semantic analysis in the dialog system. In this work, a deductive object-oriented logic programming language named F-logic, for semantic analysis is used. F-logic [11] provides a natural way of defining a conceptual model of data semantics and Web data manipulation. Further, F-logic is a formalism that can capture formal concept analysis. All rules for semantic analysis are defined in F-logic and implemented in FLORA-2 system. It is shown how proposed semantic analysis with a concept lattice can be used as a part of the spoken dialog system for weather information in Croatia.

The second section of this paper introduces semantic analysis and semantic matching process. The third section presents how semantic relations defined in the monolingual dictionary can be represented using concept lattice. Section four presents how concept lattice can be implemented as a part of a spoken dialog system for the Croatian language. Finally, some possible improvements are discussed and some future work plans are presented.

2 Semantic analysis and semantic matching

Semantic analysis can be viewed as a task of translating a natural language sentences into a formal meaning representation language [1]. An important issue of semantic analysis and natural language understanding in general is how to treat semantically close words or phrases. Therefore a natural language understanding module needs to have additional knowledge about semantic relations between words. In this paper, an elementary process of linking semantically close words (concepts) is called semantic matching.

For two words w_1 and w_2 there are three possible relations that describe cases of semantic closeness: equivalence ($\equiv$), more specific ($\sqsubseteq$), partial overlapping ($\sqcap$). If two words have no semantic closeness then they are disjoint ($\perp$). Additionally, a relation less specific ($\sqsupseteq$) can be introduced as an inverse relation of more specific. For every two words w_1 and w_2 there is only one possible relation and it depends on how close these words are in meaning. Therefore, it is possible to define a function that connects two words by assigning a semantic relation that holds between them.

Definition 1. *Let W be the set of all possible words and let R be the set of relations, $R = \{\equiv, \sqsubseteq, \sqsupseteq, \sqcap, \perp\}$. Mapping f_{rel} from $W \times W$ to R assignes to each pair of words $(w_1, w_2) \in W \times W$ an approprate relation from R that holds between* w_1 and w_2.

There are also measures defined for measuring semantic distance [16]. This is not considered in this paper, but eventually function f_{rel} can be extended with a measure of closeness and that is a plan for future work. Furthermore, semantic matching may be described with semantic correspondences called mappings attached to one of the following semantic relations: disjointness, equivalence, more

specific, less specific, overlapping [8]. Although semantic matching has a broader definition as described in the introduction, the basic task of semantic matching in the context of semantic analysis is to connect the word (phrase) with a set of words (phrases) that are semantically close to it. For that purpose another mapping is defined.

Definition 2. *Let W and R be the sets as in the previous definition. Mapping f_{match} from $W \times R$ to a partitive set of W, $\mathcal{P}(W)$ assigns to each pair (w, r) where $w \in W$ and $r \in R$ a set of words W_m that for each word w_m from W_m holds $f_{rel}(w, w_m) = r$.*

Using mapping f_{match} it is possible to link each word with a set of words that are semantically close into it. Therefore, mapping f_{match} may be used to accomplish an operation of semantic matching. In the next section it is shown how all relations between words, that are a result of semantic matching operation, can be presented and visualised using concept lattice. Furthermore, a concept lattice that describes semantic relations can be used for semantic analysis.

3 Monolingual dictionary formalization using formal concept analysis

3.1 From a monolingual dictionary to a concept lattice

In this section an approach for dictionary formalization using the formal concept analysis (FCA) technique is presented. A similar approach may be defined by using WordNet as a resource of semantically close words and their relationships. There are many research projects that deal with WordNet and some of them use formal concept analysis as a technique for exploiting semantic relations, visualization and other research[9,12,21]. In this research a monolingual dictionary is used instead of WordNet. One reason is that WordNet for Croatian language is not completely finished and available. Eventually, there is more information that can be extracted from a dictionary than from WordNet. Moreover, the described approach can be used to define or update Croatian WordNet because FCA based semantic matching gives synsets as the final result.

Data in dictionaries are usually presented with implicitly defined structure. Important attributes of a word are organized following implicit dictionary structure. Each word may have more than one meaning. Each meaning has its own description with its own set of attributes described. This complex way of representing a word defines a dictionary structure that can be captured using formal grammar. A monolingual Croatian dictionary [2] is used for automatic extraction of semantically close words. The formal grammar is defined in F-logic in order to capture this structure. After formalizing the structure of a dictionary, the final goal was to extract words that are semantically close.

Set of semantically close words are further analyzed using FCA technique. Different semantic relations can be found in dictionaries (synonyms, near synonyms, hypernym, hyponym, etc.). These relations form a hierarchical structure

between semantically close words. In [14] it is shown how that can be modeled as a formal context and viewed as a concept lattice.

If a word in a dictionary has more than one meaning and if any of these meaning is descriptive (not explained with another synonym), than a set of special marks is introduced (as zd, $zd1$, $zd2$,...). These marks are appearing in the formal context and concept lattice, denoting that there are more meanings for some word but with no synonyms in a dictionary.

Figure 1 shows three possible relations between two words (w_1, w_2) that have semantic overlapping. In the first case (a) word w_1 can replace word w_2 in any context and vice versa. In the second case (b) word w_2 can be replaced with word w_1 in any context, but not vice versa since word w_1 has a more general meaning. In the third case (c) both words w_1 and w_2 have some additional meaning and therefore can be replaced with each other only in certain contexts. These cases actually reflect the relationships between two synonyms or hyperonym and hyponym described as relations of equivalence, less specific and more specific respectively .

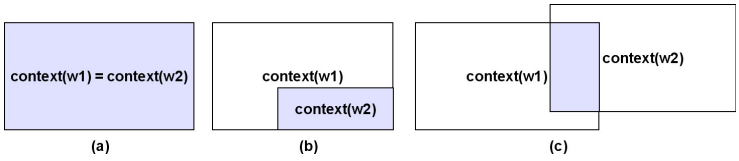


Fig. 1. Cases of semantic overlapping

The idea was to define formal context in a way that concept lattice as a result shows a naturally established hierarchy in a set of words. Firstly, the general case is analysed in order to present the basic idea of the proposed approach. Let $W_n = w_1, w_2, ..., w_n$ be a set of n words, then a formal context $C = (O, M, I)$ may be defined for set W_n. Set O is a set of words from W_n, $(O = W_n)$ and set M is a set of words that overlap with words from W_n. In some generalized cases set of attributes can be a subset to a set of objects, but here it is $M = O$. Rules for transforming dictionary data into formal concepts are defined in F-logic as it is shown in [14]. The transformation process assumes adding relations of reflexivity, symmetry and eventual transitivity. A final model should also capture a possibility of a different representation of meanings. Apart from this, incompleteness and irregularities can also appear in a dictionary and thus, have to be included into a model. Therefore, many additional rules for handling these specific situations are formed.

The final result of FCA technique applied to a dictionary is a concept lattice of words. The described model of a concept lattice reflects semantic closeness between words, therefore, it may be called a semantic relation concept lattice. On the higher levels of the lattice there are words with more general meaning and on the lower levels there are words with more specific meaning. These relations

naturally correspond to relations defined within the semantic matching operator. The interpretation of a concept lattice is given in the next section.

3.2 Concept lattice interpretation

For the purpose of this paper, a small set of words is translated into English. A set of words with similar relations between them is chosen in order to show the main idea of semantic closeness representation using concept lattice. However, there may be slight differences in lattice interpretation using English language from which we have using Croatian language in the original example. An example of formal context is described for a given set of words $W_1 = \{infinite, endless, prominent, noted, enormous, strong, huge, well-known, eminent, big, high, remarkable\}$. All these words translated into Croatian language have semantic similarity detected in the Croatian monolingual dictionary. This particular set of words has been chosen as an example because it generates a rich concept lattice structure. Other sets of words with semantic similarity from Croatian monolingual dictionary contains a smaller number of words. Some smaller set of words are presented in the next section where an application of concapt lattice-based semantic analysis for the weather forecast domain is shown.

Using the rules for defining concepts shortly presented in a previous section, a set of 19 concepts is generated. Relations between concepts defined by a conceptual lattice are shown in Fig. 2.

Using the proposed technique, it is possible to define a concept lattice for any set of words that have semantic similarities. Each concept connects words from a dictionary in a way that concept extent corresponds to the words given in an intent set. For example, concept $k14$ with extent defined as $k14[extent \rightarrow [infinite, endless]]$ links words $infinite$ and $endless$ in a way that these two words have semantic overlapping with the same set of words given as the intent of the same concept, $k14[intent \rightarrow [infinite, endless, enormous, huge, big, zd1]]$. Moreover, these are the only two words that have semantic overlapping with this exact set of words represented as the concept intent. Hierarchical relationships defined between concepts in the concept lattice reflect the possible relations between words that can be attached using function f_{rel} defined in the second section. The hierarchy is defined in the way that the words belonging to extents of concepts of a higher level are less specific ($\sqsupseteq$) than the words which belong to lower level concepts. For example, in Croatian language the word *big* that is an extent of a concept $k10$ has less specific meaning than the word *huge* that belongs to an extent of concept $k2$, that is a subconcept of a concept $k10$. In further analysis of the concept $k10$ it can be noticed that an extent of the concept $k10$ is only the word *big*. Hierarchy of a lattice provides information that the word *big* semantically links all words from set W_1. For some words in a lattice the word *big* includes all their meaning and for other words it includes only partial meaning. For example, the word *big* can replace words *prominent*, *remarkable* in every context, but cannot replace the words *endless*, *infinite* in every context. In the set of words that is given in this example there is no word that represents generalization of all the words because the concept $k1$ is an

empty concept. The first lower level has five different concepts ($k14$, $k17$, $k10$, $k19$ and $k7$). These concepts can not be compared. The extents of these five concepts deal with the words that have a more general meaning.

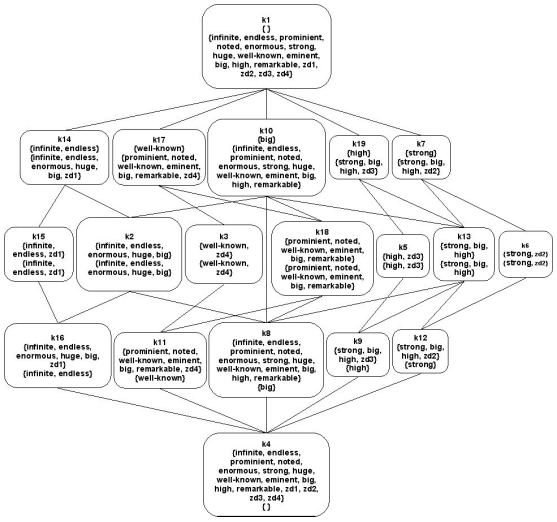


Fig. 2. A concept lattice for set of words W_1

Concept lattice can be presented in another way, as it is shown in Fig. 3. This kind of representation is called reduced lattice representation [4]. In the reduced lattice, each word is shown only once. The extent of a concept is formed by collecting all objects that can be reached by edges that connect that concept with concepts on a higher level. It is important that in reduced representation some relations between sets of words are visualized in a clearer way. Besides, this kind of representation shows only necessary words. In a reduced lattice, each word appears only once in the exact position that shows where it belongs in a word hierarchy. It is expressed that words *enormous* and *huge* are synonyms and that can be described as an equivalence relation. Set of words S=$\{prominent$, $noted$, $eminent$, $remarkable\}$ is also a set of synonyms with some exceptions. Words $well-known$ and big are more general than the words in set S. This means it is possible to replace every word from a set S with word $well-known$ or big in every context with no change in meaning. This reflects a relation more specific (less specific, on the contrary) between each word from W_1 and the word $well-known$. Henceforth the words $well-known$ and big semantically overlap, but have separate meaning nevertheless. It refers to an overlapping relation.

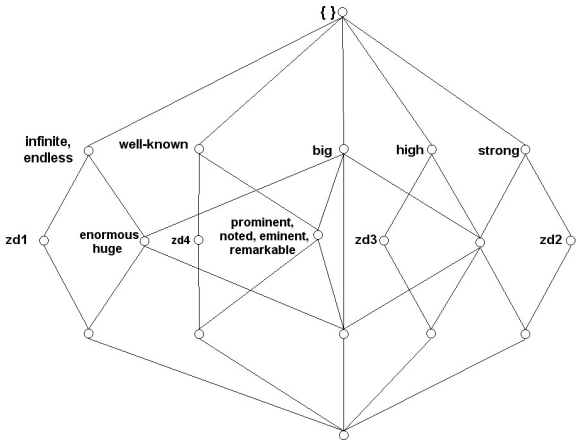


Fig. 3. A reduced concept lattice for set of words W_1

4 An application of semantic relation concept lattice

One possible application of semantic relation concept lattice is to use it as background knowledge in the process of semantic analysis. Semantic analysis is a fundamental part of the natural language understanding module. In this section an example of semantic matching using semantic relation concept lattice in the Croatian language dialog system for the weather forecast domain is described. The current status of a Croatian spoken dialog system prototype is presented in [13].

There are different approaches defined for semantic analysis that can be implemented in the natural language understanding module [10]: syntax-driven semantic analysis, semantic analysis based on formal grammar and information extraction. The Croatian weather data semantic analysis combines information extraction slot filling technique with grammar [15]. This combined approach is chosen mainly because of the limited weather domain and highly flective nature of the Croatian language. Information extraction is used with limited domain and when no detailed comprehention is needed. In information extraction process, knowledge can be described with simple templates. Templates consist of frames with slots that need to be filled with data from the text. In those situations only relevant information from the input text is used for filling the slots and the rest of the text is ignored. Information extraction with the slot filling technique is used in many semantic parsers of spoken dialog systems [22].

The slot filling technique is focused on predefined keywords and matching tempaltes. The problem of such a key word-based matching interpretation are words with semantic similarities. In the proposed approach knowledge of semantic similarities is automatically extracted from a dictionary and stored in the concept lattice. Therefore, for each word w it is possible to use f_{match} function

to get all words that have semantic similarities (descibed relations from previously introduced set R) with the word w. One simple example of using f_{match} function to interpret the question from the weather forecast dialog system is shown in Table 3. Information about wind and weather is given as a small part of the weather forecast for the Adriatic coast. Wind names are specific for the Adriatic coast and therefore are not translated into English (note that *jugo* and *široko* are synonyms for the same wind and *bura* is a wind name, also). The result of applying concept lattice for matching words from the question, more precise answers are obtained.

Table 1. An example of question answering with and without using concept lattice (CL)

Example of question	Answer with no CL	f_{match} mapping from the CL	Answer using CL
An example of weather forecast data that is used for answering questions: *Jugo* is going to blow today with occasional precipitation on the Adriatic coast. Tomorrow, *bura* is a possibility.			
Does *široko* blow on the Adriatic coast?	No	$f_{match}(široko, \equiv) = jugo$	Yes
Is there precipitation on the Adriatic coast?	No	$f_{match}(rain, \sqsubseteq) = precipitation$	Yes
Is wind going to blow tomorrow?	No	$f_{match}(wind, \sqsupseteq) = bura$	Yes

5 Conclusion

In this paper an approach of concept lattice-based semantic matching is introduced. The main motivation of this research was to improve the process of natural language understanding in the previously developed Croatian language dialog system for the weather forecast domain. At first, the idea of how to use semantic match operators in the process of semantic analysis is introduced. Secondly, the idea of using FCA technique for representing semantic relationships between words in a dictionary is presented. Three different models of semantic relationships are modeled as three different formal contexts using F-logic. This way, a set of words can be represented using formal context designed in order to represent semantic hierarchy between words. The final result is a concept lattice that shows a semantic overlapping between words. Implementation and results are presented in the fourth section. The conceptual model is represented in F-logic and after that implemented in FLORA-2 system.

The final result is more precise semantic analysis and more precise answer generation with concept lattice for semantic matching (Table 1). This hypothesis

has been proved for a set of examples, but no evaluation for the whole system has been done yet. The evaluation of concept lattice-based semantic matching is a topic for further research.

References

1. Allen, J.: Natural language understanding. Benjamin/Cummings series in computer science, Benjamin/Cummings Pub. Co. (1995), http://books.google.hr/books?id=l4lQAAAAMAAJ
2. Anić, V.: Rječnik hrvatskoga jezika. Novi Liber (2005)
3. Boutari, A.M., Carpineto, C., Nicolussi, R.: Evaluating term concept association measures for short text expansion: two case studies of classification and clustering. In: Proceedings of the Seventh International Conference on Concept Lattices and their Applications (CLA 2010) (2010)
4. Carpineto, C., Romano, G.: Concept data analysis: theory and applications. Wiley (2004), http://books.google.hr/books?id=-F8OoVXQioAC
5. Falk, I., Gardent, C.: Combining formal concept analysis and translation to assign frames and thematic grids to french verbs. In: Amedeo Napoli, V.V. (ed.) International Conference on Concept Lattices and Their Applications CLA 2011. pp. 223–228. INRIA Nancy - Grand Est and LORIA (2011)
6. Ganter, B., Wille, R.: Formal concept analysis: mathematical foundations. Springer (1999), http://books.google.hr/books?id=cN1QAAAAMAAJ
7. Ganter, B., Wille, R.: Applied lattice theory: Formal concept analysis. In: G. Gatzer editor, B. (ed.) General Lattice Theory (1997)
8. Giunchiglia, F., Shvaiko, P., Yatskevich, M., Giunchiglia, F., Shvaiko, P., Yatskevich, M.: S-match: an algorithm and an implementation of semantic matching. In: In Proceedings of ESWS. pp. 61–75 (2004)
9. Hotho, A., Staab, S., Stumme, G.: Explaining text clustering results using semantic structures. In: In Principles of Data Mining and Knowledge Discovery, 7th European Conference, PKDD 2003. pp. 217–228. Springer-Verlag (2003)
10. Jurafsky, D., Martin, J.: Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition. Prentice Hall series in artificial intelligence, Pearson Prentice Hall (2009), http://books.google.hr/books?id=fZmj5UNK8AQC
11. Kifer, M., Lausen, G., Wu, J.: Logical foundations of object-oriented and frame-based languages. Journal of the ACM 42, 741–843 (1995)
12. Martin, B., Eklund, P.: Applying formal concept analysis to semantic file systems leveraging wordnet. In: Proceedings of the 10th Australasian Document Computing Symposium (2005)
13. Meštrović, A., Bernić, L., Pobar, M., Martinčić-Ipšić, S., Ipšić, I.: A croatian weather domain spoken dialog system prototype. CIT. Journal of computing and information technology 18, 309–316 (2010)
14. Meštrović, A., Čubrilo, M.: Monolingual dictionary semantic capturing using concept lattice. International Review on Computers and Software (I.RE.CO.S.) 6, 173–184 (2011)
15. Meštrović, A., Martinčić-Ipšić, S., Čubrilo, M.: Weather forecast data semantic analysis in f-logic. Journal of Information and Organizational Sciences 31, 115–129 (2007)

16. Mohammad, S., Gurevych, I., Hirst, G., Zesch, T.: Cross-lingual distributional profiles of concepts for measuring semantic distance. In: Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (2007)
17. Old, J.: Homograph disambiguation using formal concept analysis. In: Missaoui, R., Schmid, J. (eds.) 4th International Conference on Formal Concept Analysis, Lecture Notes in Computer Science. vol. 3874, pp. 221–232. Springer-Verlag (2006)
18. Potemkin, S.: Concept lattice implementation in semantic structuring of adjectives. In: Ignatov, D., Kuznetsov, S., Poelmans, J. (eds.) Concept Discovery in Unstructured Data. pp. 63–70 (2011)
19. Priss, U.: Linguistic applications of formal concept analysis. In: Wille, G.S. (ed.) Formal Concept Analysis, Foundations and Applications. pp. 149–160. Springer Verlag (2005)
20. Priss, U., Old, L.J.: Modelling lexical databases with formal concept analysis. Journal of Universal Computer Science, Vol 10, 967–984 (2004)
21. Priss, U.E.: The formalization of wordnet by methods of relational concept analysis. In: WordNet: An Electronic Lexical Database and Some of its Applications. pp. 179–196. MIT Press (1998)
22. Seneff, S., Hurley, E., Lau, R., Pao, C., Schmid, P., Zue, V.: Galaxy-ii: A reference architecture for conversational system development. In: in Proc. ICSLP. pp. 931–934 (1998)

Self-Tuning Semantic Image Segmentation

Sergey Milyaev[1,2], Olga Barinova[2]

[1] Voronezh State University
sergey.milyaev@gmail.com
[2] Lomonosov Moscow State University
obarinova@graphics.cs.msu.su

Abstract. In this paper we present a method for finding optimal parameters of graph Laplacian-based semantic segmentation. This method is fully unsupervised and provides parameters individually for each image. In the experiments on Graz dataset the accuracy of segmentation obtained with the parameters provided by our method is very close to the accuracy of segmentation obtained with the parameters chosen on the test set.

1 Introduction

Methods based on graph Laplacian (L2-norm regularization) have shown state-of-the-art results for interactive image segmentation [1] and image matting [2]. In [1] Grady suggested explanation of using Laplacians for interactive segmentation in terms of random walks. In [3] the use of graph Laplacian for interactive image segmentation was explained in terms of transductive inference. The parameters of graph Laplacian are usually chosen by validation on hold-out dataset. However, the optimal values of parameters can vary significantly from one image to another, therefore choosing the parameters individually for each image is desirable.

In this paper we consider the task of finding optimal parameters of graph Laplacian for semantic image segmentation. We propose a new method that tunes the parameters individually for each test image without using any ground truth segmentation. The idea of our method is based on the properties of graph Laplacian to approximate the Laplace-Beltrami operator studied in [4]. Proposed self-tuning method is computationally efficient and achieves performance comparable to choosing the parameters on the test set.

The remainder of the paper if organized as follows. In section 2 we describe the image segmentation framework used in this paper. In section 3 we present our method for unsupervised learning of graph Laplacian parameters. In section 4 we present the experimental evaluation of the proposed method.

2 Semantic segmentation framework

Let us denote $W : W_{ij} = \exp\left(-d\left(\mathbf{x}_i, \mathbf{x}_j\right)\right)^2\Big)$ - a weight matrix with Gaussian kernel. Let $g_i = \sum_j w_{ij}$ stand for a sum of W along the i-th row. Let D be

a diagonal matrix with values g_i on diagonal. Graph Laplacian is defined as a matrix $L = W - D$.

The methods for image segmentation and matting solve the following energy function with respect to vector $\mathbf{f} = (f_1, ..., f_N)$:

$$E(\mathbf{f}) = \sum_i c_i \left(f_i - y_i\right)^2 + \sum_{i,j} w_{ij} \left(f_i - f_j\right)^2 . \tag{1}$$

In the matrix form (1) takes the following form:

$$E(\mathbf{f}) = \left(\mathbf{f} - \mathbf{y}\right)^T C \left(\mathbf{f} - \mathbf{y}\right) + \mathbf{f}^T L \mathbf{f} , \tag{2}$$

where C denotes a square diagonal matrix with c_i on diagonal and $\mathbf{y}$ denotes an N-dimensional vector of initial likelihood scores y_i. This optimization problem reduces to solving a sparse linear system:

$$(L + C)\mathbf{f} = C\mathbf{y}. \tag{3}$$

The object/background segmentation algorithm then consists in: 1) computing graph Laplacian matrix L; 2) solving the sparse linear system (3); 3) thresholding the output. We assume that initial estimates y_i and confidences c_i are provided by local models (e.g. appearance model of a specific category).

This framework can be extended to a multi-class segmentation. Let K denote the number of labels corresponding to object categories. If we solve (3) for each label l vs all other labels $1, \cdots, l-1, l+1, \cdots, K$ and obtain the values $y_i^{(l)}$ for all image pixels; at the end, an i-th image pixel is assigned to the label l_{max}, where $l_{max} = arg \max_{l=1,\cdots,K} y_i^{(l)}$.

3 Self-tuning method

Suppose that the distance function d is represented as a weighted sum of metrics $d_i : \mathbb{R} \times \mathbb{R} \to R^+; i = 1, ..., K$:

$$d(\mathbf{x}_i, \mathbf{x}_j)^2 = \frac{1}{\epsilon} \sum_{k=1}^{K} \alpha_k d_k(\mathbf{x}_i, \mathbf{x}_j)^2 , \tag{4}$$

with fixed $\alpha_1 = 1$. Therefore the parameters of graph Laplacian $\alpha_i, i = 2, ..., l$ are the weights of features $\mathbf{x}^k, i = 2, ..., l$ and the kernel bandwidth ϵ. Below we show that optimal value of ϵ is determined by the values of $\alpha_i, i = 2, ..., l$.

Choosing the kernel bandwidth ϵ with fixed α. We start by fixing the parameters $\alpha_i, i = 2, ..., l$. As shown in [5], if we assume that L provides a good approximation of Laplace-Belrami operator then the following condition holds:

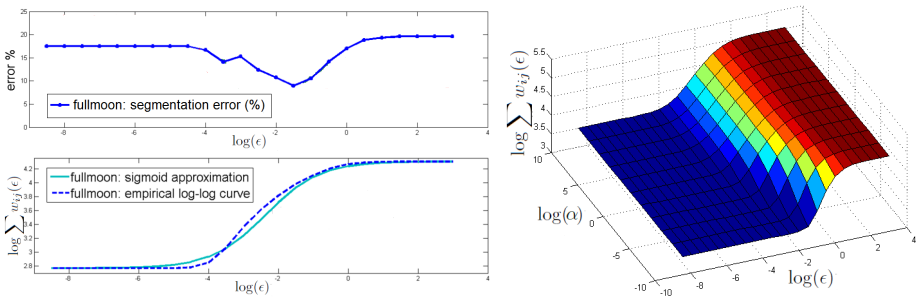
$$\log \sum_{i,j} w_{ij}(\epsilon) \approx m/2 \log(\epsilon) + \log \left(\frac{N^2 (2\pi)^{m/2}}{vol(M)} \right) , \tag{5}$$

where m is a dimensionality of corresponding manifold M and w_{ij} are the elements of the weight matrix W.

Consider the *logarithmic plot* of $\log \sum_{i,j} w_{ij}$ with respect to $\log \epsilon$. Figure (3) shows the plot of $\log \sum_{ij} w_{ij}$ with respect to $\log \epsilon$ and $\log \alpha$ for one image from GrabCut dataset. According to (5) if the approximation is good then the slope of this plot ϵ should be about the half dimensionality of corresponding manifold.

In the limit $\epsilon \to \infty$, $w_{ij} \to 1$, so $\sum_{ij} w_{ij} \to N^2$. On the other hand, as $\epsilon \to 0$, $w_{ij} \to \delta_{ij}$, so $\sum_{ij} w_{ij} \to N$. These two limiting values set two asymptotes of the plot and assert that logarithmic plot cannot be linear for all values of ϵ.

Therefore in order to get better approximation of Laplace-Beltrami operator with $\alpha_1, ..., \alpha_K$ fixed we have to choose the value of ϵ from the linear region of logarithmic plot. We use the point of maximum derivative as the point of maximum linearity.



(a) (b)

Fig. 1. (a) - *Top:* segmentation errors for the "fullmoon" image from GrabCut database with respect to $\log \epsilon$ (α is fixed). *Bottom:* Dashed line - logarithmic plot for the "fullmoon" image with respect to $\log \epsilon$ (α is fixed). The optimal value of ϵ is chosen in the point of maximum derivative of the logarithmic plot; Solid line - sigmoid fit of the logarithmic plot. (b) - The plot of $\log \sum_{ij} w_{ij}$ with respect to $\log \epsilon$ and $\log \alpha$. Th plot shown in shown in (3, bottom) corresponds to the 2-d slice of this 3-d plot for fixed α. Note that the slope of linear region are not constant for all values of α. We seek for α such that the slope in the linear region equals 0.5.

Unsupervised learning of $\alpha_1, ..., \alpha_K$ *and* ϵ. As follows from (5) the slope of the logarithmic curve near optimal value of ϵ has to be close to $m/2$, where m is the dimensionality of manifold M. In our case $m = 1$, therefore the slope of the logarithmic plot has to be 0.5. If the plot has different slope in the linear region, this indicates that the second term in (5) is large.

In order to find optimal values of $\alpha_2, ..., \alpha_K$ we solve the following optimization problem:

$$(\alpha_2^{(opt)}, ..., \alpha_K^{(opt)}) = arg \min_{\alpha_2,...,\alpha_K} \|S(\alpha_2, ..., \alpha_K) - 0.5\|, \tag{6}$$

where $S(\alpha_2, ..., \alpha_K)$ is the slope of the logarithmic plot in the point of maximum derivative.

$S(\alpha_2, ..., \alpha_K)$ can be estimated numerically. We can compute $\log \sum_{ij} w_{ij}$ for different values of ϵ and estimate the slope of this function in the point of maximum derivative. Therefore the optimization problem (6) can be solved using standard optimization methods, e.g. Nelder-Mead simplex method.

The unsupervised learning method for graph Laplacian therefore has two steps:

- Find $\alpha_2, ..., \alpha_K$ by solving optimization problem (6)
- Find ϵ with $\alpha_2, ..., \alpha_K$ as the point of maximum derivative of the logarithmic plot.

Implementation details For the experiments in this work we use the distance function from [3]:

$$\tilde{d}^2(\mathbf{x}_i, \mathbf{x}_j) = \frac{\|r_i - r_j\|^2}{\sigma_r^2} + \frac{\|x_i - x_j\|^2}{\sigma_g^2}, \tag{7}$$

where r encodes mean RGB color in the superpixel, x encodes coordinates of the center of the superpixel, $\sigma_r > 0$ and $\sigma_g > 0$ are the parameters of the method. The meaning $\sigma_r > 0$ and $\sigma_g > 0$ is the scale of chromatic neighbourhoods and the scale of geometric neighbourhoods respectively.

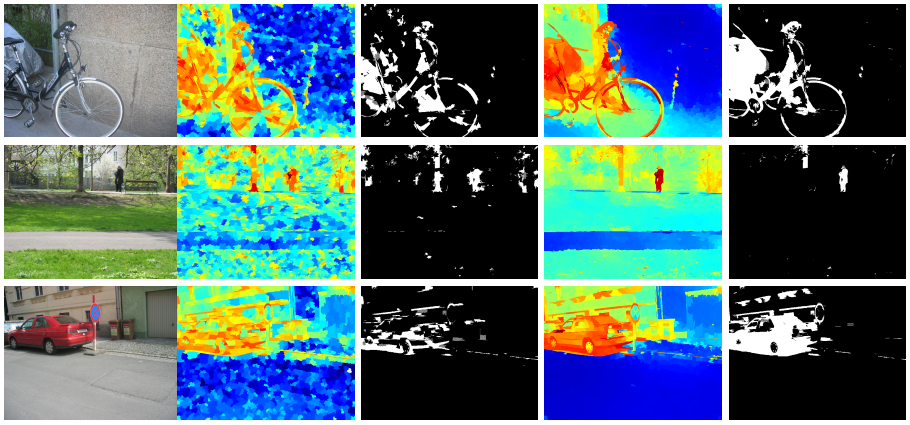
This distance function (7) can be rewritten in the form of (4) as follows:

$$\tilde{d}^2(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{\epsilon}\left(\|r_i - r_j\|^2 + \alpha \|x_i - x_j\|^2\right), \tag{8}$$

where $\epsilon = 0.5\sigma_r^2$ and $\alpha = \sigma_r^2/\sigma_g^2$. Therefore, the distance function has two parameters ϵ and α.

In the second step of the learning method (3) we use the sigmoid fit of the logarithmic plot. The shape of logarithmic plot can be approximated with a sigmoid function:$T(x) = \frac{A}{B+\exp(Cx+D)} + E$. Since the asymptotes of the sigmoid are set by (5) and the slope in the linear region of the sigmoid should be 0.5 the sigmoid has only one free parameter that controls the shift of the sigmoid along horizontal axis. Figure (3) illustrates the choice of ϵ according to sigmoid approximation.

In most cases the slope of the logarithmic plot $S(\alpha)$ is monotonic function of α. Monotonicity of $S(\alpha)$ allows using simple bin-search for optimization problem (6).



(a) *input image* (b) *local model* (c) *thresholded (b)* (d) *Laplacian* (e) *thresholded (d)*

Fig. 2. Results of SVM and graph Laplacian method for images from Graz dataset. (a) - input images of "bike", "person" and "cars" classes; (b) - real-valued output from local SVM model, color ranges from blue to red and encodes the real-valued output; (c) - results of thresholding the SVM outputs; (d) - real-valued output of graph Laplacian using SVM as a local model with the parameters learnt by our method, color ranges from blue to red and encodes the real-valued output; (e) - thresholded output of our method. Note how graph Laplacian refines the output from SVM. It doesn't oversmooth the result and preserves fine details like the wheel of the bike and the small figure of the person.

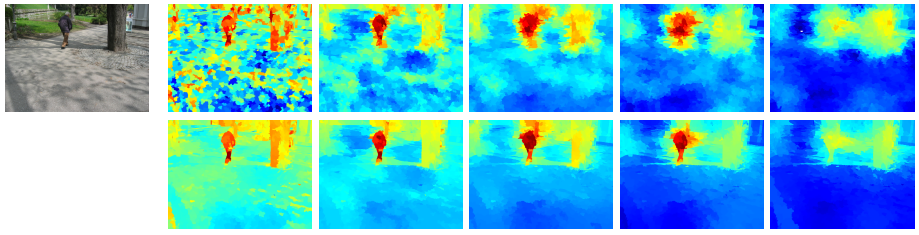
4 Experiments

In all experiments graph Laplacian operated with superpixels produced by image over-segmentation methods. Each superpixel was linked with a fixed number of it's nearest neighbours, and the distances to other superpixels were assumed infinite. For all experiments we used confidences that are a linear function of the outputs of local appearance models $c_i = 0.5(1 - |p_i - 0.5|)$.

Graz dataset [1] contains 1096 images of three classes: "person", "bike" and "car". In our experiments we solved a separate binary segmentation problem for each category. To measure the quality of segmentation we used a standard metric - percent of incorrectly classified pixels in the image.

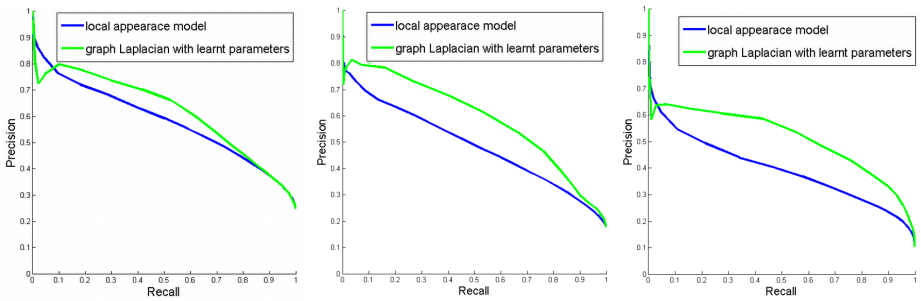
In our experiments we used an open-source VlBlocks toolbox [2], which implements the method described in [6]. We chose it for comparison for the following reasons. First, it allows using different local appearance models. The method has a parameter N meaning number of neighbouring superpixels which features are used for classification of each particular superpixel. So we report performance metrics for different values of N to illustrate the performance of proposed graph Laplacian framework applied to different local models. Second, the toolbox in-

[1] available at `http://www.emt.tugraz.at`
[2] code available at `http://vlblocks.org/index.html`



(a) *input image* (b) *N=0* (c) *N=1* (d) *N=2* (e) *N=3* (f) *N=4*

Fig. 3. Results of using different local models. The first row shows real-valued output of local appearance models. The color ranges from blue to red and encodes the real-valued output from the segmentation framework. The second row shows results of our method. Parameter N sets the size of superpixel neighborhood in the local model. The effect of using graph Laplacian is better visible for smaller N.



(a) *"bike" class* (b) *"person" class* (c) *"car" class*

Fig. 4. Precision-recall curves for "bike", "person" and "car" classes of Graz dataset. *Blue curves* - local appearance model (N=0); *Green curves* - graph Laplacian with learnt parameters.

cludes implementation of discrete CRF with graph-cut inference, which we use for comparison. Note, this CRF model uses similar types of features (color and spatial coordinates of superpixels) to those used in our graph Laplacian.

In our experiments on GrabCut dataset we used the same over-segmentation and the same local appearance model based on SVM as [6]. To obtain initial estimates y_i for graph Laplacian framework we scaled SVM outputs to $[0, 1]$ interval for each image.

In the first experiment the parameters ϵ and α were validated on the GrabCut dataset. In the second experiment we validated the parameters on the test set. In the third experiment we used our unsupervised learning method for choosing the parameters individually for each image. We also compared with Vlblocks implementation of CRF with graph-cut inference. The strategy for choosing internal parameters of CRF was the same as in [6].

Table 1 contains results of the comparison. Our unsupervised learning gives results comparable to upper bound on performance of graph Laplacian with

	N=0			N=1			N=2			N=3			N=4		
	cars	bike	pers	cars	bike	pers	cars	bike	pers	cars	bike	pers	cars	bike	pers
SVM	41.9	56.5	49.4	59.6	66.9	63.6	68.0	69.2	66.6	69.4	70.7	65.2	66.5	71.9	63.6
GraphCut	43.0	57.7	49.3	60.2	67.1	63.9	70.1	70.2	66.9	70.7	71.0	65.4	68.8	72.2	64.2
Ours (valid.GrabCut)	50.0	60.1	56.0	65.5	68.7	68.5	71.6	70.8	70.8	72.2	72.0	69.5	70.0	**73.2**	67.3
Ours (valid.testset)	**56.6**	**63.3**	**59.1**	66.3	68.4	68.8	71.9	70.4	70.4	72.6	71.2	69.4	70.8	72.2	68.0
Ours (learnt)	54.2	60.9	58.5	65.1	66.8	**69.4**	**72.0**	69.5	**71.3**	**73.3**	70.3	**70.2**	**71.4**	71.5	**68.9**

Table 1. Performance on Graz dataset at equal precision and recall rates for "cars", "bike" and "person" classes. First row: local appearance model (from VlBlocks toolbox). Second row: result of applying discrete CRF with graph cut inference (from VlBlocks toolbox). Third row: graph Laplacian with parameters validated on GrabCut dataset. Fourth row: graph Laplacian with parameters validated on the test set. Fifth row: graph Laplacian with parameters learnt individually for each image. For each appearance model used in our experiments (we varied the number of neighboring regions as in [6]) the best result is shown in **bold font**. Underlined are the best overall results.

fixed parameters from the second experiment. The value of performance gain compared to local appearance model differs for different values of parameter N. The smaller N is the smaller neighborhood is considered by low-level model, and the more significant is the gain in performance attained by both CRF and graph Laplacian.

The gain in performance of graph Laplacian is almost uniformly higher than the performance gain obtained by discrete CRF. Figure 2 shows results provided by local appearance model (SVM) and corresponding results of using graph Laplacian with learnt parameters. Figure 3 shows how the results vary for different local models.

The running time is the following: learning phase takes about 0.2 seconds on average, solving of linear system 3 takes about 0.02 seconds on average.

5 Conclusion

We presented a method for tuning internal parameters of graph Laplacian in a fully unsupervised manner individually for each test image. Proposed method has a low computational cost and shows better performance compared to discrete CRF with graph-cut inference. In the future work we plan to use more complex distance functions and investigate the case then distance function has more parameters.

References

1. Grady, L.: Random walks for image segmentation. IEEE Trans. on Pattern Analysis and Machine Intelligence **28**(11) (2006) 1768–1783
2. Levin, A., Lischinski, D., Weiss, Y.: A closed form solution to natural image matting. IEEE Trans. on Pattern Analysis and Machine Intelligence (2008)
3. Duchenne, O., Audibert, J.Y., Keriven, R., Ponce, J., Segonne, F.: Segmentation by transduction. In: CVPR. (2008)

4. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computataion **15** (2003) 1373–1396
5. Coifman, R.R., Shkolnisky, Y., Sigworth, F.J., Singer, A.: Graph laplacian tomography from unknown random projections. IEEE Trans. on Image Processing
6. Fulkerson, B., Vedaldi, A., Soatto, S.: Class segmentation and object localization with superpixel neighborhoods. In: ICCV. (2009)

A Neural Network-Like Combinatorial Data Structure

Xenia A. Naidenova

Military Medical Academy, Saint-Petersburg, Russian Federation

ksennaidd@gmail.com

Abstract. A generalization of algorithm is proposed for implementing the well-known effective inductive method of constructing sets of cardinality (q+1) ((q+1)-sets) from their subsets of cardinality q ((q)-sets). A new neural network-like combinatorial data-knowledge structure supporting this algorithm is advanced. This structure can drastically increase the efficiency of inferring functional and implicative dependencies as like as association rules from a given dataset.

Keywords: Level-wise algorithm, Inferring dependencies from a dataset, Neural network-like data structure, Knowledge representation.

1. Introduction

Mining association rules from databases has attracted great interest because of its potentially very useful application. Mining association rules was firstly introduced by Agrawal et al. in [1], where an algorithm, called AIS (for Agrawal, Imielinski, and Swami), has been proposed. Another algorithm, called SETM (for Set Oriented Mining), has been introduced in [2]. This algorithm uses relational operations in a relational database environment.

The next step in solving the problem of inferring association rules has been done in [3-4], where the algorithms Apriori, AprioriTid, and AprioriHybrid have been presented. These algorithms use an effective inductive method of constructing sets of cardinality (q+1) ((q+1)-sets) from their subsets of cardinality q ((q)-sets). A (q+1)-set can be constructed if and only if there exist all its proper (q)-subsets. The same principle underlies the algorithm Titanic for generating key patterns [5] and the algorithm TANE for discovering functional and approximate dependencies [6]. The level-wise inductive method of (q+1)-sets' construction has also been proposed for inferring good diagnostic (classification) tests for a given classification or class of objects [7-9]. These tests serve as a basis for extracting functional dependences, implications, and association rules from a given dataset.

In all enumerated problems, the same algorithm deals with different sets of elements (items, attributes, values of attributes, transactions, indices of itemsets) and checks the different properties of generated subsets. These properties are "to be a frequent (large) itemset", "to be a key pattern", "to be a test for a given set of examples", "to be an irredundant set of attribute values", "to be a good test for a given set

of examples", and some others. If an obtained subset does not possess a required property, then it is deleted from consideration. This deletion reduces drastically the number of subsets to be built at all greater levels. In section 2, we introduce a Background algorithm solving the task of inferring all maximal subsets of set *S* (i.e., such subsets that cannot be extended) possessing a given PROPERTY. The set *S* can be interpreted depending on the context of a considered problem. This algorithm implements the level-wise inductive method of (q+1)-sets' construction. In section 3, we consider some possible ways of increasing the efficiency of Background Algorithm. Finally, in Section 4, we propose a neural network-like combinatorial data structure for constructing (q+1)-sets from their q-subsets.

2. Background algorithm

The Background Algorithm. By $s_q = (i_1, i_2, \ldots, i_q)$, we denote a subset of *S*, containing q element of *S*. Let *S*(test-q) be the set of subsets $s = \{i_1, i_2, ..., i_q\}$, q = 1, 2, ..., nt, satisfying the PROPERTY. Here nt denotes the cardinality of *S*. We use an inductive rule for constructing $\{i_1, i_2, ..., i_{q+1}\}$ from $\{i_1, i_2, ..., i_q\}$, q = 1, 2, ..., nt-1. This rule relies on the following consideration: if the set $\{i_1, i_2, ..., i_{q+1}\}$ possesses the PROPERTY, then all its proper subsets must possess this PROPERTY too and, consequently, they must be in *S*(test-q). Thus the set $\{i_1, i_2, ..., i_{q+1}\}$ can be constructed if and only if *S*(test-q) contains all its proper subsets. Having constructed the set $s_{q+1} = \{i_1, i_2, ..., i_{q+1}\}$, we have to determine whether it possesses the PROPERTY or not. If not, s_{q+1} is deleted, otherwise s_{q+1} is inserted in *S*(test-(q+1)). The algorithm is over when it is impossible to construct any element for *S*(test-(q+1)).

Background Algorithm. Inferring all maximal (not extended) subsets of *S* possessing a given PROPERTY.

1. Input: $q = 1$, $S = \{1,2,\ldots, nt\}$, S(test-q) = {{1}, {2}, ..., {nt}}.

Output: the set *SMAX* of all maximal subsets of S possessing the PROPERTY.

2. S_q := S(test-q);

3. While $||S_q|| \geq q + 1$ do

3.1 Generating S (q + 1) = $\{s = \{i_1, \ldots, i_{(q+1)}\}: (\forall\, j)\ (1 \leq j \leq q + 1)\ (i_1, \ldots, i_{(j-1)}, i_{(j+1)}, \ldots, i_{(q+1)}\} \in S_q\}$;

3.2 Generating S(test-(q + 1)) = $\{s = \{i_1, \ldots, i_{(q+1)}\}: (s \in S(q + 1))$ & (PROPERTY(s)) = *true*$)\}$;

3.3 S(test-q) := $\{s = \{i_1, \ldots, i_q\}: (s \in S(\text{test-}q))$ & $((\forall\, s')(s' \in S(\text{test-}(q + 1))\ s \not\subset s')\}$;

```
3.4. q := q + 1;

3.5. max := q;

end while

4. TGOOD := ∅;

5. While q ≤ max do SMAX := SMAX ∪ {s: s = {i1, ..., is}
∈ S(test-q) };

5.1 q:= q + 1;

end while

end
```

The most important factor influencing on computational complexity of Background Algorithm is the method of inductive generating of q-sets in the level-wise manner. Generally, we use the following inductive rules, where SN is the family of sets S_q of cardinality equal to q, $q = 1, \ldots, nt$, $S_q \subseteq S = \{1, \ldots, nt\}$ and $C_S(q)$ denotes the number of combinations of S on q.

(1) $q = 1$, $q + 1 = 2$;
$s_q = \{i\}$, $s_{(q+1)} = \{i, j\}$, $(\forall j)$ $(i \neq j, \{j\} \in SN$;

(2) $q = 2$, $q + 1 = 3$;
$s_q = \{i, j\}$, $s_{(q+1)} = \{i, j, l\}$, where l different from i, j and such that there are in SN
a) two sets $s_1 = \{i, l\}$, $s_2 = \{j, l\}$ or
b) $s = \{l\}$;

(3) $q = 3$, $q + 1 = 4$;
$s_q = \{i, j, m\}$, $s_{(q+1)} = \{i, j, m, l\}$, where l different from i, j, m and such that there are in SN
a) three sets $s_1 = \{i, j, l\}$, $s_2 = \{i, m, l\}$, $s_3 = \{j, m, l\}$ or
b) three sets $s_1 = \{i, l\}$, $s_2 = \{j, l\}$, $s_3 = \{m, l\}$ or
c) $s = \{l\}$;
……
(q) q, $q + 1$;
$s_q = \{i_1, i_2, \ldots, i_q\}$, $s_{(q+1)} = \{i_1, i_2, \ldots, i_q, l\}$, where l different from $i_1, i_2, \ldots, i_q$ and such there are in SN
a) sets the number of which is equal to $C_S(q) = C_S(nt - q)$ and the cardinality of which is equal to q, such that $\{i_1, i_2, \ldots, i_{p-1}, i_{p+1}, \ldots, i_q, l\}\backslash\{i_p\}$ for all $p = 1,\ldots, q$ or
b) sets the number of which is equal to $C_S(q - 1) = C_S(nt – (q\text{-}1))$, the cardinality of which is equal to q-1, such that $\{i_1,i_2,\ldots,i_q, l\}\backslash\{i_{pi}, i_{pj}\}$ for all $\{p_i, p_j\} \subseteq \{1,\ldots, q\}$ or

c) sets the number of which is equal to $C_S(q - 2) = C_S(nt – (q\text{-}2))$, the cardinality of which is equal to q-2 , such that $\{i_1, i_2,...., i_q, l\}\backslash\{i_{pi}, i_{pj}, i_{pk}\}$ for all $\{p_i, p_j, p_k\}$, $\{p_i, p_j, p_k\} \subseteq \{1,2,..., q\}$ or

……

d) sets the number of which is equal to $C_S(1) = C_S(nt - 1)$, the cardinality of which is equal to 1, such that$\{l\}$, $l \notin \{i_1, i_2,..., i_q\}$.

The Background Algorithm has an essential disadvantage consisting in the necessity to generate all subsets of s in S_q, $q = 1, 2,\ldots, qmax$. But it is possible constructing directly an element $s \in S_q$, $s = \{i_1, i_2,\ldots, i_q\}$ without generating all of its subsets.

3. A structure of interconnected lists for implementing Background Algorithm

The inductive rules can be used not only for extending sets, but also for cutting off both the elements of S and the sets themselves containing these deleted elements. If element j enters in s_{q+1}, then it must enter in q proper subsets of s_{q+1}. If we observe that j enters in only one doublet (pair), then it cannot enter in any triplet. If we observe that j enters in only one triplet, then it cannot enter in any quadruplet and so on. If an element enters in two and only two doublets, it means that it can enter only in one triplet. If an element enters in three and only three doublets, it can enter in only one quadruplet.

These inductive reasoning are applicable to constructing triplets from doublets, quadruplets from triplets and so on. For instance, if a doublet enters in two and only two triplets, then it can enters in one quadruplet. If a triplet enters in two and only two quadruplets, then it can enter in only one set of five elements. The removal of a certain element (or set of elements) from the examination draws the removal of doublets, triplets, quadruplets,…) into which it enters.

Let us name the procedure for removal of elements and sets containing these elements the procedure of "winnowing". It is convenient to realize this procedure with the use of a Matrix of Correspondences the columns of which are associated with elements of S, and the rows are associated with subsets of S. An entrance $\{i, j\}$ in this matrix equals 1, if index associated with column j enters in s associated with row i.

Consider the following example. The set $S = \{\{1\}, \{2\}, ..., \{14\}\}$.

Consider the Matrix of Correspondences (Table 1) between the 2-component subsets of S possessing a given PROPERTY (S(test-2)) and elements of S appearing in these subsets. In this matrix, the columns are ordered by increasing the number of subsets associated with the columns.

Element 9 enters in one and only one doublet, hence (9,11) cannot be included in any triplet. We can delete the corresponding column and row. We conclude also that set (9,11) cannot enter in any triplet.

Element 5 enters in two and only two doublets, hence it is included in only one triplet (1,5,12). Element 5 cannot be included in any quadruplet. We can delete the corresponding column and rows 2, 3.

Element 6 enters in three and only three doublets, hence it is included in only one quadruplet (4,6,8,11). Element 6 cannot be included in a subset of five indices. We can delete the corresponding column and rows 4, 5, 6.

By analogous reason, we conclude that collection (1,2,12,14) cannot be extended and we can delete the corresponding column and rows 7, 8, 9.

Note that all subsets (9,11), (1,5,12, (4,6,8,11), and (1,2,12,14) possess the PROPERTY.

Element 10 enters in three and only three doublets, hence it is included in only one quadruplet (2,3,8,10). This set does not possess the PROPERTY. In this case, we have to construct all the triplets with element 10. These triplets (2 8 10), (2 3 10), (3 8 10) do not possess the PROPERTY, it means that subsets (2,10), (3,10), (8,10) are maximal ones possessing the PROPERTY. Element 10 can be deleted together with rows 10, 11, 12. Currently, we have generated the following 7 subsets: (1,5,12), (4,6,8,11), (1,2,12,14), (2,3,8,10), (2,3,10), (2,8,10), (3,8,10). Table 2 shows the reduced Matrix of Correspondences.

Table 1. The Matrix of Correspondences for the S(test-2)

Subset	9	5	6	14	10	1	11	8	12	3	7	4	2
(9,11)	1						1						
(1,5)		1				1							
(5,12)		1							1				
(4,6)			1									1	
(6,8)			1					1					
(6,11)			1				1						
(1,14)				1		1							
(2,14)				1									1
(12,14)				1					1				
(2,10)					1								1
(3,10)					1					1			
(8,10)					1			1					
(1,2)						1							1
(1,4)						1						1	
(1,7)						1					1		
(1,12)						1			1				
(3,11)							1			1			
(4,11)							1					1	
(7,11)							1				1		
(8,11)							1	1					
(2,8)								1					1
(3,8)								1		1			
(4,8)								1				1	
(7,8)								1			1		
(2,12)									1				1
(3,12)									1	1			

(4,12)									1			1	
(7,12)									1		1		
(2,3)										1			1
(3,4)										1		1	
(3,7)										1	1		
(2,7)											1		1
(4,7)											1	1	
(2,4)												1	1

Table 2. The reduced Matrix of Correspondences (Reduction 1)

Subset	1	11	8	12	3	7	4	2
(1,2)	1							1
(1,4)	1						1	
(1,7)	1					1		
(1,12)	1			1				
(3,11)		1			1			
(4,11)		1					1	
(7,11)		1				1		
(8,11)		1	1					
(2,8)			1					1
(3,8)			1		1			
(4,8)			1				1	
(7,8)			1			1		
(2,12)				1				1
(3,12)				1	1			
(4,12)				1			1	
(7,12)				1		1		
(2,3)					1			1
(3,4)					1		1	
(3,7)					1	1		
(2,7)						1		1
(4,7)						1	1	
(2,4)							1	1

Element 1 enters in 4 doublets. In this case, we construct the following triplets including element 1: (1,2,4), (1,2,7), (1,2,12), (1,4,7), (1,4,12), (1,7,12). Only two triplets possess the PROPERTY: (1,4,7) and (1, 2, 12). We conclude that element 1 cannot be included in any quadruplet possessing the PROPERTY; hence it can be deleted from consideration with rows 13, 14, 15, 16. Since (1, 2, 12) $\subseteq$ (1, 2, 12, 14)), we conclude that subset (1,4,7) corresponds to maximal subset possessing the PROPERTY but subset (1, 2, 12) is not maximal with respect to the PROPERTY.

Analogously, the consideration of **element 11** leads to constructing the following subsets: (3,4,11), (3,7,11), (3,8,11), (4,7,11), (4,8,11), (7,8,11) from which only

(7,8,11) and (4,8,11) possess the PROPERTY. We conclude that element 11 cannot be included in any quadruplet possessing the PROPERTY; hence it can be deleted from consideration with rows 17, 18, 19, 20. We also conclude that subset (7,8,11) is maximal with respect to the PROPERTY, but (4, 8, 11) does not.

Currently, we have constructed 7 + 12 = 19 subsets. Table 3 shows the reduced Matrix of Correspondences.

With **element 8**, the following subsets can be constructed: (2,3,8), (2,4,8), (2,7,8), (3,4,8), (3,7,8), (4,7,8). But only (2,7,8) possesses the PROPERTY. We conclude that it is maximal with respect to the PROPERTY. Element 8 can be deleted together with rows 21, 22, 23, 24.

For **element 12**, the following triplets can be constructed: (2,3,12), (2,4,12), (2,7,12), (3,4,12), (3,7,12), (4,7,12). Only (3,7,12), (4,7,12) possess the PROPERTY. Since element 12 cannot be included in any quadruplet possessing the PROPERTY, we conclude that (3,7,12), (4,7,12) are maximal with respect to the PROPERTY. Element 12 can be deleted together with rows 25, 26, 27, 28.

Currently, we have constructed 19 + 12 = 31 subsets. Table 4 shows the reduced Correspondent Matrix. In this table, (2,3,4,7), the union of all remaining subsets, possesses the PROPERTY, hence the process of generating subsets is over.

Table 3. The reduced Matrix of Correspondences (Reduction 2)

Subset	8	12	3	7	4	2
(2,8)	1					1
(3,8)	1		1			
(4,8)	1				1	
(7,8)	1			1		
(2,12)		1				1
(3,12)		1	1			
(4,12)		1			1	
(7,12)		1		1		
(2,3)			1			1
(3,4)			1		1	
(3,7)			1	1		
(2,7)				1		1
(4,7)				1	1	
(2,4)					1	1

Table 4. The reduced Matrix of Correspondences (Reduction 3)

Subset	3	7	4	2
(2,3)	1			1
(3,4)	1		1	
(3,7)	1	1		
(2,7)		1		1
(4,7)		1	1	

(2,4)			1	1

Currently, we have constructed 31 + 1 = 32 subsets. Without the procedure of winnowing, it is necessary in Background Algorithm to form 91 + 38 + 3 = 91 +41 = 132 subsets, where 91 doublets, 38 triplets, and 3 quadruplets. The application of winnowing reduced the total quantity of considered subsets to 123: 91 + 32 = 123.

4. A special combinatorial network for implementing the Background Algorithm

The idea of the following algorithm is based on the functioning of a combinatory network structure, whose elements correspond to subsets of a finite set S generated in the algorithm. These elements are located in the network along the layers, so that each q - layer consists of the elements corresponding to subsets the cardinality of which is equal to q. All the elements of q–layer have the same number q of inputs or connections with the elements of previous $(q - 1)$–level. Each element "is excited" only if all the elements of previous layer connected with it are active. The weight of connection going from the excited element is taken as equal to 1; the weight of connection going from the unexcited element is taken as equal to 0. An element of q–layer is activated if and only if the sum of weights of its inputs is equal to q. The possible number N_q of elements (nodes) at each layer is known in advance as the number of combinations of S on q. In the process of the functioning of the network the number of its nodes can only diminish.

An advantage of this network consists in the fact that its functioning does not require the complex techniques for changing the weights of connections and it is not necessary to organize the process of constructing q -sets from their $(q$ -1)-subsets. The nodes of network can be interpreted depending on a problem to be solved. The assigned properties can be checked via different attached procedures.

If an activated node does not possess the assigned property, then it is excluded from the network by setting to 0 all connections going from it to the nodes of above layer. Non-activated node does not require checking whether it possesses the PROPERTY or not. The work of this combinatorial network consists of the following steps:

Step 1. The setting of the first layer nodes of network to active state, the weights of connections leading to the second layer nodes are set equal to 1;

For each level beginning with the second one:

Step 2. The excitation of nodes, if they were not active and all their incoming traffic (links) have the weight equal to 1; checking the assigned property for the activated nodes of this layer;

Step 3. If the assigned property of node is not satisfied, then all the outgoing connections of this node are established to 0. If the assigned property of node is satisfied, then its outgoing connections are set to be equal to 1;

Step 4. The propagation of "excitation" to the nodes of the following higher layer (with respect to the current one) and the passage to analyzing the following layer;

Step 5. "The readout" of the active nodes not connected with above lying active nodes. Such nodes correspond to maximal (not extended) subsets possessing a given property.

In Fig. 1, all the nodes of two first levels are activated but nodes {4,10}, {7,10}, {1,8}, and {1,10} do not possess the given property and they have no active outgoing links. At the third level, only two nodes are activated among which node {4,7,8} does not possess the given property. As a result, we have two nodes corresponding to maximal subsets possessing the given property: {8,10}, {1,4,7}. In the process of network activating, only 12 nodes have been checked and 14 ones did not require to be checked.

Apparently, we can see that the size of network may be a problem if the data is large. But the decomposition of the main problem into sub-problems drastically diminishes the memory size of Background Algorithm. A subproblem is determined by a subnetwork generated by a node of the network.

Generally, the main advantages of combinatorial network are the following ones:

1. The size of network is computed in advance;
2. It is possible to decompose network into autonomic fragments;
3. Different fragments of network can be joined via common nodes;
4. The states of nodes can be established by the use of attached procedures.

This combinatorial network can be used for solving many problems of data mining such that finding frequent patterns, association rule mining, discovering functional dependencies and some others. The application of neural network models for these problems is a new field for investigating. We can refer the readers only to one work in this direction related to optical neural network model used for mining frequent itemsets in large databases [10]. The optical neural network model proposes the most optimized approach with only one database scan and parallel computation of frequent patterns.

5. Conclusion

In this paper, we describe an algorithm, called Background Algorithm, based on the method of mathematical induction. This algorithm is applicable to inferring many kinds of dependencies from a given dataset, for example, functional, implicative dependencies, and association rules. We discussed also the possible ways of increasing the efficiency of the Background Algorithm. For implementation of this algorithm, we proposed a neural network-like combinatorial structure of data an advantage of which consists in the fact that its functioning does not require the complex techniques for changing the weights of connections. The nodes of network can be interpreted depending on a problem to be solved. The assigned properties of nodes can be checked via different attached procedures.

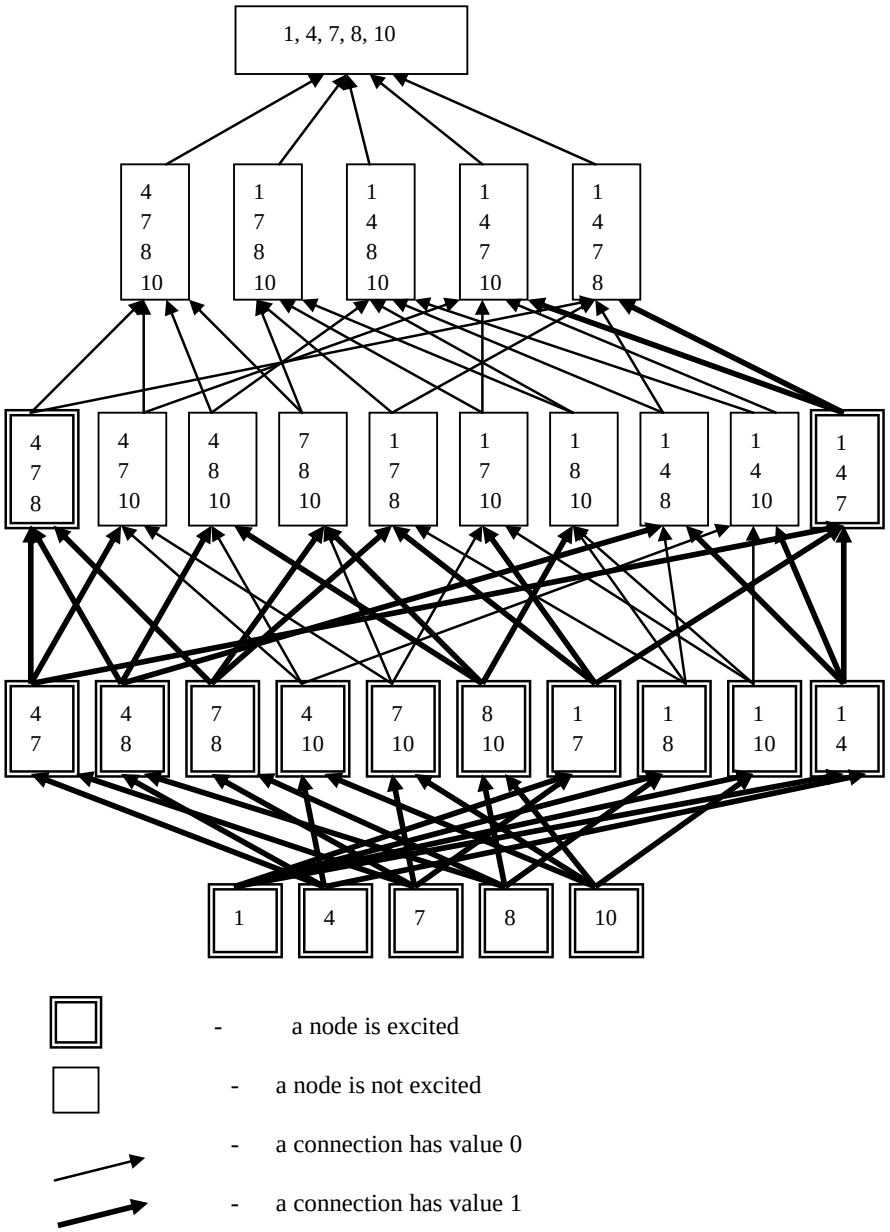


Fig. 1. An example of special combinatorial network

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: P. Buneman, S. Jajodia (eds.), Proceedings of the ACM SIGMOD Conference on Management of Data, pp. 207-216. ACM Press (1993).
2. Houtsma, M., Swami, A.: Set-oriented mining of association rules. Research Report RJ 9567 IBMAlmaden Research Center, San Jose, California (1993).
3. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: J. B. Bocca, M. Jarco, C. Zaniolo (eds.), Proceeding of the 20th VLDB Conference, pp. 487-489. Morgan Kaufman (1994).
4. Agraval, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A. I.: Fast discovery of association rules. In: Fayyard, U. M., Piatetsky-Shapiro, G., Smyth, P., Utchurusamy, R. (eds.), Advances in Knowledge Discovery and Data Mining, pp. 307-328. Menlo Park, CA: AAAI Press (1996).
5. Stumme G.: Efficient Data Mining Based on Formal Concept Analysis. In: R. Cicchetti at al. (eds.), DEXA 2002, LNCS 2453, pp. 534-546. Springer-Verlag, Berlin, Heidelberg (2002).
6. Huhtala, Y., Karkkainen, J., Porkka P., and Toivonen, H.: Tane: an efficient algorithm for discovering functional and approximate dependencies. The computer Journal, 42(2), pp. 100-111 (1999).
7. Naidenova, X.A.: Machine learning as a diagnostic task. In: Arefiev, I. (ed.), "Knowledge-Dialog-Solution", Materials of the Short-Term Scientific Seminar, pp. 26-36. Saint-Petersburg, Russia: State North-West Technical University (1992).
8. Megretskaya, I. A.: Construction of natural classification tests for knowledge base generation. In: Y. Pecherskij (ed.), The Problem of the expert system application in the national economy: Reports of the Republican Workshop (pp. 89-93). Kishinev, Moldava: Mathematical Institute with Computer Centre of Moldova Academy of Sciences (1988).
9. Naidenova, X.A.: DIAGARA: An incremental algorithm for inferring implicative rules from examples. International Journal "Information Theories & Applications, 12(2), 171-186 (2005).
10. Bhatnagar, D. Adlakha, N., Swaroopsaxena, A. A.: Distributed approach for mining frequent itemsets using optical neural network model. International Journal of Engineering Science and Technology, 3(5), pp. 3979-3981 (2011).

Extraction of Semantic Relations between Concepts with KNN Algorithms on Wikipedia

Alexander Panchenko[1,2], Sergey Adeykin[2], Alexey Romanov[2], and Pavel Romanov[2]

[1] Université catholique de Louvain, Centre for Natural Language Processing, Belgium
[2] Bauman Moscow State Technical University, Information Systems dept. (IU5), Russia
{panchenko.alexander,adeykin90,jgc128ra,romanov4400}@gmail.com

Abstract. This paper presents methods for extraction of semantic relations between words. The methods rely on the k-nearest neighbor algorithms and two semantic similarity measures to extract relations from the abstracts of Wikipedia articles. We analyze the proposed methods and evaluate their performance. Precision of the extraction with the best method achieves 83%. We also present an open source system which effectively implements the described algorithms.

Keywords: semantic relations, information extraction, Wikipedia, KNN, MKNN, semantic similarity measure, computational lexical semantics

1 Introduction

There exist many types of semantic relations between words (concepts) – synonymy, metonymy, antonymy, association, etc. In the context of this work, semantic relations are synonyms, hypernyms, and co-hypernyms (words with a common hypernym). These relations are successfully used in various NLP applications, such as word sense disambiguation [1], query expansion [2], document categorization [3] or question answering [4]. Semantic relations are fixed manually in various linguistic resources, such as thesauri, ontologies, and synonym dictionaries. However, existing resources are often not available for a given NLP application, domain, or language. Furthermore, manual construction of the required semantic resources is an extremely expensive and time-consuming process. This motivates the development of new relation extraction methods.

A popular approach to relation extraction is based on the lexico-syntactic patterns [5]. The main drawbacks of this approach are complexity of pattern construction and their language dependency. Methods, based on the distributional analysis [6, 7], do not require any manual labor, but are less precise [8]. Recently, the measures of semantic similarity based on Wikipedia[1] have been proposed [9, 10 ,11]. Wikipedia is attractive for text mining as it contains articles about all the main domains written in all the main languages. Furthermore, it is constantly updated by users. Wikipedia-

[1] Wikipedia, the free encyclopedia that anyone can edit: http://www.wikipedia.org/

based measures show excellent results on the task of correlation with human judgments. Until now, these measures were not used to extract semantic relations.

The approach described in this work fills this gap, and focuses on the application of Wikipedia-based similarity measures to semantic relation extraction. The goal of the method proposed in this article is to discover a set of relations R between a set of input concepts C (e. g. terms of a given domain). In this work, we deal with linguistic concepts, such as words or multiword expressions (not to be confused with a similar notion from the Formal Concept Analysis [23]). The proposed method does not return the type of the relationship between words, i. e. $R \subseteq C \times C$. The method is computationally efficient, sufficiently precise for the practical NLP applications, and can be applied for the languages available in Wikipedia. The main contributions of this paper are the following:

1. The new semantic relation extraction methods, which rely on the texts of Wikipedia articles, k-nearest neighbors algorithms (KNN and MKNN), and two similarity measures (Cosine and Gloss Overlap).
2. An Open Source system Serelex (LGPLv3), which efficiently implements the proposed methods.

In Section 2, we introduce our approach to semantic relation extraction. First, we describe the data and how they are preprocessed in Section 2.1. Next, we discuss the algorithms of semantic relation extraction (Section 2.2) and the used measures of semantic similarity (Section 2.3). Finally, we present key details of the extraction system Serelex (Section 2.4). In Section 3, the experimental results are presented and discussed. Section 4 deals with the related work and Section 5 wraps up with a conclusion and a description of the future research.

2 Semantic Relation Extraction Methods

2.1 Data and Preprocessing

Input data of the method is a set of definitions D for each input concept $c \in C$. We use the data available from the DBPedia.org to build a set of definitions of English terms (multi-word expressions are not included)[2]. For each input concept a pair (c, d) is built, where concept c is an exact title of a Wikipedia article, and definition d is a text of the first paragraph of this article. The experiments described in this work were conducted on a subset of articles with titles containing no numbers and special symbols. We collected 327.167 Wikipedia articles according to this principle. For the goals of our experiments, we prepared two datasets containing 775 words (824Kb) and 327.167 words (237Mb) respectively[3].

Articles were preprocessed as follows. First, we removed all markup tags and special characters. Second, we performed lemmatization and part-of-speech tagging with the TreeTagger [12]. As a result, each word was represented as a triple "to-

[2] We used the file http://downloads.dbpedia.org/3.7/en/long_abstracts_en.nt.bz2
[3] Data are available at: http://cental.fltr.ucl.ac.be/team/~panchenko/def/

ken#POS#lemma", for instance "proved#VVN#prove". An example of a definition in this format is provided below:

axiom; in#IN#in traditional#JJ#traditional logic#NN#logic ,#,#, an#DT#an axiom#NN#axiom or#CC#or postulate#NN#postulate is#VBZ#be a#DT#a proposition#NN#proposition that#WDT#that is#VBZ#be not#RB#not proved#VVN#prove or#CC#or demonstrated#VVN#demonstrate but#CC#but considered#VVN#consider to#TO#to be#VB#be either#RB#either self-evident#JJ#self-evident ,#,#, or#CC#or subject#JJ#subject to#TO#to necessary#JJ#necessary decision#NN#decision .#SENT#.

Senlar [19] and other researchers [7] mention that the methods based on the syntactic analysis achieve higher results than the methods based only on the part-of-speech tagging. However, in our method we intentionally do not use the syntactic analysis for two reasons. Firstly, the computational complexity of the parsing algorithms is very high. Secondly, such a complex linguistic analysis makes the method less robust. Prior researches suggest that quality of parsing in different languages is very different [22]. Moreover, the standard parsers make a lot of errors in the sentences which contain named entities and technical terms, the lexical units which are the most valuable in the context of information extraction.

2.2 Algorithms of Semantic Relation Extraction

Algorithms of semantic relation extraction discussed in this article, are based on the component analysis [13, 14], which states that semantically similar words have similar definitions. The proposed methods use one of two similarity measures: Gloss Overlap of definitions [15] or Cosine between vectors of definitions [16]. The method takes as an input a set of concepts C and outputs a set of relations R between them. Assume that the algorithm is processing the 5 following concepts: $C = \{alligator, animal, building, house, telephone\}$. Its goal would be to recognize the set of semantic relations $R = \{\langle alligator, animal\rangle, \langle building, house\rangle\}$ out of 10 possible pairs of concepts.

The first algorithm calculates semantic relations with the KNN algorithm, the second relies on the MKNN (Mutual KNN) algorithm. The only meta-parameter of the algorithms is the number of nearest neighbors k. The algorithms are presented in Fig.1.

The algorithms first calculate pairwise similarities between all the input terms (lines 1-12). The array of the nearest neighbors `Rmatrix` contains nearest neighbors of each term. We keep the number of elements in each row of `Rmatrix` equals k, which let us minimize the memory footprint. The last stage for the KNN is simply to print the k nearest neighbor concepts for each concept. In contrast to KNN, MKNN establishes a relation only if a pair of words (c_i, c_j) are mutual neighbors (lines 13-21). Therefore, MKNN filters out those relations extracted by KNN which are not mutually related.

Complexity of the developed algorithms is a function of the number of input concepts $|C|$. Time complexity is $O(|C^2|)$ and space complexity is $O(k|C|)$, where k is the number of nearest neighbors.

```
   R = ComponentAnalysis(C, D, k, isMutualKNN)
   Input: C – concepts, D – definitions of concepts, k – number of nearest
   neighbors, isMutualKnn – if true then MKNN, else KNN
   Output: R – set of semantic relations <c_i,c_j> in C X C
1. // Calculation of pairwise similarities between words all concepts C
2. Rmatrix = void
3. for i=0; i<count(C); i++ {
4.        for j=i; j<count(C); j++ {
5.               // Calculation of semantic similarity of two concepts
6.               s_ij = similarity(D(i), D(j))
7.               // Saving most similar concepts
8.               if( count(Rmatrix(C(i))) < k || s_ij > min(Rmatrix(C(i)) ){
9.                      Rmatrix(C(i)).addOrReplaceMin(C(j))
10.              }
11.       }
12. }
13. // Calculation of semantic relations
14. R = void
15. foreach c_i in Rmatrix {
16.       foreach c_j in Rmatrix(c_i) {
17.              if(!isMutualKNN || Rmatrix(c_j) contains c_i){
18.                     R.add(<c_i, c_j>)
19.              }
20.       }
21. }
22. return R
```

Fig. 1. Semantic relation extraction algorithms KNN and MKNN.

2.3 Measures of Semantic Similarity

Function `similarity` (line 6) in the algorithms KNN and MKNN calculates a pairwise semantic similarity of two concepts $c_i, c_j \in C$, from their definitions $d_i, d_j \in D$. The larger the value of semantic similarity, the closer the "sense" of the concepts. Two similarity functions are considered here. The first is the gloss overlap of the definitions d_i, d_j of the concepts c_i, c_j:

$$similarity(c_i, c_j) = \frac{2|d_i \cap d_j|}{|d_i| + |d_j|}.$$

Here the numerator is the number of the common words in the definitions; $|d_j|$ is the number of words in the definition d_j. The second measure is the cosine between vectors $\boldsymbol{f}_i, \boldsymbol{f}_j$ of definitions d_i, d_j representing concepts c_i, c_j:

$$similarity(c_i, c_j) = \frac{\boldsymbol{f}_i \cdot \boldsymbol{f}_j}{\|\boldsymbol{f}_i\| \, \|\boldsymbol{f}_j\|} = \frac{\sum_{k=1,N} f_{ik} f_{jk}}{\sqrt{\sum_{k=1,N} f_{ik}^2} \sqrt{\sum_{k=1,N} f_{jk}^2}}.$$

Here f_{ik} is the frequency of the lemma c_k in the definition d_i. The both similarity measures use the lemmas (e. g., animals#NNS#animal), and do not use stopwords found in the definitions. For the both of similarity measures only matches of the nouns (NN, NNS, NP), verbs (VV, VVN, VVP), and adjectives (JJ) are considered.

2.4 Relation Extraction System Serelex

The system is a console application implemented in C++ and available for Windows and Linux platforms (32/64 bits). It consists of the definition class, global parameters class, component analysis class, and several additional classes and functions (see Fig.2). The main functions of the program are:

- loading files of stopwords and input concepts C;
- loading the file with definitions D taking into account the stopwords;
- calculation of the pairwise semantic similarities between the input concepts C;
- building the list of the semantic relations R.

In order to achieve high performance of the system, we map each word to a numerical identifier. This procedure significantly reduces running time of the program. The system extensively uses the Standard Template Library (STL) and the Boost library[4]. The source code and the binaries of the Serelex are available under the conditions of the LGPLv3 license at https://github.com/AlexanderPanchenko/Serelex.

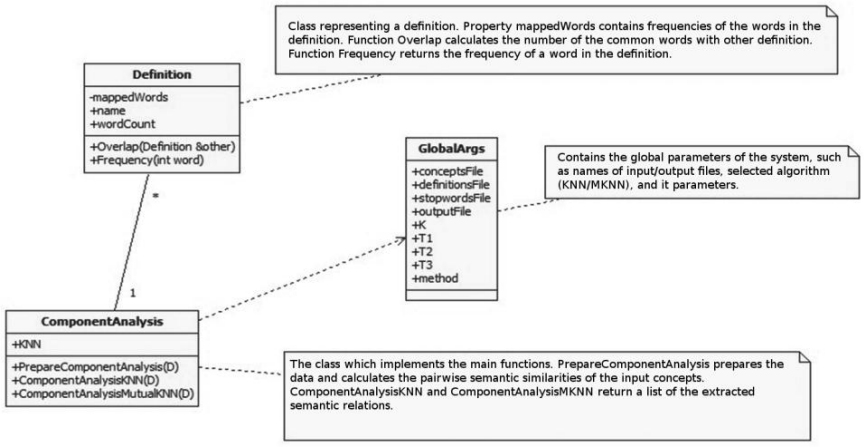


Fig. 2. The main classes of the semantic relation extraction system Serelex.

3 Results

We investigated the algorithms KNN and MKNN with the two measures described above and with various numbers of nearest neighbors k (see Fig. 3). As one may expect, the number of extracted relations linearly depends on the number of nearest neighbors k both for KNN and MKNN. The number of extracted relations depends little on the similarity measure type. The key difference between the two measures is that Cosine takes into account frequencies, while Gloss Overlap does not. The little

[4] Boost C++ libraries: http://www.boost.org/

difference in the results is likely to be due to the fact that the definitions are short. Thus, frequency information does not contribute a lot to the result.

The algorithm KNN extracts more relations than the MKNN for the same value of k. It happens because the MKNN filters out pairs of concepts which are not mutual nearest neighbors. According to our experiments, MKNN filters around 50-70% of the relations extracted by KNN, depending on the number of the concepts $|C|$ and the value of k.

We estimated the precision of the extraction between a set of 775 concepts for both algorithms with $k = 2$. In order to measure the precision, we manually labeled the files with the extracted relations. The precision was calculated as the number of correctly extracted relations to the number of extracted relations. The results are presented in Table 1. The examples of extracted relations between a set of 775 concepts with algorithm MKNN ($k = 2$) and the gloss overlap similarity measure are presented below[5]:

$$R = \{\langle acacia, pine\rangle, \langle aircraft, rocket\rangle, \langle alcohol, carbohydrate\rangle, \langle alligator, coconut\rangle, \langle altar, sacristy\rangle, \ldots \langle object, library\rangle, \langle object, pattern\rangle, \langle office, crew\rangle, \langle onion, garlic\rangle, \langle saxophone, violin\rangle, \langle saxophone, clarinet\rangle \langle tongue, mouth\rangle, \ldots, \langle watercraft, boat\rangle, \langle watermelon, berry\rangle, \langle weapon, warship\rangle, \langle wolf, coyote\rangle, \langle wood, paper\rangle\}.$$

Due to the huge number of extracted relations (see Fig. 3), it is hard to calculate manually the extraction precision for all values of k. We expect the precision to decrease for values of $k > 2$. According to our preliminary experiments, we recommend to use the number of the nearest neighbors $k \in [1; 10]$. In future, we plan to use the WordNet [17] and the standard evaluation datasets, such as BLESS [18], to estimate the precision of the method more robustly.

The performance of the developed system is rather high. For instance, the file with 775 definitions is processed for less than 3 seconds on a Linux server equipped with Intel(R) Xeon(R) CPU E5606@2.13GHz (the program does not use multithreading). The algorithm KNN used with the measure Gloss Overlap processes the file with the 327.167 definitions within 3 days 3 hours and 47 minutes.

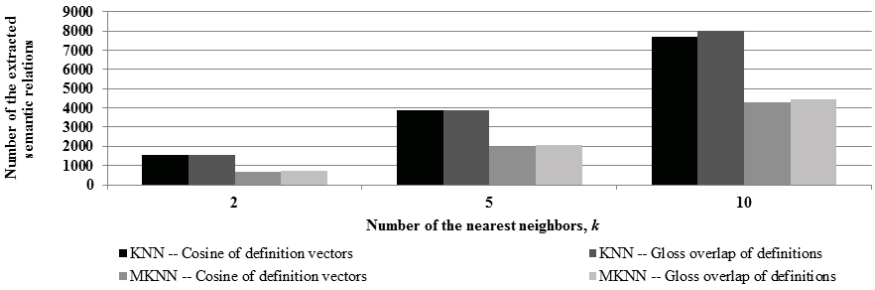


Fig. 3. Dependence of the number of extracted relations on the number of nearest neighbors k.

[5] The full list of the extracted relations with this configuration is available at http://cental.fltr.ucl.ac.be/team/~panchenko/def/results-775/overlap_mknn_2.csv

Table 1. Precision of relation extraction for 775 concepts with the KNN and MKNN ($k = 2$).

Algorithm	Similarity Measure	Extracted Relations	Correct Relations	Precision
KNN	Cosine	1548	1167	0.754
	Gloss overlap	1546	1176	0.761
MKNN	Cosine	652	499	0.763
	Gloss overlap	724	603	**0.833**

4 Related Work

Senlar [19] presents a recent overview of semantic relation extraction methods based on text corpora and electronic dictionaries. The automatic thesaurus construction system SEXTANT extracts relations between words with precision around 75%. Measures of semantic similarity based on the Web achieve precision in TOEFL synonymy test up to 74%.

WikiRelate! is the most similar system to ours. It was proposed by Strube and Ponzetto in 2006 [9]. The main differences of our method and the system from this development are the following:

- Serelex extracts semantic relations, while WikiRelate! only implements a semantic similarity measure;
- The source code of WikiRelate! is not available, while the binary version is available only for research purposes. The source code of Serelex is available under the conditions of the industry-friendly LGPLv3 license;
- Serelex can calculate similarity not only between texts of Wikipedia, but also between any other definitions represented in the corresponding format;
- Serelex implements two measures of similarity (Cosine and Gloss Overlap), while WikiRelate! implements only the Gloss Overlap measure. Furthermore, the designers of the system implement the Gloss Overlap in a different way: they count each match of n-gram as n^2 matches;
- Serelex does not use the category lattice of Wikipedia.

WikiRelate! achieves a correlation with human judgments of 0.59. However, since WikiRelate! does not extract relations, we cannot directly compare its performance with our results.

In [10] and [11], the authors proposed alternative measures of semantic similarity based on texts of Wikipedia. However, those approaches are less similar to Serelex than the WikiRelate!. In particular, these measures represent the concepts in a vector space of all Wikipedia articles, while Serelex uses the traditional bag-of-words space. Nakayama et al. [20] suggested yet another relation extraction method based on Wikipedia, which is significantly different from our method. The authors use the hyperlinks structure of Wikipedia articles to infer associations between words. Finally, Milne et al. [21] suggested to extract synonyms, hyponyms, and associations from Wikipedia category lattice, and other structure and navigational elements of Wikipedia.

5 Conclusion

We proposed and analyzed the method for semantic relation extraction from texts of Wikipedia with algorithms KNN and MKNN and two semantic similarity measures. The preliminary experiments showed that the best results (precision of 83%) are obtained with the method based on MKNN algorithm and Gloss Overlap similarity measure. We also presented an open source system, which efficiently implements the proposed method.

The developed method seems to be enough precise and computationally efficient for the practical NLP applications. Currently, we are working on integration of the extracted semantic relations in a text categorization system [24]. Another application of the extracted relations, which we would like to investigate, is query expansion.

The method is able to calculate relations between a huge number of concepts, each of which is represented by a title of a Wikipedia article. Thus, it could be used to extract relations between 3.8 million of concepts in English Wikipedia and 17 million of concepts in other 282 languages of Wikipedia. The only language-dependent resources used in the method are stoplist, part-of-speech tagger, and lemmatizer. These resources are available for the most of the European languages for free. Finally, Serelex can extract relations from other sources of definitions, such as traditional dictionaries or Wiktionary, provided these data are provided in the proper format.

The main directions of our future research are: (1) using the developed method to extract relations in Russian, French, and German languages; (2) improving the precision of the extraction by clustering of the obtained semantic relations graph.

Acknowledgments

Yuri Philippovich, Adrien Dessy, Olga Morozova, Nicolas Danhier, and three anonymous reviewers provided comments and remarks, which considerably improved quality of the paper. Alexander Panchenko thanks Wallonie-Bruxelles International (WBI) foundation for the support.

References

1. Patwardhan, S., Pedersen, T.: Using WordNet-based context vectors to estimate the semantic relatedness of concepts. In: Proceedings of the EACL 2006 Workshop Making Sense of Sense, pp. 1-12 (2006)
2. Hsu, M.H., Tsai, M.F., Chen, H.H.: Query expansion with conceptnet and wordnet: An intrinsic comparison. In: Information Retrieval Technology, pp. 1–13, Springer (2006)
3. Tikk, D., Yang, J.D., Bang S.L.: Hierarchical text categorization using fuzzy relational thesaurus. In: KYBERNETIKA-PRAHA, vol. 39(5), pp.583–600 (2003).
4. Sun, R., Jiang, J., Fan, Y., Hang, T., Tatseng, K., Yen Kan, C.M.: Using syntactic and semantic relation analysis in question answering. In: Proceedings of the TREC (2005)
5. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora, In: Proceedings of the 14th conference on Computational linguistics COLING '92, pp.539-545 (1992)

6. Lin D. Automatic retrieval and clustering of similar words. In: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics, pp.768-774 (1998)
7. Heylen, K., Peirsman ,Y., Geeraerts, D., Speelman, D.: Modelling word similarity: an evaluation of automatic synonymy extraction algorithms. In: Proceedings of the Sixth International Language Resources and Evaluation (LREC'08), pp. 3243-3249 (2008)
8. Curran, J.R., and Moens, M.: Improvements in automatic thesaurus extraction. In: Proceedings of the ACL-02 Workshop on Unsupervised Lexical Acquisition. pp. 59-66 (2002)
9. Strube, M., and Ponzetto, S.P.: WikiRelate! Computing semantic relatedness using Wikipedia. In: Proceedings of the National Conference on Artificial Intelligence, pp. 1419-1429, (2006)
10. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. In: International Joint Conference on Artificial Intelligence, pp. 12-20 (2007)
11. Zesch, T., Müller, C., Gurevych, I.: Extracting lexical semantic knowledge from wikipedia and wiktionary. In: Proceedings of the LREC, pp. 1646–1652, 2008.
12. Schmid, H..: Probabilistic Part-of-Speech Tagging Using Decision Trees. pp. 44–49, (1994)
13. Philippovich, Y.N., Prokhorov, A.V.: Semantika informatsionnih tehnologii: opiti slovarno-tezaurusnogo opisaniya (In Russian). Series «Computational Linguistics». M.:MGUP, http://it-claim.ru/Library/Books/CL/CLbook.htm (2002)
14. Kobozeva, I.M.: Komponentnii analiz leksicheskogo znacheniya (In Russian). In: Linguistical Semantics: 4rd ed.. M.: Edition «LIBRICOM», pp. 109-122 (2009)
15. Banerjee, S., Pedersen, T.: Extended Gloss Overlaps as a Measure of Semantic Relatedness,In: Proceedings of the International Joint Conference on Artificial Intelligence (2003)
16. Jurafsky, D., Manning, H. M.: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Second Edition. pp.697-701, 2009.
17. Fellbaum, C.: WordNet. Theory and Applications of Ontology: Computer Applications, pp. 231-243, Springer (2010)
18. Baroni, M., and Lenci, A.: How we BLESSed distributional semantic evaluation. In Proceedings of GEMS Workshop of EMNLP 2011 (2011)
19. Senellart, P., Blondel, V. D.: Automatic Discovery of SimilarWords. In: Survey of Text Mining II. vol. 1, pp. 25-44, Springer London (2008)
20. Nakayama, K., Hara, T., and Nishio S.: Wikipedia Mining for an Association Web Thesaurus Construction. In: Web Information Systems Engineering – WISE, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 322-334 (2007)
21. Milne, D., Medelyan, O., and Witten, I.H.: Mining Domain-Specific Thesauri from Wikipedia: A Case Study. In: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, pp. 442-448, IEEE Computer Society (2006)
22. Candito, M., Nivre, J., Denis, P. Anguiano, E. H.: Benchmarking of statistical dependency parsers for French. In: Proceedings of the 23rd International Conference on Computational Linguistics COLING '10. Stroudsburg, PA, USA, pp.108-116 (2010)
23. Ganter, B., and Wille, R., and Franzke, C.: Formal concept analysis: mathematical foundations. Springer-Verlag New York (1997)
24. Panchenko, A., Beaufort, R., Fairon, C.: Detection of Child Sexual Abuse Media on P2P Networks: Normalization and Classification of Associated Filenames. In: Proceedings of Workshop on Language Resources for Public Security Applications. The 8th International Conference on Language Resources and Evaluation (LREC), Istambul, Turkey (2012

Computerized Recognition System for Historical Manuscripts

Artem Skabin

Petrozavodsk State University, Petrozavodsk, Russia
artb00g@gmail.com

Abstract. The article describes the process of creating a universal computerized recognition system of historical manuscripts, including historical shorthand records dating back to the 19th and early 20th centuries. We discuss the problem of getting the original graphical representation of symbols from historical manuscripts using a threshold binarization method. We search for a similar graphical representation of symbols in the database. Moreover we present a prototype of a computerized recognition system of historical manuscripts.

Keywords: character recognition, historical manuscript, binarization, threshold method.

1 Introduction

Nowadays Russian archives have a large amount of encoded shorthand records. The reason is the shorthand writers' inability to decipher historical records. During the 19th and early 20th centuries Russian shorthand writing was in the making. So the existing documents were deciphered in different systems. Moreover, modern shorthand writing differs significantly from the historical stenography systems of the 19th century. The main difficulties in decoding shorthand records are:

1. the lack of specialists in the area of 19th and early 20th century shorthand writing systems. There are only old books;
2. the shorthand writer's inability to use standard characters, because usually stenographists deciphered the texts themselves and used their own symbols;
3. there was a widespread custom of skipping vowels or replacing repeated combinations of characters and words with one symbol;
4. the fact that some characters from shorthand records could have a similar spelling, but depending on certain physical parameters such as height, can have different meanings.

The aim of this work is to create a universal computerized recognition system of historical manuscripts, including from historical shorthand records of the 19th and early 20th centuries. It is to solve the problem of description and decoding historical transcripts, as well as to introduce new documents to the scientific world.

2 Description of the developed system

Special features of the developed system are a historical account of the 19th and 20th centuries spelling characteristics, an account of the individual characters of different shorthand writers, the ability of critical analysis, the usage of dictionaries for help in deciphering the texts, etc. [1] The information system will be publicly available and offered to be used by archive professionals and librarian scientists. The fine-tuning of the system was done using the transcripts by Snitkina partially decoded C. Poshemyanskoy and P. Olkhin's book [2]. Recognition of any text includes the following steps:

1. image preprocessing, usually an image binarization;
2. segmentation, i.e. selection of the text in the preprocessed image, such as characters, combinations of characters, words, lines;
3. analysis of the derived segments by establishing the values, characteristics, comparing with reference standards that could be found in the knowledge base;
4. decoding by choosing the most appropriate word forms from dictionaries of equivalents with a specific language model.

Additional difficulties in the text recognition are caused by curving rows, brightness drops, transparency of the text on the reverse side and other defects of the original text and image. Manuscript recognition is more difficult in contrast to the recognition of printed texts [3].

The goal of the research is to create fairly universal software for computerized recognition of historical manuscripts for which it has been impossible in the past. The suggested computerized recognition system of historical manuscripts with the possibility of intelligent decision support will significantly accelerate the process of conversion of manuscripts into text files and increase the accuracy of their decipherment. The software will have the following characteristics [1]:

1. the system automatically monitors the state of keying in and interactively displays information to the user;
2. the system returns the user the word forms variants sorted by frequency of occurrence in the database and information on the absence of the words typed in the database.

3 Binarization of historical manuscripts

While decyphering historical manuscripts a problem with image binarization occurs. Due to the aged condition of the image and the fact that the shorthand records were made in pencil on yellow paper the threshold method in color components (RGB) was not suitable for this task. This was caused by the characters and background pixels which have similar values as color components. As it can be seen on the histograms (Fig. 1) the absence of two clearly defined peaks, does not allow choosing the threshold value for binarization. Similar results are observed (Fig. 1), if we use HSB colour scheme decomposition (hue, saturation, brightness). If using threshold method binarization of brightness you can get a clear character, with a small amount of noise.

Threshold intensity was experimentally determined by choosing those values for which the most precise symbols came out with the least amount of noise. Binarization

was performed on around 1500 fragments from these 30 historical documents. The best results are achieved when the percentage of black pixels after binarization is approaching 13% of the total number of pixels.

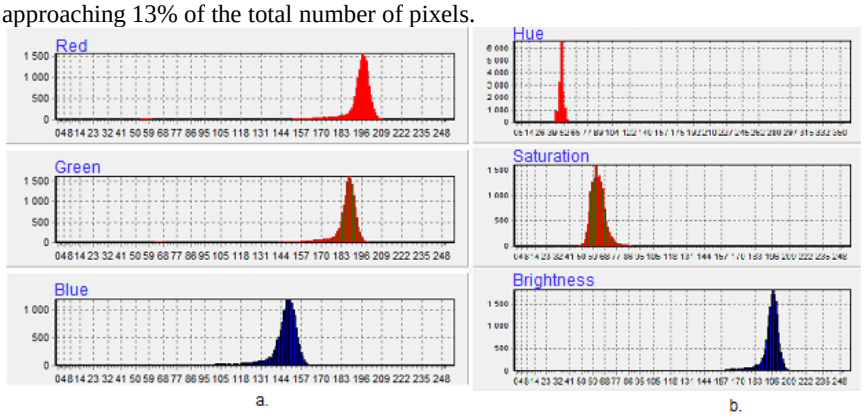


Fig. 1. Histograms of color schemes RGB (a) and HBS (b).

4 The module for creating original graphical representation of characters

The system was divided into several modules. One of them is the module for creating an original graphic representation of the characters. Fig. 2 shows an interface module for creating original graphic characters. The main window consists of two areas: the original image (original transcript) is on the left panel where the user selects the desired character (Fig. 2), the selection location is displayed on the second panel. There the processed shorthand record is located, i.e. all of the characters that could be found in the same place where the original symbols are.

After selecting a character the user should click on the "hot key" or a combination thereof. Then the system performs binarization selection and segmentation. If you receive multiple segments, the system prompts the user to choose which segment or segments correspond to the original character. If you selected several segments, the system binds [4] the broken "pieces" and provides the user with the result. When the user is satisfied with the result, the symbol is saved in the database and is located on the right panel respectively to the location (coordinates) on the original image. If the result does not meet user's requirements, it is possible to edit the received symbol with a simplified graphics editor.

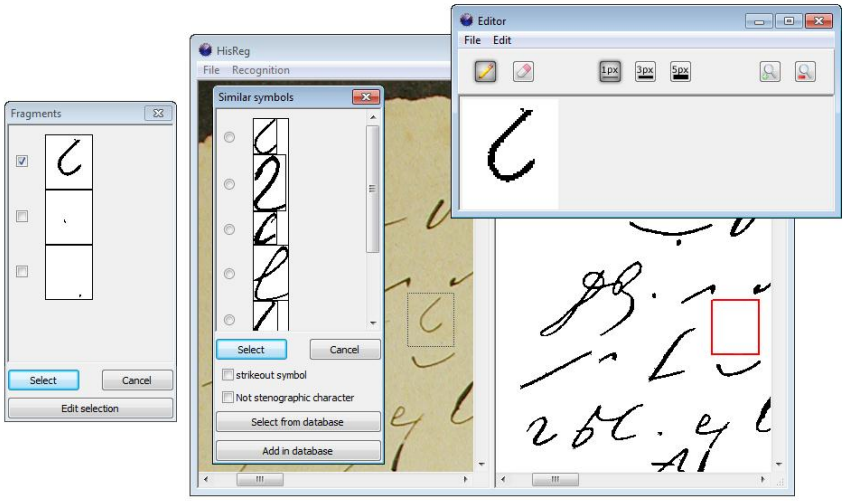


Fig. 2. Interface module for creating original graphical representation of characters

The creation of an original graphical representation of characters is a difficult task for the following reasons:

1. The original image is quite old and was written in pencil on yellowed paper which has distortions, various types of damage; moreover some shorthand records have irrelevant records with no meaning or there are lines intersecting with the symbols;
2. There were gaps of characters in binarization, as some pixels of the character had a similar color to the pixels of the paper;
3. There was a need for segmentation into individual characters of the symbols that were written together.

5 Character search in the database

During the creation of original graphic characters, the problem of forming a characters database appeared. The database is needed to avoid duplication and redundancy of graphic symbols. The database is extensible, i.e. if in the process of recognition the character has not been previously met, it is added to the database. The basis of the database was taken from a sample of 250 characters, which is equal to the number of characters in the alphabet used by Snitkina, randomly selected from the manuscripts. We used the following methods to compare the current characters with the characters from the database: pattern matching, the comparison with the skeleton of the pattern, the boundary distance method and method of projections. Comparison of these methods is represented in Table 1.

Table 1. Comparison of methods for the learning sample.

characteristic / method	Search time	Accuracy
pattern matching	3 sec. (depending on the size of the symbol)	< 30%
the comparison with the skeleton of the pattern	1-2 sec. (depending on the size of the symbol)	~40%
method of projections	~0.5 sec.	~40%
the boundary distance method	< 0.01 sec.	> 60%

The low accuracy of the comparison with the pattern is caused by binarization. The character could have different thickness depending on the size of the shorthand records selected. While comparing the skeletons of characters, for skeletolization we used Zhang Suen's algorithm [5]. This algorithm for finding characters in the database works as follows: using the method of classification symbols of height to width ratio, the characters are divided into three classes: wide, high, square. Next, is determined to which class the current character belongs. And then there is a search for similar character in this class using the boundary distances method. The boundary distances method consists of choosing symbols with a similar height to width ratio. Then the current symbol is measured $\{l_1, l_2, \ldots, l_8\}$ (Fig. 3).

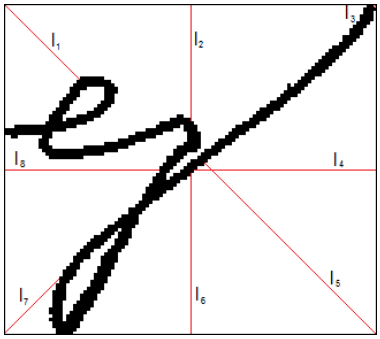


Fig. 3. The boundary distance method

The characters, that are selected from the database, have the distance $\{l_1', l_2', \ldots, l_8'\}$ and are located in the interval $(l_1 \cdot k - \varepsilon, l \cdot k + \varepsilon)$, where k is the height to width ratio of the current symbol, $\varepsilon = k \cdot l \cdot \alpha$ where $\alpha = 0.1$.

During processing of 29 sheets of shorthand records, more than 2,500 diagrams of the original characters have been allocated. After that we met a problem of matching

the distance parameters derived from the book Snitkina and partially transcribed records. The prototype interface of this system is presented below.

6 The prototype of computerized recognition system of historical manuscripts

Fig. 4 shows the interface of the prototype of our computerized recognition system for historical manuscripts. This system has four main areas: the area with the original image, the range of possible values of deciphered symbols or groups of characters.

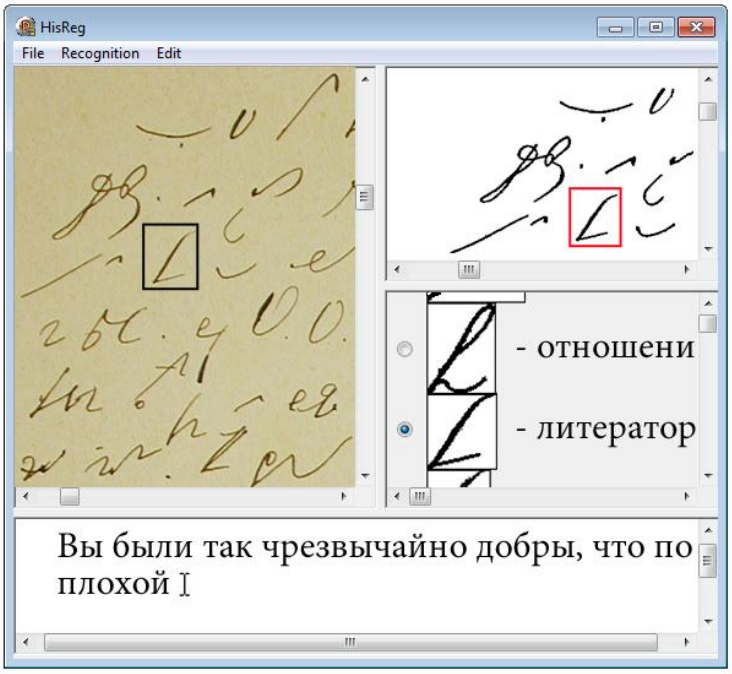


Fig. 4. Interface of the computerized system for historical manuscript recognition.

When the user selected a character in the original image, the image of the symbol is located in the 2nd panel, at the same place where it is on the original shorthand record. The fourth area displays the decoded symbols. The system analyzes components of the word in the process of keying it in, and offers the user the closest interpretations in meaning from the database. The system produces an automatic decoding of similar characters or groups of characters by analyzing the original image while the characters are keyed in. The main advantages of this system are the following:

1. the ability to use "hot keys" which accelerates the keying in of a historical shorthand record;
2. the connection of graphic and text representation of a shorthand record;

3. intellectualized typing;
4. the ability to automatically recognize similar combinations of characters, words in the text;
5. the opportunity to work with a multi-user dictionary.

This system is designed to accelerate the process of deciphering handwritten historical shorthand records. The system's future realization as a Web-service for organizations working with shorthand records is possible.

7 Conclusion

The threshold method of image binarization is used in the module; with the selected parameters the binarization goes to the utmost. These parameters are specific to each type of document, so there is a need to use a more adaptive method of binarization as described in [6].

Current methods for image search on database do not provide high accuracy; because of that original symbol diagram redundancy may occur. As a result, it is necessary to use a method that gives a higher accuracy. This will be analyzed by the methods of individual signatures verification [7].

Acknowledgments. This research supported by a grant RGNF № 11-01-12026v (head Rogov AA). We would like to thank Dmitry Ignatov and Jonas Poelmans for language improvements of the final submission.

References

1. Rogov, A.A., Talbonen, A.N., Varfolomeev, A.G.: Automated recognition of handwritten historical documents. Digital libraries: advanced methods and technologies, digital collection: Proceedings of the XII All-Russian Scientific Conference RCDL'2010, Kazan. Univ of Sciences, P. 469-475 (2010) (in Russian)
2. Olkhin, P.: Guide to the Russian shorthand. Printing Dr. M. Khan, St. Petersburg (1866) (in Russian)
3. Gorski, N., Anisimov, V., Gorskaya, L.: Mountain handwriting recognition tech-hundred: from theory to practice. Polytechnics, St. Petersburg (1997) (in Russian)
4. Nagabhushan, P., Anami, B. S.: A knowledge-based approach for recognition of handwriting Pitman shorthand language storkes. In: P. Nagabhushan, Basavaraj S. Anami. Sadhana., Vol. 27, Part 5, P. 685–698 (2002)
5. Zhang, T.Y.: A fast parallel algorithm for thinning digital patterns / T. Y. Zhang, C. Y. Suen. Commun. ACM. Vol. 27, №3, P. 236-239 (1984)
6. Pratikakis, I., Gatos, B., Ntirogiannis, K.: ICDAR 2011 Document Image Binarization Contest (DIBCO 2011). In ICDAR, P. 1506–1510 (2011)
7. Kukharev, G.A.: Biometric Systems: Methods and means of identification of human personality. Polytechnics, St. Petersburg (2001) (in Russian)

An Ontology–Based Approach to Text–to–Picture Synthesis Systems

Dmitry Ustalov, Aleksandr Kudryavtsev

Ural Federal University, Yekaterinburg, Russia
dmitry@eveel.ru, vt@dpt.ustu.ru

Abstract. In this paper, we present an ontology–based approach to text–to–picture synthesis. Our approach operates with an ontology in the RDF/XML format. This provides loose coupling of the system components, unification of the interacting objects representation and their behaviour, and makes possible verification of system information resources.

Keywords: text–to–picture, text–to–scene, natural language processing, depiction rules, semantic representation, Web Ontology Language, Resource Description Framework.

1 Introduction

A picture is worth of a thousand words. The text–to–picture synthesis problem is actual because of existence of many domains where clearness of textual information is necessary: foreign language learning [12], traffic accident visualization [1], rehabilitation of people with cerebral injuries [4], etc.

Utkus [10] is a text–to–picture synthesis system (TTP system) that is developed since 2011 and is designed to work with small texts of 1–3 Russian sentences: fragments from children literature, microblog posts, news summaries, comments on Web–sites. These texts are suitable for automatic processing and further visualization [13]. The current work differs from previous TTP systems in its focus on conveying the gist of general, semantically unrestricted Russian language text.

TTP systems have three stages of processing [2]:

1. A stage of *linguistic analysis* — tokenization, morphological and syntactic parsing, obtaining the semantic representation of the input text;
2. A stage of *depictors generation* — generation of the set of graphical depictors that corresponds with obtained semantic representation;
3. A stage of *picture synthesis* — construction of vector or raster image from the graphical primitives that are positioned following the generated depictors.

In TTP systems, every processing stage strongly depends on many information resources, including:

– Thesaurus that containts words and their relations (synonymy, hyponymy, etc);

- Gallery that contains different graphical primitives for interacted objects (actors), which becomes rendered in the final images;
- Depiction rules that define how one or many actors can be depicted into the final images;
- Frames, which describe allowed properties of actors.

The volume and complexity of these resources are high. Therefore, TTP systems must have a straight way to connect such resources during the text processing.

2 Related works

There are several full–functional analogues that are described in various papers [1,2,4,5,11,13]. Unfortunately, an approach to unification the information resources is presented only in [2]. That paper presented the WordsEye system, which builds 3D scenes by with certain descriptive English sentences, e.g., "The huge head is on the tan horse. The horse is on the extremely tall mountain range. The fence is 10 feet behind the horse. The fence is 50 feet long."

The following desicions are made in the WordsEye system:

1. WordNet thesaurus is used to identify the semantic relations between separate words;
2. During the text processing, specially defined frames are mapped into the found syntactic groups to obtain additional information about actors: colour, size, etc;
3. Behaviour that is implemented in known actions (verbs), and is described in *depiction rules*, which are defined in a declaration–style Lisp program;
4. A proprietary Izware Mirai 3D animation system is used with Viewpoint Model Library to perform visualization problem.

Note two significant drawbacks of these decisions:

1. Despite of rich possibilities of the Lisp programming language, usage of this language complicates the replenishment of depiction rules set because of high requirements of developers experience;
2. Work in 3D demands considerable efforts and resources, which are not justified by final quality: in most cases, it is enough to deal with 2D images [12].

3 Suggested approach

Similarly to [2], we consider actors in terms of object paradigm:

1. Actors have properties: colour, etc;
2. Actors have methods: functions that reflect actors relations: *to fall*, *to lay*, etc.

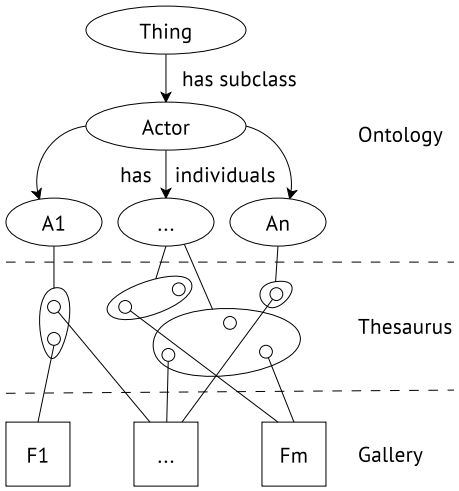


Fig. 1. Connection of ontology, thesaurus, and gallery

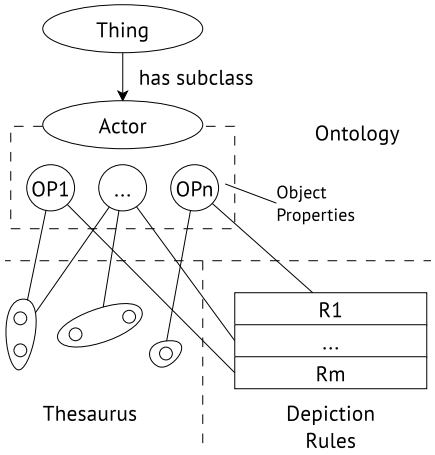


Fig. 2. Connection of ontology, thesaurus, and depiction rules

We propose to formalize into an ontology all the knowledge about actors: their possible characteristics and relations. We also propose to split the ontology, thesaurus, depiction rules and gallery to provide loose coupling of these components of the TTP system (Fig. 1, 2):

– Words and their semantic relations are represented in a thesaurus;

- Several figures from gallery can be associated with each word in thesaurus: despite words *tomcat* and *cat* are antonymes by gender, they both are hyponymes to word *animal*;
- Ontology has the class *Actor*, and instances of this class are linked to synsets in thesaurus. Therefore, for every *set of synsets* an *Actor* instance can be defined by correspondent properties;
- There are defined *object properties* for *Actor* instances. These object properties are associated with verb synsets in thesaurus and represent all possible relations among actors (i.e., `fall(actor)` and `fallTo(actor1 actor2)`);
- Also, there *data properties* are defined and represent different parameters of actors (e.g., colour);
- Depiction rules that specify the behaviour of each object property (Fig. 2) are defined in a separate XML document.

Elements of ontology are linked to thesaurus synsets using the OWL annotation mechanism. It is important to note that one element can be linked to many synsets. These synsets can belong to thesauri of different language because of internationalization method that is implemented in OWL.

3.1 Examples

The *Actor* class is a direct subclass of the *Thing* class:

```
<owl:Class
  rdf:about="http://utkus.eveel.ru/World.owl#Actor"/>
```

Instances of the *Actor* class (Fig. 3) can be linked to synsets using OWL annotations:

```
<owl:NamedIndividual
  rdf:about="http://utkus.eveel.ru/World.owl#Man">
    <rdf:type
      rdf:resource="World.owl#Actor"/>
    <synset xml:lang="ru">2039</synset>
    <synset xml:lang="ru">2040</synset>
    <synset xml:lang="ru">238</synset>
    <synset xml:lang="ru">6939</synset>
    <synset xml:lang="ru">75</synset>
</owl:NamedIndividual>
```

Object properties are also defined for the *Actor* class, and they represent all the predicates that are operated by the system. *Object properties* are connected with depiction rules that specify their behaviour.

Equivalence relation is possible between *object properties*. In our approach, the SPO–triples[1] `(fall man)` and `(fall man chair)` will be attributed to different *object properties*: `fall` and `fallTo`.

[1] SPO — a tuple of predicate, subject, and object.

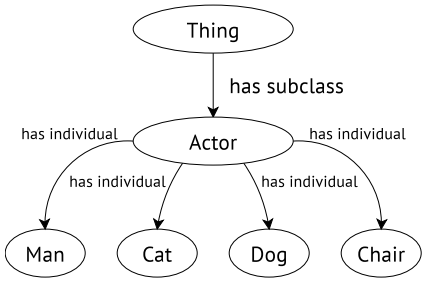


Fig. 3. Ontology fragment with *Actor* instances

```
<owl:ObjectProperty
  rdf:about="http://utkus.eveel.ru/World.owl#fall">
    <synset xml:lang="ru">106</synset>
    <synset xml:lang="ru">397</synset>
    <synset xml:lang="ru">406</synset>
    <rdfs:domain
      rdf:resource="World.owl#Actor"/>
    <owl:equivalentProperty
      rdf:resource="World.owl#fallTo"/>
</owl:ObjectProperty>

<owl:ObjectProperty
  rdf:about="http://utkus.eveel.ru/World.owl#fallTo">
    <synset xml:lang="ru">106</synset>
    <synset xml:lang="ru">397</synset>
    <synset xml:lang="ru">406</synset>
    <rdfs:domain
      rdf:resource="World.owl#Actor"/>
    <rdfs:range
      rdf:resource="World.owl#Actor"/>
</owl:ObjectProperty>
```

To represent detected *object properties* on the final picture, it is necessary to assign the specific behaviour to each known *object property*. This behaviour is specified by *depiction rules* which are declared in a separate XML document. For the `fallTo` *object property* we have the following *depiction rule*:

```
  <rule rdf:about="http://utkus.eveel.ru/World.owl#fallTo">
    <rotate>
      <yield id="subject" />
      <yield id="object" />
    </rotate>
    <together>
```

```
      <yield id="subject" />
      <yield id="object" />
    </together>
  </rule>
```

In this example, the *subject* and *object* of the predicate would be put together, and the *subject* will be diverted onto *object*.

4 Implementation

The Utkus prototype under discussions was written on the Ruby programming language:

1. Link Grammar for Russian syntactic parser [6] is used because of its availability and easy parseable format;
2. Only verb phrases and related noun phrases are extracted from the dependency tree of each sentence of the source text. These syntactic groups are mapped into the SPO–triplets;
3. Ontology is defined in the RDF/XML format using the Protegé editor;
4. There are only synsets in our Russian dictionary [9]: no hyponyms, etc;
5. Gallery is composed by sprites from The Noun Project [7] collection. These sprites are cropped, rasterized, and associated with noun synsets;
6. Final rendering is performed using GD2 library in form of PNG raster images of 640×480.

As example, there are four images that been generated by Utkus system. With a view of place economy, these images been cropped. These images (Fig. 4(a), 4(b), 4(c), 4(d)) are correspond to texts:

1. A man has fallen into the fire[2];
2. Several houses[3];
3. There are a man and a woman in the house[4];
4. A certificate, a bear, a rain[5].

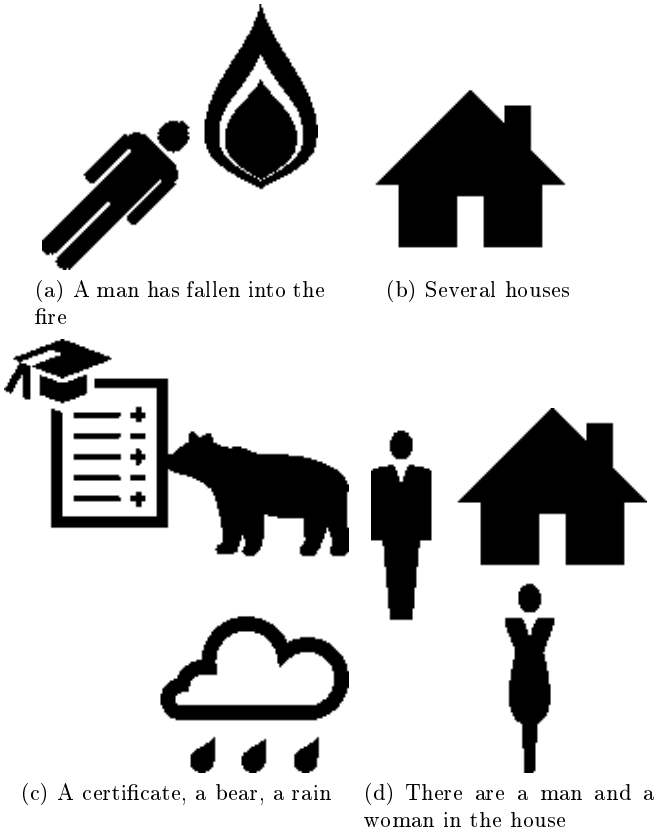
It should be noted that Utkus system is unable to represent plural words (Fig. 4(b)) at this moment.

5 Conclusion

We have presented the approach to organize the TTP system information resources. This approach provides loose coupling of ontology, thesaurus, gallery and depiction rules.

Main advantages of this approach are:

[2] Человек упал в огонь, in Russian.
[3] Несколько домов, in Russian.
[4] В доме находились мужчина и женщина, in Russian.
[5] Аттестат, медведь, дождь, in Russian.



(a) A man has fallen into the fire

(b) Several houses

(c) A certificate, a bear, a rain

(d) There are a man and a woman in the house

Fig. 4. Depiction of the texts

1. Simplicity of development and modification all the information resources that are used by TTP system:
   - Ontology can be modified with any available ontology editor (e.g., Protegé);
   - Depiction rules can be edited with any text editor, or any XML editor;
   - Thesaurus and gallery data can be modified as any data in relational database (in our implementation, PostgreSQL is used).
2. RDF/XML ontology allows one to reuse these resources in other applications and domains;
3. Verification instruments (such as inference systems) can help us to control the quality of information resources.

Figures 4(a), 4(b), 4(c) are produced during testing our Utkus TTP system under development. The Utkus TTP system is based on this approach.

5.1 Future Work

We have several reasons for future work:

1. To switch to the full–featured thesaurus to unify the thesauri resources (e.g., Russian WordNet [8]);
2. To enhance the linguistic analysis subsystem to handle such parts of speech as adjectives, pronouns, numerals, etc;
3. To solve the problem of predicate ambiguation [3] when generating the semantic representation;
4. To perform experiments on the Utkus prototype and make changes in the system components, if necessary.

Acknowledgements. Authors would like to thank the Institute of Mathematics and Mechanics UrB RAS for the provided computer equipment.

References

1. Åkerberg, O., Svensson, H., Schulz, B., Nugues, P.: CarSim: an automatic 3D text-to-scene conversion system applied to road accident reports. In: Proceedings of the 10th Conference on European Chapter of the Association for Computational Linguistics–Volume 2. pp. 191–194. Association for Computational Linguistics (2003)
2. Coyne, B., Sproat, R.: Wordseye: an automatic text-to-scene conversion system. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques. pp. 487–496. ACM (2001)
3. Fomichov, V.: A comprehensive mathematical framework for bridging a gap between two approaches to creating a meaning-understanding web. International Journal of Intelligent Computing and Cybernetics 1(1), 143–163 (2008)
4. Goldberg, A., Rosin, J., Zhu, X., Dyer, C.: Toward text-to-picture synthesis. In: NIPS 2009 Mini-Symposia on Assistive Machine Learning for People with Disabilities (2009)
5. Li, H., Tang, J., Li, G., Chua, T.: Word2image: Towards visual interpretation of words. In: The 16th ACM International Conference on Multimedia (2008)
6. Link Grammar for Russian, `http://slashzone.ru/parser/`
7. NounProject, `http://thenounproject.com`
8. Russian Wordnet, `http://www.wordnet.ru`
9. Russian Language Dictionaries, `http://speakrus.ru/dict/index.htm`
10. Utkus, `http://utkus.eveel.ru`
11. Yamada, A., Yamamoto, T., Ikeda, H., Nishida, T., Doshita, S.: Reconstructing spatial image from natural language texts. In: Proceedings of the 14th Conference on Computational Linguistics–Volume 4. pp. 1279–1283. Association for Computational Linguistics (1992)
12. Yoshii, M., Flaitz, J.: Second language incidental vocabulary retention: The effect of text and picture annotation types. CALICO journal 20(1), 33–58 (2002)
13. Zhu, X., Goldberg, A., Eldawy, M., Dyer, C., Strock, B.: A text-to-picture synthesis system for augmenting communication. In: Proceedings of the National Conference on Artificial Intelligence. vol. 22, p. 1590. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999 (2007)

Evaluating the Quality Level of Projects, Authors and Experts

Vorobev Alexandr

Moscow State University, Faculty of Mathematics and Mechanics, Department of Probability Theory
Witology
alexandr.vorobyev@witology.com
http://mech.math.msu.su/probab/index-e.html
http://witology.com/

Abstract. There are some situations when we need to compare work results of different people. For example, it may be an announced tender, a grants' distribution or a competition in poetry. And sometimes it is allowed for one participant to present more than one work. It is extremely actual when we want not only to identify the winner but also to get a lot of effective projects. We don't want to bound the author's creativity therefore. Then having expert evaluations we must obtain integral evaluation for each work. How to do this? Also it may be useful to evaluate author's level on the basis of his works' evaluations. Moreover, we may want to evaluate expert's level on the basis of his evaluations. In this paper we solve these problems by constructing a Bayesian network model for such a competition and applying a maximum a posteriori estimation method to it.

Keywords: Bayesian network, maximum a posteriori estimation, maximal likelihood method, expert evaluations, expert competency, parameter evaluation.

1 Model description

Let us take a competition between projects, which allows an author to present more than one project. The participants present their projects and experts evaluate these projects, not more than one evaluation given by one expert for one project.

The aim is to estimate level of each project, level of each expert and level of each author on the basis of expert evaluations.

We introduce the following model of the competition.

1) Each author has his level C(Creator) - degree of his ability to create the effective projects.

2) Each expert has his level E (Evaluator) - degree of his ability to give for a project the proper evaluation.

3) Each project has its level L(Level) - degree of its effectiveness. This level is random value and appears when its author creates this project. We also assume that L has a distribution depended on its author level C - $Rand_L(C)$.

4) Each project evaluation e given by an expert is also a random value and its distribution depends on the project level and on the expert level - $Rand_e(L, E)$.

5) The number of projects presented by author is independent from his level.

Distributions $Rand_L(C)$ and $Rand_e(L, E)$ may be fixed if appropriate parameters are fixed or may have another parameters for which we don't have prior exact values. Then in general case we have distributions $Rand_L(C, P)$ and $Rand_e(L, E, Q)$ where P and Q are vectors of parameters.

Thus we can formulate the following problem:
Given 3-dimensional vector of project evaluations

$$(e_{i,j,k})_{i=1..n,j=1..n,k=1..m_j},$$

where $e_{i,j,k}$ - the evaluation given by the expert i to the k^{th} project of the author j (we number projects of one author beginning from the one);

m_j - number of the projects of the author j.

The evaluations of the following values have to be found:

$(C_j)_{j=1..n}$ - vector of the author levels;

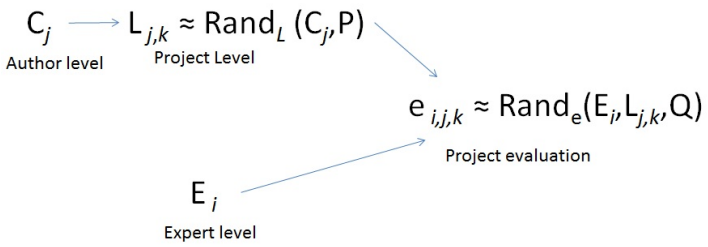
$(E_j)_{j=1..n}$ - vector of the expert levels;

$(L_{j,k})_{j=1..n,k=1..m_j}$ - 2-dimensional vector of the project levels;

P - vector of the distribution parameters $Rand_L$;

Q - vector of the distribution parameters $Rand_e$.

Thus we have constructed a Bayesian network as a model for our competition.



2 Method of problem solving

It may be interesting to find posterior distribution for unknown parameters in this model too, but here we are seeking for parameters' estimates only. The standard method for this problem is the maximum a posteriori estimation method. We do not have any special suggestions about the parameter distribution. Therefore we consider the prior parameter distribution to be uniform. In this case maximum a posteriori estimates coincide with maximal likelihood estimates.

It means that the estimate of a parameters' vector equals such a value, that conditional probability of the fact, that the expert evaluations are equal to their observed values

$$(e_{i,j,k})_{i=1..n,j=1..n,k=1..m_j},$$

is maximal.

Up to this moment all our values could lay in any set. But we need to specify the kind of these sets to write the formulae. We consider the finite sets due to the following reason. It is well known (see [1]) that it is natural for an expert to say which of two projects is better relative to a property or group of properties but it is not natural for him to give quantitative evaluation of project. A finite set for evaluations is closer to the theory than a continuous or discrete infinite set.

Then it is natural for evaluations $e_{i,j,k}$ and project levels $L_{j,k}$ to lay in the same set. Then evaluations can be considered as project level evaluations.

Let us also consider the values C, E, P, Q to be discrete. It makes the results more simple to be interpreted (values C, E) and makes the narrative more simple.

To simplify our figures we drop indexes in the names of vectors if they may have all the possible values. So we write

$(A_{i,j,k})_{k=1..m_j}$ instead of $(A_{i,j,k})_{i=1..n,j=1..n,k=1..m_j}$

and $(A_{i,j})$ instead of $(A_{i,j})_{i=1..n,j=1..n}$.

Also we mean that the index under the operator of sum or multiplication runs through the whole set of its possible values.

Then maximum a posteriori estimates have the following form:

$$\begin{aligned}
&((\widetilde{L_j}),(\widetilde{C_j}),(\widetilde{E_j}),\widetilde{P},\widetilde{Q}) = \\
&= arg \max_{(L_{j,k})_{k=1..m_j},(C_j),(E_j),P,Q} P((e_{i,j,k})_{k=1..m_j}|(L_{j,k})_{k=1..m_j},(C_j),(E_j),P,Q) = \\
&= arg \max_{(L_{j,k})_{k=1..m_j},(C_j),(E_j),P,Q} \prod_{i,j,k}(P(e_{i,j,k}|E_i,Q,L_{j,k})P(L_{j,k}|C_j,P)) = \\
&= arg \max_{(L_{j,k})_{k=1..m_j},(C_j),(E_j),P,Q} \sum_{i,j,k}(lnP(e_{i,j,k}|E_i,Q,L_{j,k}) + lnP(L_{j,k}|C_j,P))
\end{aligned} \tag{1}$$

It is a problem of discrete optimization which is NP-complete. It means that the complexity of computations of an exact answer is huge for the big number of parameters. Hence we have to use optimization methods which give an approximate answer.

There are different optimization algorithms for this problem, for example the expectation–maximization algorithm, the gradient method, the analogue of belief propogation algorithm. The choice of such a method depends on the model. Probably we should try different methods to understand which is better in our case.

3 Model specification

Here we specify the distribution functions $Rand_L(C, P)$ and $Rand_e(L, E, Q)$. As we decided random values $L_{j,k}$ and $e_{i,j,k}$ are distributed on the same finite set. This set has to be small because of the big complexity of the optimization problem solving. In our data this set is the set if integers from the interval $[-3, 3]$.

Then we consider the following assumptions to be natural:

1) The mathematical expectation of the expert evaluation $e_{i,j,k}$ equals the true level of this project $L_{j,k}$.

2) Its volatility depends on the expert level: the higher the expert level, the less the volatility. It means if we give the big set of projects of the same level to the expert (even bad) the average of his evaluations will be close to this true level. But the better the expert the less standard deviation of his evaluations will be.

It is the best practice to use normal distribution in the models due to the existence of the central limit theorem. But we have discrete sets of variables. Hence we use the value of normal density divided by normalizing constant as the probability function value for each discrete point. Then the probability function has the bell curve shape.

Now we have to define the volatility as function of the expert level. Let us take volatility to be equal to $\frac{q}{E_i}$. We need constant q because E_i is bounded (it lays in the finite set), but we don't know the bound for volatility - it depends on the data. And we suppose that E_i is integer from the interval [0,3]. The vector Q consists of the only value q in this case.

Thus value $e_{i,j,k}$ is distributed according to the law

$$P(e_{i,j,k} = a | L_{j,k}, E_i) = \frac{e^{-\frac{(a-L_{j,k})^2 E_i^2}{2q^2}}}{\sum\limits_{b=-3}^{3} e^{-\frac{(b-L_{j,k})^2 E_i^2}{2q^2}}}, a = -3..3 \tag{2}$$

We define the same discretized normal distribution for the project level $L_{j,k}$ where the mathematical expectation equals the author level C_j and the volatility equals the constant parameter p. It means the higher the author level the higher the average level of his projects. Volatility is constant because we don't know any facts about the dependence between it and the author level.

Then C_j is the integer from the interval [-3,3] and the vector P consists of the only value p in this case.

Thus value $L_{j,k}$ is distributed according to the law:

$$P(L_{j,k} = a | C_j) = \frac{e^{-\frac{(a-C_j)^2}{2p^2}}}{\sum\limits_{b=-3}^{3} e^{-\frac{(b-C_j)^2}{2p^2}}}, a = -3..3 \tag{3}$$

4 Relation with other problems

There is a standard problem which can be formulated in the terms of this paper as the following: it is necessary to evaluate the project levels and the expert levels on the basis of the evaluations given by these experts to these projects.

Except our attempt to evaluate the author level there is the principal difference between our approach to this problem and standard approach. In our problem we have authors of the projects. And we think that the projects of one author will more probably be similar in their levels than the projects of two different authors. This fact underlies our model.

The standard approach (see [4]) consists in applying an iteration procedure. On each step for each expert we measure the closeness between his project evaluations and the current weighted-average project evaluations where the weight of the expert evaluation equals the expert level. Then the expert levels are reobtained on the basis of this closeness and so on.

5 Conclusions

The paper presents a new model of a competition. Within this model a maximum a posteriori estimation can be applied to estimate the project levels, the project author levels and the expert levels on the basis of the evaluations given by these experts to these projects.

Notice that the presented algorithm does not have any requirements for the existence of the evaluation given by the definite expert to the definite project or for the number of evaluations per one project. Another question is that the more evaluations we have the more exact the results of the algorithm are.

We are going to apply this method to the data in the nearest future. Then we will be able to compare this algorithm with the standard approach in estimation of $((L_{j,k})_{k=1..m_j}, (E_j))$.

Also we are going to consider this model when the authors are the experts at the same moment. Then in our model we can take into account a correlation between participant level as an author and his level as an expert. Or we can evaluate this correlation on the basis of the results of this method if we don't take it into account in the model.

References

1. Orlov, A.: Expert Evaluations. Moscow (2002)
2. Lemeshko, B., Denisov, V., Postovalov, S.: Applied Statistics. Rules of Testing of Consensus between Sampling and Theoretical Distributions. Methodical recommendations. Part I. Chi-squared-type Tests (in Russian). NSTU, Novosibirsk (1998)
3. Orlov, A.: About Tests of Consensus with Parametrical Distribution Family. Zavodskaya laboratoriya. Diagnostika materialov/ 63,№5, 49-50 (1997)
4. Vidyapin, V.: Bachelor of Economics. Vol. 2. Triada, Moscow (1999)
5. Heckerman D.: A Tutorial on Learning With Bayesian Networks. Technical Report MSR-TR-95-06, Microsoft Research, Advanced Technology Division (1996)

Author Index