DOI: 10.15439/2016F571
ACSIS, Vol. 9. ISSN 2300-5963

Recognition of Compound Objects Based on Network of Comparators

Łukasz Sosnowski
Systems Research Institute, Polish Academy of Sciences
Newelska 6, 01-447 Warsaw, Poland
Dituel Sp. z o.o.
Ostrobramska 101 lok. 206, 04-041 Warsaw, Poland
Email: Lukasz.Sosnowski@ibspan.waw.pl

Marcin Szczuka
Institute of Informatics, University of Warsaw
Banacha 2, 02-097 Warsaw, Poland
Email: szczuka@mimuw.edu.pl

***Abstract*—This paper proposes a methodology for compound objects' recognition based on comparators and comparator networks. The methodology is supported by a collection of techniques and algorithms for construction and learning of comparator networks. Formal description of the methodology is accompanied by selected examples of its application in real-life problems. The described methodology has been implemented as a software library and may be used for a variety of future applications.**

I. Introduction

IN THIS position paper we propose an approach to computational tasks which involve management, identification, classification and modelling of data objects that are intrinsically complex, compound and described by means of various types of information. Our proposed approach is based on *comparators* and *comparator networks*. A comparator is a basic, relatively simple computational element that models local similarity between objects of possibly compound nature, such as phenomena, processes or sub-systems. They can be arranged into comparator networks that are capable of aggregating and summarizing local information about similarity into a global measure of proximity between data objects. This versatility provides the ability to solve the problem at hand by decomposing it into simpler, localised steps that can be processed quickly. Then, local results from elementary units can be aggregated and processed in the network producing the overall, possibly complex and multi-dimensional final result.

The overall purpose of both the comparator and the network is to generate the vector of numerical values that indicates the levels of various similarities between the object provided as an input and the already processed, well-known *reference objects*. The reference set, i.e. a set consisting of reference objects, can be regarded as the underlying knowledge base. For example, if we attempt to make an assessment of the situation on the road and associated risks at the moment we may be provided by the comparator network with assessment stating how close our situation resembles the previously seen situations. The key advantage of the comparator network is the fact that it is able to adapt to various levels and contexts that occur in data. In comparison with the other general, similarity-based approaches, such as Case Based Reasoning (CBR, see [1]), the one we propose is more flexible and provides more possibilities for fine-tuning the solution.

The approach based on comparators and comparator networks is a field-proven technology. Several uses in both Research and Development (R&D) and production software systems have been reported. The applications of comparator networks range from shape and character recognition in images to information retrieval in natural language text corpora (see [2]–[5]). A big incentive for using this technology is the existence of software library in Java that can be readily adapted for the needs of potential user. The library, as well as the accompanying services[1] are publicly available (see [6]).

II. Comparators

Compound object comparator is a construct dedicated to processing complex objects represented as data entities. We denote such a construct by com^{ref}. Such comparator can be identified with a function:

$$\mu^{ref}_{com} : X \times 2^{ref} \to [0,1]^{ref}, \qquad (1)$$

where $X \subseteq U$ is a set of input objects to be compared and ref is a set of reference objects that we infer the similarity from. $[0,1]^{ref}$ denotes a space of vectors $\vec{v}$ of dimension $|ref|$, where each i-th coordinate in $v[i] \in [0,1]$ corresponds to an element $y_i \in ref$, $ref = \{y_1, ..., y_{|ref|}\}$. We will further call ref a *reference set*, while each $Y \subseteq ref$ will be referred to as *reference subset*. Additionally, $a(x)$ is a function that provides a representation of an object $x \in X$ w.r.t. a given attribute a. This representation is then used by the comparator while processing x. Similarly, each reference object $y \in Y$ is processed using its representation $a(y)$ for a given attribute a. If we are given an ordering on elements of reference set ref, i.e, $ref = \{y_1, \ldots, y_{|ref|}\}$ we can represent the function corresponding to the comparator as:

$$\mu^{ref}_{com}(x, Y) = Sh(F(\vec{v})), \qquad (2)$$

where Sh is a (result) *sharpening function*, F is a function responsible for filtering the result before sharpening (defuzzyfication) and $\vec{v}$ is a vector of dimension equal to the cardinality

[1]Comparators at Dituel http://www.dituel.pl/Service,908/Comparators,1136/index.html

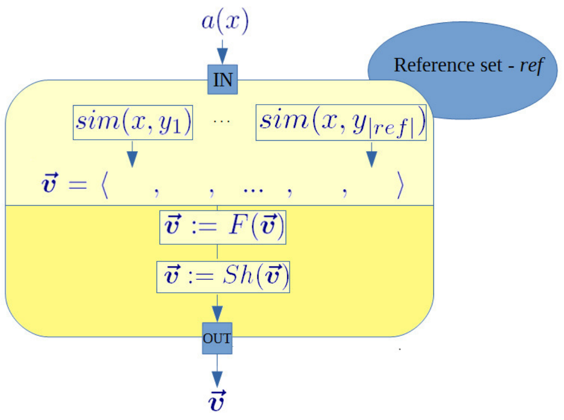

Fig. 1. Block diagram of the comparator's internal structure. The structure consists of two main layers: concurrent – responsible for the calculation of values for the initial proximity vector; sequential – responsible for processing the values from the previous stage. Blocks marked with *sim* are charged with calculation of unit similarity values. The internal structure of a such similarity block is described in Fig. 2 using UML.

of ref, composed of proximity (similarity, closeness) values between the object x and each of the reference objects in ref. Typically, F is based on combination of some standard, idempotent functions such as min, max, top, etc. [7] When Y is a proper subset of ref the positions in $\vec{v}$ corresponding to $y_i \notin Y$ are filled with zeros. Non-zero elements of $\vec{v}$ determine the degree of similarity (proximity) between the object x in question and each element of reference subset Y. Hence, the *proximity vector* can be defined as:

$$\vec{v}[i] = \begin{cases} 0 & y_i \notin Y \\ sim(x, y_i) & y_i \in Y \end{cases} \tag{3}$$

The value of similarity $sim(x, y)$ used above is calculated by means of a fuzzy relation [8] combined with additional mechanisms, as described in the next section.

III. Architecture of a Comparator

A comparator consists of two main layers (stages) – concurrent and sequential. The first of those layers is responsible for the calculation of values of proximity vector's coordinates for $x \in X$, $y_i \in ref$, with use of similarity function:

$$sim(x, y_i) = \begin{cases} 0: & Exc^{ref}_{Rules_i}(x) = 1 \vee y_i \notin Y \\ t_h(\mu(x, y_i)): & otherwise \end{cases}, \tag{4}$$

where $Y \subseteq ref$, t_h is a threshold function given as

$$t_h(z) = \begin{cases} 0: & z<p \\ z & z \geq p \end{cases}, p \in [0, 1]. \tag{5}$$

The value of p corresponds to the lowest acceptable similarity, μ is the basic similarity function, $Exc^{ref}_{Rules_i}$ is the function associated with prohibitive rules and i is an index of the coordinate of proximity vector for which the similarity is derived.

Similarities for different y_i can be calculated concurrently because they do not depend on each other. The internal, layered structure of a comparator is shown in Fig. 1. While each coordinate of the vector $\vec{v}$ is calculated independently, the final calculation of the value of the function (4) has to be performed in a sequence for a given pair of objects (x, y_i). This sequence of operations is illustrated in Fig. 2 in form of an UML activity diagram [9]. The processing in the first layer ceases when all coordinates of the proximity vector are derived. Only then can the comparator activate the next layer.

The processing in the comparator's second layer is performed sequentially. Hence, operations such as filtering by means of $F(\vec{v})$ and sharpening of proximity vector with $Sh(\vec{v})$ are performed one-by-one. This sequence yields the vector given by (3) as the final result.

If we take a wider look at the comparator for complex objects we may notice that if all the notions introduced above are composed, it can be expressed as:

$$\mu^{ref}_{com}(x, Y) = Sh(F(\langle sim(x, y_1), \ldots, sim(x, y_{|ref|})\rangle)) \tag{6}$$

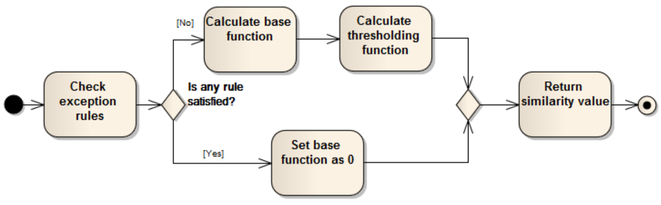

Fig. 2. An UML activity diagram of the *sim* block for a comparator calculating similarity between the input and reference object.

The notions introduced in sections II and III provide a systematic overview of the concept of comparator as a basic computing unit in systems aimed at supporting decisions, recognising patterns and at similar tasks. A single comparator (unit) is capable of measuring the level of similarity (proximity) of a given object to the reference set with regard to the attribute for which it was created. Comparator itself can be regarded as a decision support system, albeit a very rudimentary one. It can only handle very straightforward tasks associated with modelling a simple similarity. In order to use comparators in a more complicated situation one has to combine several of them into a network. Such a network is then an example of similarity-based inference system capable of modelling complex (similarity) relations occurring in data [2].

IV. Components of a Comparator Network

The operation of a comparator network can be interpreted as a calculation of a function:

$$\mu^{ref_{out}}_{net} : X \to [0, 1]^{|ref_{out}|}, \tag{7}$$

which takes the input object $x \in X$ as an argument and ref_{out} is a reference set for the network's output layer. The target set (codomain) of $\mu^{ref_{out}}_{net}$ is the space of proximity vectors. As in the previous situation, the proximity vector from the target space will be denoted by $\vec{v}$. Such a vector encapsulates information about similarities between a given input object x and objects from the reference set ref. Similarly to the case of a single comparator, by ordering the reference set, i.e. taking $ref = \{y_1, ..., y_{|ref|}\}$, we get the value network's function of:

$$\mu_{net}^{ref}(x) = \langle SIM(x, y_1), ..., SIM(x, y_{|ref|})\rangle, \qquad (8)$$

where $SIM(x, y_i)$ is the value of *global similarity* established by the network for an input object x and a reference object y_i. Global similarity depends on partial (local) similarities calculated by the elements of the network (unit comparators). Through application of aggregation and translation procedures at subsequent layers of the network these local similarities are ultimately leading to the global one.

A. Layers in Comparator Network

Each comparator network is composed of three types of layers: input, intermediate (hidden/internal) and output. A given network may have several internal layers [10]. Layer consists of comparators that are grouped together by the common purpose of processing a particular piece of information (attributes) about the object in question. Each layer contains a set of comparators working in parallel and a specific translating/aggregating mechanism. The translating and aggregating mechanisms are necessary to facilitate the flow of information (similarity vectors) between layers. As sets of comparators in a particular layer corresponds to a specific combination of attributes, the output of the previous layer has to be aggregated and translated to fit the requirements. This is done by elements called translators and aggregators, respectively. The translator converts comparator outputs to information about reference objects that would be useful for the next layer. The role of the aggregator is to choose the most likely outputs of the translator, in case there was any non-uniqueness in assigning information about input objects to comparators. The operation of a layer in the comparator network can be represented as a mapping:

$$\mu_{layer}^{ref} : X \to [0, 1]^{ref_l}, \qquad (9)$$

where $x \in X$ is an input object and ref_l is the reference set for the layer.

Within a given layer only the local reference sets associated with comparators in that layer are used to establish (local) similarities. However, through aggregation and translation these local similarities become the material for synthesis of the output similarity and reference set for the layer. This synthesis is based on a translation matrix, as described in [11]. Function (9) is created as a superposition of: comparator's function (1), local (layer) aggregation function and translation. Local translation operation is responsible for filtering the locally aggregated results.

The input and internal (hidden) layers in the comparator network contain comparators with function (1) together with translators and local aggregators. The output layer contains the global aggregator responsible for returning the final result. The components of the comparator network are described briefly below. For details please refer to [11], [12].

B. Local Aggregator

Aggregators are a mandatory part of the network responsible for the synthesis of the results obtained by comparators. Aggregators are functions that operate on partial results of comparators. In the simplest case the network only needs a single *global aggregator* in the output layer. However, in the other network architectures it is included in other layers as well, in form of a local aggregator.

The local aggregator processes partial results of the network at the level of a given layer. The aggregator's operation depends on the type of reference objects and the output of comparators. It can be represented as:

$$f_{agg}^{ref_l} : [0, 1]^{ref_1} \times \ldots \times [0, 1]^{ref_k} \to [0, 1]^{ref_l}, \qquad (10)$$

where k is the number of comparators in a given layer l, i.e. the number of inputs in the aggregating unit (local aggregator). ref_l is the output (resulting) reference set for layer l composed by means of the *composition rules* from the reference sets ref_i $(i = 1, \ldots, k)$ used by comparators in layer l.

C. Translator

The translator is a network component associated with the adaptation of results of one layer to the context of another layer (the one to be fed with). In other words, this element expresses the results of the previous layer (their reference objects) in reference objects of the current one. It uses reference objects of the next layer, taking into account the relationships between the objects of both layers [13]. The translator is defined by means of the translation matrix:

$$M_{ref_l}^{ref_k} = [m_{ij}], \qquad (11)$$

where $i \in \{1, \ldots, m\}$, $j \in \{1, \ldots, n\}$ for m and n denoting cardinality of ref_k and ref_l, respectively. The matrix $M_{ref_l}^{ref_k}$ defines the mapping of objects in the set ref_k onto objects in the set ref_l. In practice ref_k is just a union of reference sets for all comparators in a given layer and ref_l is the target reference set. Values in the matrix are within $[0, 1]$.

D. Projection Module

This network unit appears in selected layers whenever there is a need for selecting a subset of coordinates (project the vector onto subspace) in proximity vector that will be further used in calculations. The selection of a particular coordinate may be based on its value (above/below threshold) and/or on the limitations regarding the number of coordinates that can be preserved. For the i-th coordinate in the proximity vector the projection can be the following:

$$\mu_{proj}(v[i]) = \begin{cases} \vec{v}[i] & projection(\vec{v}[i]) = 1 \\ 0 & projection(\vec{v}[i]) = 0 \end{cases} \qquad (12)$$

where $i \in \{1, \ldots, |ref|\}$ and $projection(a)$ for $a \in [0, 1]$ is a function of the form:

$$projection : [0, 1] \to \{0, 1\}, \qquad (13)$$

The function $projection$ is the actual selecting mechanism. It decides whether a given coordinate is set at 0 or not. This function can be defined as a threshold, maximum, ranking function, etc.

E. Global Aggregator

The global aggregator is a compulsory element of the output layer. Unlike local aggregators, which process results within a single layer, the global one may process values resulting from all layers at the same time. In the simplified, homogeneous case, when all layers use exactly the same reference set, the global aggregator may be expressed by:

$$\mu_{agg}^{ref_{out}} : \left([0,1]^{ref}\right)^m \rightarrow [0,1]^{ref_{out}}, \tag{14}$$

where m is the number of **all** comparators in the networks, i.e. the number of inputs to the global aggregator.

In the more complicated, heterogeneous case, the sets in subsequent layers and comparators may differ. In this case the aggregator constructs the resulting (global) reference set ref_{out} in such a way that every element $y \in ref_{out}$ is decomposed into y_1 in reference set ref_1, y_2 in reference set ref_2 and so on, up to y_m in reference set ref_m. For a given input object $x \in X$ the value of similarity between x and each element of in $ref_1, \ldots, ref_m$ is known, as this is the output of the corresponding comparator. To obtain the aggregated result we use:

$$\mu_{agg}^{ref_{out}} : [0,1]^{|ref_1|} \times \ldots \times [0,1]^{|ref_m|} \rightarrow [0,1]^{ref_{out}} \tag{15}$$

Note, that formula (15) is similar to the one for local aggregator (10). The essential difference is in the fact that the local aggregator is limited to a subset of comparators contained in a given layer, while the global one looks at all comparators in the network.

With all the definitions of units the comparator network can be expressed as a composition of mappings in subsequent layers:

$$\mu_{net}^{ref_{out}}(x) = \mu_{layer-out}^{ref_{out}}(\mu_{layer-int}^{ref_{k-1}} \cdots (\mu_{layer-in}^{ref_1}(x)) \ldots), \tag{16}$$

where ref_i stands for the reference set corresponding to layer i and ref_{out} is the reference set for the network as a whole. The general scheme of the comparator network is shown in figure 3.

V. Selected applications of comparator networks

Two examples of successful application of the comparator network methodology are presented. The two applications presented originate from rather unrelated areas. The purpose of showing them here is to highlight the versatility of comparator networks as a tool for dealing with real-life challenges. The first of the examples presented is associated with recognition and classification of texts in (quasi-)natural language. The second example relates to risk management in CBR systems [1]. The two examples presented are just a selection from a range of applications that were reported. For more examples refer to [2]–[5].

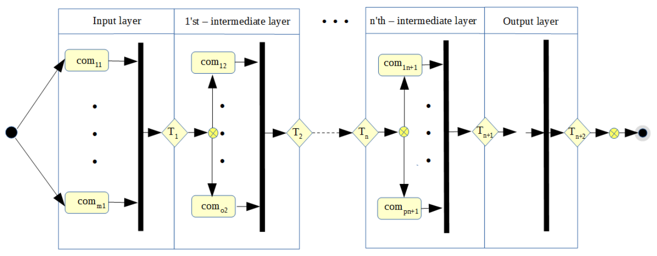

Fig. 3. General scheme of a comparator network in UML-like representation. Notation: com_{ji}- comparators, T_j- translators. Symbols: oval – comparator, thick vertical line – aggregator, rhombus – translator, encircled cross – projection module.

A. Classification of References in Scientific Articles

This application was conceived as an answer to a need defined in the course of the SYNAT project[2]. The overall goal of this project was to construct an integrated network platform for the storage, retrieval, management and delivery of digital information in areas of science and technology [14].

One of the tasks within the scope of SYNAT was to design and implement a sub-system for searching within repositories of scientific information (articles, biographical notes, etc.) using their semantic content (SONCA). The information contained in documents is by nature compound and relations discovered between their parts and other entities in data are crucial to preserve. One of the compound parts is the reference part in form of unstructured texts. The task is to identify the part of text that is a reference and recognise (name, classify) its parts, so that they can be given a semantical context which can greatly improve information retrieval and management. The result should be expressed by means of (sub-)objects classified to the already known classes such as authors, titles, journals, the publication date, and so on. For example, the following text found in the document: "*Sosnowski, L., Slezak, D.: Networks of Compound Object Comparators. In: Proc. of FUZZ-IEEE 2013 (2013)*" might be resolved by assigning the pattern ATRY to its parts, where: A stands for authors, T – title, R – reference volume (pRoceedings), and Y – the publication date (year).

TABLE I
Selected characteristics for similarities between the closest objects.

Measure	Value
Max	1.00
Min	0.50
Average	0.85
Median	0.87
Standard deviation	0.12
Variance	0.02

The whole process is divided into several stages: preprocessing, parsing and classification. The first stage is responsible for cleaning the data, sifting unnecessary data from the text

[2]SYNAT (www.synat.pl) was a major national R&D project financed by NCBiR as a part of the strategic research initiative.

TABLE II
TOP LEFT: BEST RESULTS FOR EXPERIMENT 1, TOP RIGHT: WORST RESULTS FOR EXPERIMENT 1, BOTTOM LEFT: BEST RESULTS FOR EXPERIMENT 2, BOTTOM RIGHT: WORST RESULTS FOR EXPERIMENT 2. NOTATION: P_1 - PRECISION FOR EXPERIMENT 1, R_1 - RECALL FOR EXPERIMENT 1, $F1_1$ - F1-SCORE FOR EXPERIMENT 1, P_2 - PRECISION FOR EXPERIMENT 2, R_2 - RECALL FOR EXPERIMENT 2, $F1_2$ - F1-SCORE FOR EXPERIMENT 2.

Pattern	P_1	R_1	$F1_1$	P_2	R_2	$F1_2$
ATR	1.00	1.00	1.00	0.75	1.00	0.86
RY	1.00	1.00	1.00	1.00	1.00	1.00
ATJVY	1.00	1.00	1.00	1.00	0.80	0.89
AT	1.00	0.98	0.99	0.61	0.92	0.73
ATRYP	1.00	0.93	0.96	1.00	0.73	0.83
ATVPYD	0.91	0.98	0.95	0.92	0.84	0.86
ATJVPYD	1.00	0.90	0.94	0.98	0.71	0.81
ATJVPY	1.00	0.86	0.92	1.00	0.58	0.72
AJVPYD	0.86	1.00	0.92	0.86	1.00	0.92
ATVPY	1.00	0.85	0.91	1.00	0.60	0.75

Pattern	P_1	R_1	$F1_1$	P_2	R_2	$F1_2$
ATC	0.33	0.67	0.44	0.00	0.00	0.00
ATYATYP	0.60	0.43	0.50	1.00	0.43	0.60
AYTRY	0.43	0.60	0.50	1.00	0.60	0.75
ATPYC	0.54	0.80	0.64	0.60	0.60	0.60
ATJVYP	1.00	0.50	0.65	0.50	0.25	0.33
ATVY	0.60	0.75	0.67	1.00	0.75	0.86
ATVYPR	0.56	0.83	0.67	1.00	0.50	0.67
ATJPY	0.94	0.58	0.70	0.89	0.60	0.71
ATRPYC	0.71	0.77	0.73	1.00	0.77	0.86
ATAT	0.57	1.00	0.73	0.00	0.00	0.00

Pattern	P_1	R_1	$F1_1$	P_2	R_2	$F1_2$
RY	1.00	1.00	1.00	1.00	1.00	1.00
ATRY	0.89	0.88	0.86	1.00	0.88	0.93
ATRPY	0.93	0.90	0.90	0.99	0.88	0.92
AJVPYD	0.86	1.00	0.92	0.86	1.00	0.92
ATRJPY	0.83	0.83	0.83	1.00	0.83	0.91
ATRPYCD	0.84	0.85	0.84	0.97	0.85	0.91
ATPYD	0.83	1.00	0.91	0.83	1.00	0.91
ATY	0.88	0.91	0.87	0.90	0.90	0.90
ATJVY	1.00	1.00	1.00	1.00	0.80	0.89
ATRP	1.00	0.75	0.86	0.88	0.88	0.88

Pattern	P_1	R_1	$F1_1$	P_2	R_2	$F1_2$
AYTJP	1.00	0.70	0.82	0.00	0.00	0.00
ATC	0.33	0.67	0.44	0.00	0.00	0.00
ATAT	0.57	1.00	0.73	0.00	0.00	0.00
ATJYR	1.00	0.60	0.75	0.00	0.00	0.00
AYT	0.94	0.87	0.87	0.13	0.13	0.13
ATJVYP	1.00	0.50	0.65	0.50	0.25	0.33
AYTP	0.73	0.85	0.76	0.37	0.30	0.33
ATYR	1.00	0.75	0.86	0.50	0.50	0.50
ATYP	0.83	0.69	0.75	0.61	0.47	0.53
ATPYC	0.54	0.80	0.64	0.60	0.60	0.60

structure and making it clear (e.g. lowercase, trimmed, etc.). It contains the prefix cleaning procedure, which eliminates unwanted prefixes e.g. "*[12] D., E., Willard, New trie data structures which support very fast search operations, Journal of Computer and System Sciences, v.28 n.3, p.379-394, June 1984 u00A0[doi>10.1016/0022-0000(84)90020-5]*" is an input text. The mentioned procedure cuts the part "*[12]*". The main idea of this stage is to dispose of all signs which can interrupt the further parsing or comparing process. The second operation is to replace all quotation marks which are not necessary but often interfere with the results.

The results reported below were obtained from two numerical experiments. For these experiments 400 examples of references (pieces of text) were drawn randomly from the corpus of scientific texts. The first experiment (Experiment 1) involved generating patterns for sub-objects (fragments of reference text) and composing (concatenating) them into a pattern for the whole reference. In the second experiment (Experiment 2) the pattern resulting from Experiment 1 was additionally compared with a reference set consisting of all possible patterns. The reference sets used were taken from the collection of already processed and classified articles in the SONCA subsystem [14]. Each of these reference objects had a clearly defined pattern it belonged to.

The experimental data were split into the training and testing parts containing 132 and 268 cases, respectively. The training sample was used to build the (local and global) reference set for the comparator network and for tuning of network's parameters. The test sample was used to assess the quality of classification (pattern identification). The experiment involved two main steps: (i) generation of reference patterns using similarities between sub-objects and (ii) comparison of generated patterns with the reference set of known (certain) patterns of bibliographic references.

Both experiments involved the use of a comparator network designed for this purpose. The network used local similarities to build (infer) the overall result. The particular network used had the most of the local similarity values at a relatively high level, within the $[0.8, 0.85]$ range. The other parameters of the network were set to default values. In particular, the threshold parameter p was set to middle (0.5) and the results were aggregated using simple averaging, i.e., assigning the same weights to all considered attributes. Table I presents certain statistics regarding the similarity calculated by the network.

The quality assessment was done with use of measures typical for classification applications such as *precision*, *recall* and *F1-score*. The solution (network) constructed achieved a global F1-score of 0.86 for Experiment 1 and 0.78 for Experiment 2. Both of these results can be considered sufficient for practical application. Table II contains more a detailed presentation of the best and worst results obtained during Experiments 1 and 2.

Patterns assigned to objects and patterns in the reference set were frequently quite complicated, which is a direct consequence of the almost non-existent format requirements and use of (quasi-)natural language. The table illustrates the kinds of patterns that appeared as a reference and were used to obtain results presented in table II. The types of elements of patterns corresponding to comparators in the network: Book (B), Country (C), DOI (D), Journal (J), Pages (P), Proceedings (R), Series (S), Title (T), Volume (V), Year (Y), Authors (A).

The experimental results presented show that the method used in Experiment 1 was more effective than the one used in Experiment 2. A closer look at the level of particular cases (objects) revealed that the reference set used in the second layer of the comparator network during Experiment 2 was

TABLE III
EXAMPLES OF OBJECTS IN NETWORK'S REFERENCE SETS.

Year
$START-CYYYY$
Jan $START-CYYYY$
Dec $START-CYYYY$

Pages
[p][0-9]{1,}-[0-9]{1,}
[p][\.][0-9]{1,}-[0-9]{1,}
[p][0-9]{1,}–[0-9]{1,}
[p][\.][0-9]{1,}–[0-9]{1,}

DOI
u00A0\[doi>.{0,}\]
u00A0\[doi>.{0,}

Authors
A. A. Abatan
A. A. Abonamah
A. A. Agrawal
C. W. Lin
F. Bonner

Volume
v\.\d+
v\d+\.
v\d+
v\.\d+ n\.\d+

Title
A comparison between conceptual clustering and conventional clustering
A Kalman-filter approach to equalization of CDMA downlink channels
A KDD System for the Discovery of Quantified Exception Rules

Journal
Fundamenta Informormaticae
IEEE transactions on computers
Journal of computer and system scien.

Proceedings
.{0,}\bproceedings\b \.{0,}
.{0,}\bproc\b\..{0,}
.{0,}\bproc\b\.\s.{0,}
.{0,}\bproc\b\s.0,

Country
POLAND
UNITED STATES
CHINA
INDIA

Structural patterns
ATJPY
ATRPY
ABTSVY
ATJY
ATJYP

insufficient. Some of the patterns that appeared in test data for Experiment 2 had no representative among the reference patterns, making comparison and identification invalid. There were also cases of errors in reference texts that were impossible to detect during preprocessing, but they were successfully filtered out by internal layers of the network, albeit at the cost of slightly reduced quality. Overall, the method based on the comparator network proved a useful addition to the system aimed at solving the task of processing texts in natural language.

B. *Risk Assessment During Fire and Rescue Operations*

1) Problem description: A Fire and Rescue (F&R) action is considered to be one of the most challenging environments for modeling and decision support. To date, there have been very few attempts to automate the decision making process in this area [15], [16], at least partially. One such attempt is the R&D project called ICRA (http://www.icra-project.org). The main goal of ICRA is to build a modern AI-based, risk-informed decision support system for the Incident Commander (IC), which improves situational awareness of the IC during F&R action, thus increasing the safety of firefighters. The basic ramifications and goals of the project can be found in [17].

One of the techniques introduced is the course of ICRA project is the *Threat Matrix*. The threat matrix groups the major types of threats with possible threat subjects. The threat matrix designed in the ICRA project is a significant extension of the model used currently by the German Fire Service. The new matrix is enriched with the possibility to define the degree (level) of the current threat which improves the representation of the current level of risks in the course of F&R operation. The extended version of the threat matrix is referred to as the *Risk Matrix*. The example of threats considered during the action using the ICRA version of the risk matrix is shown in table IV.

The experiments presented below were aimed at assessing the level of potential risks associated with the current operation through comparison (measuring similarity) with the previous, already fully described and classified cases retrieved from the repository. The underlying assumption of this approach is that the similarity between (circumstances of) two operations is an indicator of the similar types and levels of risks (threats) involved. In the operation scenario the IC would be informed

TABLE IV
RISK MATRIX REFLECTING OCCURRENCE OF THREATS WITH RESPECT TO THREAT TYPE AND POSSIBLE THREAT SUBJECTS. NOTATION: A1 - FEAR, A2 - TOXIC SMOKE, A3 - RADIATION, A4 - BURN-OUT, C - DANGEROUS CHEMICAL AGENT, E1 - COLLAPSE, E2 - ELECTROCUTION THREAT, E3 - DISEASE OR INJURY, E4 - EXPLOSION

Subject/Threat	A1	A2	A3	A4	C	E1	E2	E3	E4
People (ME)									
Animals (T)									
Environment (U)	-								
Property (S)	-	-				-	-	-	
Rescuers (MA)								-	
Equipment (G)	-	-						-	

that the current evolution of the situation shows similarities with a certain historical events and that some risks factors are more likely to emerge. As part of implementation of this experimental approach representation of an F&R operation as a vector of values have been devised.

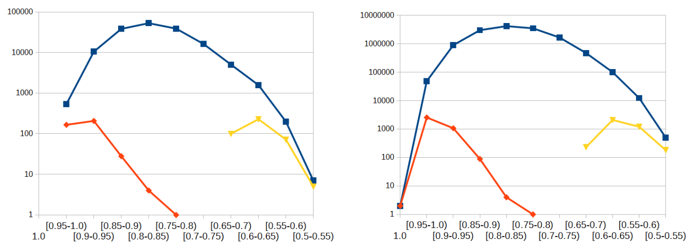

Fig. 4. Probability distributions for all, the best and the worst pairs of objects identified with the *leave-one-out* method. The graph on the left corresponds to results obtained from the examination of a set containing 406 operation reports. The graph on the right correspond to interventions. Both graphs use logarithmic scale. The Y axis shows the number of pairs while the X axis shows the clustered values of probability.

2) Experimental results: The proposed solution, based on comparator networks, was verified in the course of two experiments. The first of experiments was performed with use of a small data set of 406 F&R operation reports retrieved from the EWID[3] system. The experiment used the same data set, the same requirements and the same initial assumptions as the experiments reported in [18]. That allowed fully justified and

[3]Incident Data Reporting System used by the Polish State Fire Service

fair comparison of results obtained with methods used in [18] and those based on the comparator network.

The second experiment used data retrieved from the EWID as well. In this case the set contained 3736 reports. In both experiments the *leave-one-out* scheme was adopted resulting in, respectively, 164430 and 13953960 pairs of objects (reports) being considered in the calculation of similarities. It should be mentioned that the numbers of pairs that were actually used is lower than the cardinality of the respective cartesian products. This is due to filtering out those pairs that displayed very low similarity value, i.e. the pairs of very dissimilar reports.

The central tendency for results of both experiments is presented in table V. The distributions of similarity values obtained in experiments are shown in Fig. 4.

TABLE V
SELECTED MEASURES FOR CENTRAL TENDENCY FOR PAIRS OF OBJECTS: * - RESULTS FOR 406 CASES, ** - RESULTS FOR 3736 CASES. NOTATION: *all* - ALL PAIRS OF OBJECTS CONSIDERED, *bsp* - BEST SIMILARITY FOR INPUT PAIR, *wsp* - WORST SIMILARITY FOR INPUT PAIR.

Factor	*-all	*-bsp	*-wsp	**-all	**-bsp	**-wsp
Max	0.985	0.985	0.690	1.000	1.000	0.676
Min	0.530	0.791	0.530	0.477	0.797	0.477
Average	0.815	0.940	0.627	0.811	0.956	0.606
Median	0.820	0.946	0.630	0.824	0.961	0.608
Standard deviation	0.061	0.028	0.031	0.062	0.023	0.030
Variance	0.004	0.001	0.001	0.004	0.001	0.001

The results presented here for methods NoC $\frac{1}{n}$ and NoC WA were obtained with the default value of the threshold parameter $p = 0.5$. The weights for global aggregation used by NoC WA were calculated by means of the evolutionary algorithm. The best result was achieved for weights $\frac{2}{89}, \frac{60}{89}, \frac{1}{89}, \frac{26}{89}$ associated with, respectively, comparators for activities (sub-processes): notifications, disposals, recognitions, actions. These weights were obtained with the use of 33% of data (136 cases).

TABLE VI
COMPARISON OF VARIOUS METHODS FOR RECOGNITION OF RISKS. NOTATION: ESA - EXPLICIT SEMANTIC ANALYSIS, k-NN CANBERRA - k THE NEAREST NEIGHBORS WITH *Canberra* DISTANCE MEASURE [19], NOC $\frac{1}{n}$ - NETWORK OF COMPARATORS WITH AGGREGATION CALCULATED AS AVERAGE, NOC WA - NETWORK OF COMPARATORS WITH AGGREGATION CALCULATED AS WEIGHTED AVERAGE.

Method	Precision	Recall	F1-score
Naive Bayes	0.68	0.64	0.61
ESA	0.48	0.70	0.54
k-NN Canberra	0.74	0.74	0.69
NoC $\frac{1}{n}$	0.73	0.70	0.66
NoC WA	**0.79**	**0.75**	**0.71**

The final results for comparator networks were assessed in two steps. As part of the first step, the values of precision, recall and F1-score were calculated for every case of F&R operation in the data set and then averaged. These results are shown in table VI in comparison with the alternative classification methods applied to the same data. The second steps involved the analysis of the effectiveness of the method with respect to particular risks as identified in the risk matrix developed for the ICRA project. These results are presented in table VII.

TABLE VII
COMPARISON OF F1-SCORE VALUE FOR VARIOUS METHODS OF RECOGNITION WITH W.R.T. RISK TYPES. RISK TYPES ARE DEFINED TAKEN FROM RISK MATRIX [16]. NOTATION: NB- NAÏVE BAYES, ESA - EXPLICIT SEMANTIC ANALYSIS, k-NN C - k NEAREST NEIGHBORS WITH *Canberra* DISTANCE MEASURE [19], NOC $\frac{1}{n}$ - NETWORK OF COMPARATORS WITH AGGREGATION CALCULATED AS AVERAGE, NOC WA - NETWORK OF COMPARATORS WITH AGGREGATION CALCULATED AS WEIGHTED AVERAGE; * - RESULTS FOR 406 CASES, ** - RESULTS FOR 3736 CASES.

Risk	NB*	ESA*	k-NN C*	NoC $\frac{1}{n}$*	NoC WA*	NoC $\frac{1}{n}$ **
A1_MA	0.38	0.45	0.34	0.36	**0.39**	0.47
A1_ME	0.86	0.82	**0.91**	**0.91**	0.90	0.91
A1_T	–	0.07	0.09	0.12	**0.16**	0.14
A2_MA	0.81	0.84	**0.89**	0.85	0.88	0.70
A2_ME	0.83	0.84	**0.90**	0.89	0.89	0.84
A2_S	**0.29**	0.22	0.09	0.17	0.1	0.20
A2_T	0.05	0.14	0.09	0.13	**0.17**	0.17
A2_U	0.39	0.30	0.44	0.34	**0.45**	0.38
A4_G	–	0.08	0.21	0.11	**0.25**	0.14
A4_MA	0.30	0.22	0.35	0.24	**0.47**	0.34
A4_ME	0.27	0.17	**0.41**	0.16	0.36	0.33
A4_S	–	–	**0.40**	0.23	0.30	0.34
A4_T	–	0.13	–	–	**0.67**	0.06
E1_MA	–	0.11	**0.48**	0.22	0.42	0.40
E1_ME	–	–	**0.22**	–	–	0.21
E2_MA	0.11	**0.31**	0.17	0.07	0.24	0.30
E2_ME	–	**0.24**	0.20	0.09	0.14	0.24
E2_S	–	**0.15**	0.13	–	0.12	0.03
E3_G	–	0.12	**0.40**	–	–	0.08
E3_MA	–	**0.50**	0.16	0.13	0.17	0.28
E3_ME	–	–	0.12	**0.28**	0.13	0.42
E4_MA	–	–	–	–	**0.14**	0.13
E4_ME	–	–	–	–	–	0.14
E4_S	–	–	–	–	–	0.12

3) Interpretation of results: As shown in tables VI and VII, the best results were obtained for the *NoC WA* (Network of Comparators with Weighted Average) approach. In both experiments these assessments were reflected by values of standard quality measures [20], such as precision, recall and F1-score. The average value of precision for comparator-based approach was equal to 0.79 and by 0.05 higher than the average for the next best solution. A similar difference was observed for other measures (recall, F1-score) when comparing NoC WA with other classification methods. Detailed examination of the results w.r.t. particular types of risks, as shown in table VII, also favours the approach based on comparator networks. Overall, the NoC WA method proved to be a very promising approach to automation of decision support process for F&R operations [15], [16].

VI. SUMMARY OF THE POSITION

In the paper we put forward a novel methodology for reasoning as regards compound objects on the basis of their similarity to elements of knowledge base. This methodology offers a systematization of the process of constructing a system that identifies and classifies a compound object by comparing it to reference objects at various levels of abstraction. The proposed comparator networks complement and extend the existing gamut of similarity-based approaches to identification, management and classification of compound data entities.

An important advantage of the methodology presented in this paper is the availability of implementation. The implementation was done with domain experts in mind. For a user who understands the nature of the problem the materialization of a solution in form of a comparator network comes quite naturally. The library `comparators-lib` that contains algorithms is implemented in JAVA [6]. It provides a collection of base classes and interfaces corresponding to constructs in the underlying methodology, such as a class for comparator network, class for layer, class for comparator, etc. The classes seamlessly facilitate data flow and parameter setting, so that the user can concentrate on designing the architecture of the solution and defining basic elements that lead to the final computational system. In this way, the whole process of building the comparator network for a given application becomes easier.

The current status of the technology based on the concept of comparators and comparator networks is presented in this paper. The functions and operation of underlying concepts and definitions were illustrated with examples of real-life applications. The next step for the development of this technology could be the creation of a networked platform providing access to methods and algorithms for comparator network architecture. Equipped with a well-designed graphical interface such a platform would greatly simplify the construction of network models for new applications. The platform would also provide a collection of templates that could be used for fast prototyping of the solution and finding optimal model parameters more efficiently. This can make it a useful tool for many different types of approaches based on similarity, i.e. semantic similarity [21], similarity-based reasoning, mereological similarity and approximate reasoning [22] as well as many others. All of them can be implemented by means of compound objects comparators.

References

[1] A. Aamodt and E. Plaza, "Case-based reasoning: Foundational issues, methodological variations, and system approaches," *Artificial Intelligence Communications*, vol. 7, no. 1, pp. 39–59, 1994. [Online]. Available: http://dl.acm.org/citation.cfm?id=196108.196115

[2] D. Ślęzak and Ł. Sosnowski, "SQL-based Compound Object Comparators: A Case Study of Images Stored in ICE," in *Proc. of FGIT-ASEA 2010*, ser. Communications in Computer and Information Science, vol. 117, 2010. doi: 10.1007/978-3-642-17578-7_30 pp. 303–316. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-17578-7_30

[3] Ł. Sosnowski, "Characters recognition based on network of comparators," in *Techniki informacyjne teoria i zastosowania*, A. Myśliński, Ed. IBS PAN, 2012, vol. 4, pp. 123–134. ISBN 83-894-7555-3

[4] ——, "Inteligentne dopasowanie danych przy użyciu teorii zbiorów rozmytych w systemach przetwarzania danych," in *Analiza systemowa w finansach i zarządzaniu*, J. Hołubiec, Ed. IBS PAN, 2009, vol. 11, pp. 214–218. ISBN 9788389475220

[5] Ł. Sosnowski and D. Ślęzak, "How to design a network of comparators," in *Brain and Health Informatics*, ser. Lecture Notes in Computer Science, K. Imamura, S. Usui, T. Shirao, T. Kasamatsu, L. Schwabe, and N. Zhong, Eds., vol. 8211. Springer, 2013. doi: 10.1007/978-3-319-02753-1_39. ISBN 978-3-319-02752-4 pp. 389–398. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-02753-1

[6] Ł. Sosnowski, "Framework of compound object comparators," *Intelligent Decision Technologies*, vol. 9, no. 4, pp. 343–363, 2015. doi: 10.3233/IDT-140229. [Online]. Available: http://dx.doi.org/10.3233/IDT-140229

[7] R. W. Quackenbush, "On the composition of idempotent functions," *algebra universalis*, vol. 1, no. 1, pp. 7–12, 1971. doi: 10.1007/BF02944949. [Online]. Available: http://dx.doi.org/10.1007/BF02944949

[8] J. Kacprzyk, *Multistage Fuzzy Control: A Model-based Approach to Fuzzy Control and Decision Making*. John Wiley & Sons, 2012. ISBN 9780470744161

[9] J. Rumbaugh, I. Jacobson, and G. Booch, *The Unified Modeling Language Reference Manual, 2nd Edition*. Pearson Higher Education, 2004. ISBN 0321245628

[10] Ł. Sosnowski and D. Ślęzak, "Networks of compound object comparators," in *FUZZ-IEEE*. IEEE, 2013. doi: 10.1109/FUZZ-IEEE.2013.6622547. ISBN 978-1-4799-0020-6 pp. 1–8. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/FUZZ-IEEE.2013.6622547

[11] ——, "Fuzzy set interpretation of comparator networks," in *Pattern Recognition and Machine Intelligence - 6th International Conference, PReMI 2015, Warsaw, Poland, June 30 - July 3, 2015, Proceedings*, 2015. doi: 10.1007/978-3-319-19941-2_33 pp. 345–353. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-19941-2_33

[12] Ł. Sosnowski, A. Pietruszka, and S. Łazowy, "Election algorithms applied to the global aggregation in networks of comparators," in *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems*, ser. Annals of Computer Science and Information Systems, M. P. M. Ganzha, L. Maciaszek, Ed., vol. 2. IEEE, 2014. doi: 10.15439/2014F494 pp. pages 135–144. [Online]. Available: http://dx.doi.org/10.15439/2014F494

[13] Ł. Sosnowski, "Applications of comparators in data processing systems," *Technical Transactions, Automatic Control*, pp. 81–98, 2013.

[14] R. Bembenik, Ł. Skonieczny, H. Rybiński, and M. Niezgódka, Eds., *Intelligent Tools for Building a Scientific Information Platform*, ser. Studies in Computational Intelligence. Springer, 2012, vol. 390. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-24809-2

[15] L. Han *et al.*, "Firegrid: An e-infrastructure for next-generation emergency response support," *Journal of Parallel and Distributed Computing*, vol. 70, no. 11, pp. 1128 – 1141, 2010. doi: http://dx.doi.org/10.1016/j.jpdc.2010.06.005. [Online]. Available: http://dx.doi.org/10.1016/j.jpdc.2010.06.005

[16] A. Krasuski, A. Jankowski, A. Skowron, and D. Ślęzak, "From sensory data to decision making: A perspective on supporting a fire commander," in *Web Intelligence/IAT Workshops*. IEEE Computer Society, 2013, pp. 229–236. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/WI-IAT.2013.188

[17] Ł. Sosnowski, A. Pietruszka, A. Krasuski, and A. Janusz, "A resemblance based approach for recognition of risks at a fire ground," in *Active Media Technology - 10th International Conference, AMT 2014, Warsaw, Poland, August 11-14, 2014. Proceedings*, 2014. doi: 10.1007/978-3-319-09912-5_47 pp. 559–570. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-09912-5_47

[18] A. Krasuski and A. Janusz, "Semantic tagging of heterogeneous data: Labeling fire & rescue incidents with threats," in *FedCSIS*, M. Ganzha, L. A. Maciaszek, and M. Paprzycki, Eds., 2013, pp. 77–82. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6643979

[19] F. Malik and B. Baharudin, "Analysis of distance metrics in content-based image retrieval using statistical quantized histogram texture features in the {DCT} domain," *Journal of King Saud University - Computer and Information Sciences*, vol. 25, no. 2, pp. 207 – 218, 2013. doi: http://dx.doi.org/10.1016/j.jksuci.2012.11.004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1319157812000444

[20] A. M. Rinaldi, "An ontology-driven approach for semantic information retrieval on the web," *ACM Transactions on Internet Technology*, vol. 9, pp. 10:1–10:24, July 2009. doi: http://doi.acm.org/10.1145/1552291.1552293. [Online]. Available: http://doi.acm.org/10.1145/1552291.1552293

[21] P. Resnik, "Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language," *CoRR*, vol. abs/1105.5444, 2011. doi: http://dx.doi.org/10.1613/jair.514. [Online]. Available: http://arxiv.org/abs/1105.5444

[22] L. Polkowski, *Approximate Reasoning by Parts: An Introduction to Rough Mereology*, ser. Intelligent Systems Reference Library. Springer, 2011. ISBN 9783642222795