Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/080606

Detecting and Extracting Semantic Topics from Polish Fire Service Raports

Marcin Mironczuk[*]

Institute of Computer Science of the Polish Academy of Sciences, Jana Kazimierza 5, Warsaw, Poland

Received: 8 Oct. 2013, Revised: 6 Jan. 2014, Accepted: 7 Jan. 2014
Published online: 1 Nov. 2014

Abstract: This article presents results of structuring text documents using the classification process. The proposed system based on classification process which used to extract information about the semantics (meaning) segments (sentences) that build text documents. The analysis was made on the reports coming from the National Fire Service (Polish Fire Service) event evidence system. The article describes the results of classification using the proposed classifiers and presents some future directions of research.

Keywords: reports analysis, text classification, segments, sentence classification, semantic extraction, semantic extraction information, text document classifier, text mining, Naive Bayes, Rocchio classifier, k-nn classifier, nearest neighbor classifier

1 Introduction

In the National Fire Service (Polish Fire Service) after each fire-brigade emergency (intervention) is made electronic and paper documentation. The form of this documentation is governed by the regulation [4].The electronic version is stored in the event evidence system EWIDSTAT [1,2]. Part of the electronic version, the field entitled *Descriptive data for the information of event*, was proposed as a source of cases. These cases are processed in the Case Based Reasoning CBR system [30,17]. CBR is a basic platform of prototype Decision Support Systems DSS such as Hybrid Decision Support System HDSS [32,22]. Using natural language the commanders of rescue operation in the *Descriptive data for the information of event* field, describe different aspects of the surveillance activities such as: n*eutralization process hazard*, *type of equipment*, *description of the event place*, *type of intervention*, *meteorological conditions* etc. [4]. So, this section contains information that can be stored in the CBR system. These cases can be used by commanders to support a decision making in the rescue action. Research conducted by the author showed that the direct adaptation of the descriptions from the *Descriptive data for the information of event* field as a case of CBR system is limited [23]. This limitation results from the ability to search information in such cases by commanders. During the mapping of information from paper to electronic documentation, semantic information is lost i.e. meaning of the sentence is lost. Field *Descriptive data for the information of event* on paper documentation is divided into six subsections: *description of the emergency actions (hazards and difficulties, worn out and damaged equipment), description of the units arrived to the place of accident, description of what was destroyed or burned, weather conditions, conclusions and comments arising from the conduct of rescue operations* and *other comments about the data filled in form for the event*. In the EWIDSTAT system there is no such subsections division and a single report is written - semantic information is lost - there is no possibility to find information in the appropriate subsections and limit the search to a particular subsections. As a result, when searching the electronic section, the commander may get unexpected results, for example: query in the form of *hydrants Mickiewicza street* may request the information not only about the *hydrants* but also of all the rescue and firefighting actions on this street [23]. This problem will be solved by-developed system for structuring the *Descriptive data for the information of an event* field. The author has developed a system that implements text mining process. Thanks to this system unstructured report which is described in natural language is transformed into a semi-structured (the first step of analysis) and fully structured report (the second step of analysis). Semi-structured report is further expressed in natural

[*] Corresponding author e-mail: m.marcinmichal@gmail.com

language, but it has a semantic annotations. Are partially restored, the original sections of a paper documentation i.e. *Descriptive data for the information of event* field. This first step of analysis improves the quality of information retrieval and enables further information transformation to a fully structured form. This form is a model as attribute-value. Using this model can be stored and extract the information about for example *water points - hydrants*. In this case, rather than phrases in natural language descriptions such as *hydrants number 1234 Mickiewicz street efficient* are available structured descriptions of attribute-value for example *identifier = 1234*, *localization.streetName = Mickiewicza*, is*Efficient = True*. In this article author describes the first step of structuring the *Descriptive data for the information of event* field. The author has defined structuring as a kind of *information extraction* [24,25], which relies on the recovery of semantic information, context *Q* or otherwise the meaning of the text segment. Term *segment* using in this article is a synonym of *sentence* which is a part of the wider text document. The segment has a definite beginning and end. The segment usually begins with a capital letter and ends with a dot or other punctuation. Finding the meaning in the text segment is done by setting and then give the label as a semantic class name. Semantics, i.e. meaning the segment is determined by the terms that make up the segment. Meaning is defined by a function describing the combination of individual terms in the segment. Meaning the segment, is determined on the basis of function the classification model in the field of *artificial intelligence*. So understood structuring is an intermediate form between the classification of entire text documents and the study of individual terms. In the *text mining*, classification, text documents are usually considered as a set of terms represented by the matrix in a vector space or by using graph [29,26,8]. At the stage of pre-processing of text documents is filtered information as for instance: unnecessary alphanumeric characters (”,”, ”.”, ”’” etc.), terms from stop list etc. Text mining process usually ignores the study of grammar and morphology depending on the level of individual terms. These research areas are the domain of natural language processing NLP [31,13]. Optionally NLP is complementary to the pre-processing of text documents by providing solutions such as: morpho-syntactic operation, tokenization, lematization, stemming etc. However, both the first and second approach is insufficient in the study conducted by the author (pure text mining and NLP). This follows from the that they ignore the study segment, the wider the text as an independent object that can carry the information itself.

Although the reports are expressed in natural language the author did not apply NLP tools for deep parsing of text (part of speech, morpho-syntactic operation etc.). The author did not apply this tool to classify due to the ambitious objective of the experiment. The aim was to create a ”light” version of the tool to structuring the *Descriptive data for the information of an event* field. This tool would use the word (n-grams) and optionally information about the structure of the report (the report length and position classified segment in the report) without time-consuming NLP tools to deep parsing. Moreover, the use of these tools in the process of classification and study of their impact on this process is planned for future author’s studies. In this article, the author proposes a ”light” tool to structure the text based on the text mining technique and author’s classifier. Structuring is achieved through the classification process used to find the semantics of the segments. The aim was therefore to answer the following question: What describes a segment of the report? Does segment describe: the place of the event, equipment, damage, etc.?

In section 2 of this article author describes the process of structuring. Author in this section presented at the goal structuring and explained on the example of proposed process. Subsequently, author has described, a Set Of Reference Segments SORS, a software stack for its processing, selected classifiers and presented the results of the classification. The study used classifiers: Naive Bayes, *k*-nearest neighbors *k*-nn, Rocchio (centroid) and authors modifications of centroid classifier. Section 3 provides a summary of the research.

2 Structuring text documents

The first stage of structuring was demonstrated with an example. It was assumed that the description of the event (report) from the field *Descriptive data for the information of event* has the following form:
”After arriving at place of accident concluded that the balcony on three floors open fire burn cabinets, wicker baskets, rags, windows and facade. The activities consisted of the administration of two currents of water on the offensive: 1 out of the land on the balcony, 2 - staircase led to the apartment. Doors were destroyed during balancing. The smoke was removed from the room, place of accident was submitted to the owner ——. The car to tunk up on the Labiszynskiej street, hydrant number 1673 - efficient.”

The author has established based on the qualitative analysis descriptions of the events that they can distinguish five types of classes. The author has created five classes (semantic classes), after reading about four thousand reports. By using heuristic rules, author manually assigned segments to the classes. For example, the heuristic rule is: *if the segment contains the words associated with the damage then classify this segment to damage class*. As a result of this operation has been obtained SORS. The names of these classes and segments classified them approximate the original entries from paper documentation. These classes may include segments which build description of the event. These classes were class: *operation*, *equipment*, *damage*, *meteo*, and *description*. After the segmentation of the report (by the sentence) and classify its various segments to the

Table 1: Example of semi-structured report. Source: [own elaboration]

Segment	Class (semantic label)
after arriving at place of accident concluded that the balcony on three floors open fire burn cabinets, wicker baskets, rags, windows and facade	description
the activities consisted of the administration of two currents of water on the offensive: 1 out of the land on the balcony, 2 - staircase led to the apartment	operation
doors were destroyed during balancing	damage
the smoke was removed from the room, place of accident was submitted to the owner ——	operation
the car to tunk up on the labiszynskiej street, hydrant number 1673 - efficient	equipment

aforementioned classes is obtained semi-structured report (semi-structured case event). An example of such a semi-structured report shown in table 1.

Table 1 presents an example of a semi-structured case events. This case consists of five parts. Extracted segments have been classified into four of the five above-mentioned classes. Article in the following subsections describe: statistics a Set Of Reference Segments SORS used for classification, the stack of the program which implements semantic classification process, classification of selected models and the results obtained from the classification process.

2.1 Classification process of text segments

The author selected from the system a collection of 28,8000 records of reports. The author has chosen for further study in a random 3735 reports. These reports are divided into segments using the developed program. Reference collection of 12,753 segments obtained from the segmentation, manually assigned to classes. Frequency, percentage of segments in each class, and the cumulative value of these values presented in table 2.

Data from table 2 are presented in the Pareto chart. This chart shows in figure 1.

Figure 1 presents the distribution of segments of each class. Segments assigned to the class *operations* and *description* have the biggest share in the SORS. These classes together cover 50% percent of the data. Other 50% percent of the data form a class: *equipment*, *damage* and *meteo*.

Table 2: SORS statistic. Source: [own elaboration]

Class	Frequency	Frequency cumulative value	Percent	Percent cumulative value
operation	4727	4727	37.065788	37.065788
description	4064	8791	31.867012	68.932800
equipment	2148	10939	16.843096	85.775896
damage	949	11888	7.441386	93.217282
meteo	865	12753	6.782718	100

SORS is represented in a vector space using the term-document matrix (term-segment) *A*. The author additionally extended the matrix of meta information. Meta information describes a class segment c_k, the length of the report d_i (expressed in the count of segments) and the position p_j of segment in the report. These relationships present equation 1.

$$s_n(t_m, c_k, d_i, p_j) \tag{1}$$

Where:

–t_m - term $t_m \in T$ as T is a set of terms,
–s_n - segment $s_n \in S$ as S is a SORS and $|S| = 12753$,
–c_k - class segment and $c_k \in C$, where C is a set of classes and C = $\{$*operation, equipment, damage, meteo, description*$\}$,
–d_i - length of the report and $d_i \in D$, where D is a set the length of reports and D = $\{$*1, ..., 28*$\}$,
–p_j - segment position in the report and $p_j \in P$, where P is a set of positions and P = $\{$*1, ..., 28*$\}$.

Expanded term-segment matrix denote as *A'*. An example matrix presented in table 3.

Table 3: An example expanded term-segment matrix *A'*. Source: [own elaboration]

Segments $s_n \in S$	Terms				Class $c_k \in C$	Length $d_i \in D$	Position $p_j \in P$
	t_1	t_2	...	t_m			
s_1	$w_{1,1}$	...	...	$w_{1,m}$	c_1	2	2
s_2	...	...	...	...	c_1	1	1
s_3	...	...	...	...	c_1	3	2
s_4	...	...	...	...	c_1	3	2
s_5	...	...	...	...	c_2	2	2
s_6	...	...	...	...	c_2	3	1
s_7	...	...	...	...	c_2	1	1
s_8	...	...	...	...	c_3	1	1
s_9	...	...	...	...	c_3	2	1
s_{10}	$w_{10,1}$	...	...	$w_{n,m}$	c_3	3	2

Using the c_k, d_i and p_j can be separated any subspaces (segments subspaces or segments subset) from

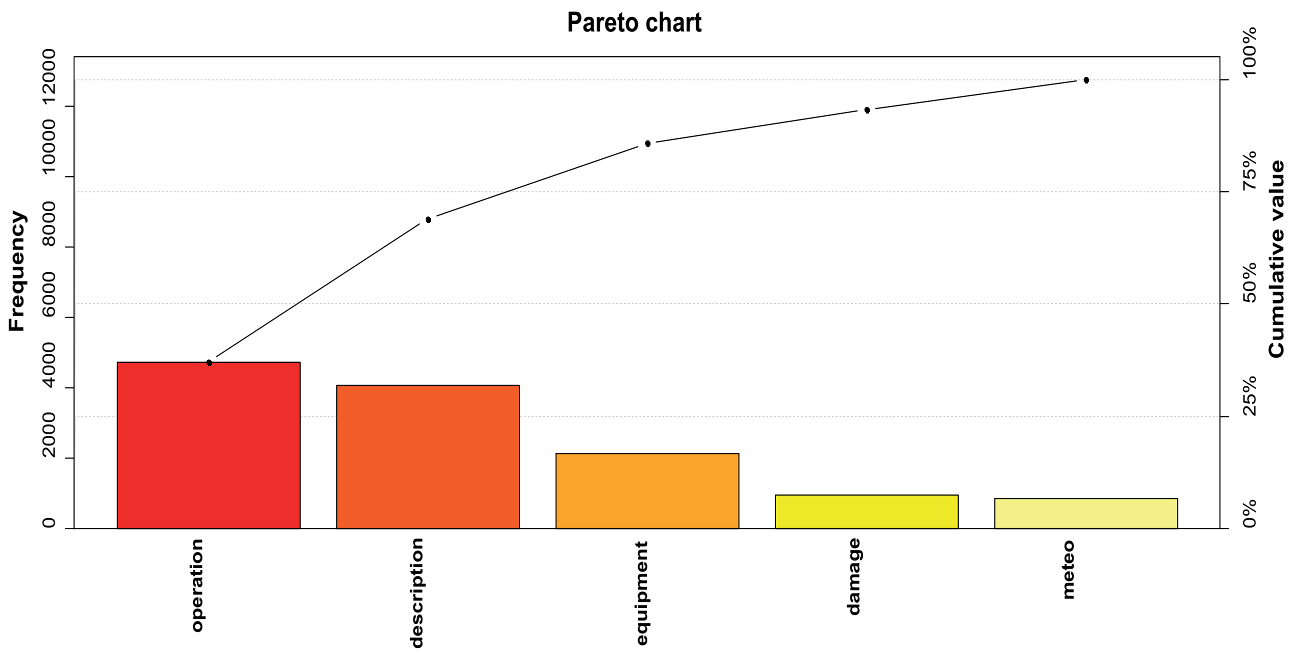

Fig. 1: Pareto chart of SORS. Source: [own elaboration, using the package [27]]

the matrix *A*' (table 3). The study consisted of reducing the space by using information about d_i and p_j, and p_j only. The author studied the effect of this limitation on the classification of the segment s_n to class c_k. For example: segment s_x should be classified in one of the classes c_k. Segment s_x is described by the terms t_m and additional meta information about the $d_i = 3$ and $p_j = 2$, i.e. $s_x(c_x, d_i = 3, p_j = 2)$ or additional information only for $p_j = 2$, i.e. $s_x(c_x, p_j = 2)$. In the presented problem (example) shows that the space of segments needed to build a classifier may be limited. In this case, the space will be limited to three segments s_3, s_4 and s_{10} for $d_i = 3$ and $p_j = 2$ and the five segments s_1, s_3, s_4, s_5 and s_{10} for $p_j = 2$. Aspect of this limitation concerned the Rocchio classifier type. Classifiers *k*-nn and Naive Bayes used unmodified standard space (meta data were not considered).

The process of segments classification from SORS was realized by the process shown in figure 2.

Figure 2 presents a plan implemented research and proposal process for the selection and evaluation of classifiers. SORS is represented as a matrix of term-segment (table 3). Weights terms $w_{n,m}$ take values: Binary, term frequency TF and term frequency-inverse document frequency TF-IDF [29,5]. Binary weights were used when testing of classifiers: *k*-nn, Naive Bayes (Bernoulli model), Rocchio and its author's modification (section 2.2). Weights TF and TF-IDF were used to test the Rocchio classifier and its modifications. Filter component reductions or no terms from SORS (its vector representation). Filter element:

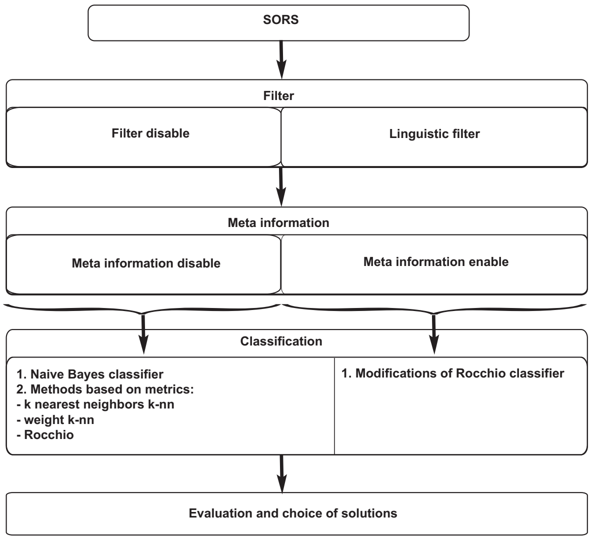
SORS
Filter
Filter disable
Linguistic filter
Meta information
Meta information disable
Meta information enable
Classification
1. Naive Bayes classifier
2. Methods based on metrics:
- k nearest neighbors k-nn
- weight k-nn
- Rocchio
1. Modifications of Rocchio classifier
Evaluation and choice of solutions

Fig. 2: The process of segments classification. Source: [own elaboration]

–filter disable - does not reduce terms space (unchanged space of 16,030 terms),
–linguistic filter - changes the terms space, which after modifications consist of 10,475 and 8,753 terms. Modifying the terms space through the use of *n*-grams and lemmatization gives a 10,475 terms. Modification of the process that uses only lemmatization gives

8753 terms. *N*-grams base (e.g. two-grams like a *hydrant number* and tree-gram like a *begin process evacuation*) was constructed using the Weka software [21]. The process of lemmatization use Morfologic program [3]. Linguistic filter is part of the *construction features* (stage *feature extraction*) in Knowledge Discovering From Databases KDD [12, 11]. Furthermore, this filter does not have a component related to *features selection* [10, 18, 6, 14]. Before applying the *n*-grams base and lemmatization, is used stop list. From matrix *A'* are removed, do not affect the analysis, terms like *"a"*, *"and"* etc. All of the above mentioned filter elements consist of reports pre-processing step.

Segments were processed by the filter encapsulation or not by meta information. The segments are then classified. SORS divided so that 80% of the data set was used for model building classifications and 20% of the data set was used as a test set. The whole classification was evaluated by using the *10-fold cross validation*. The processing ends with the choice of a classifier model for semantic structuring reports.

2.2 Classification models

The literature generally describes the type of selected classifiers: *k*-nn and weighted version of *k*-nn, Naive Bayes (Bernoulli model) and Rocchio [20, 28, 15]. For these reasons, the author presents only six modified Rocchio classifier which uses the meta information.

2.2.1 Classifier based on the nearest internal subclass centroid

The first modification Rocchio classifier, based on the *nearest internal subclass centroid*, expressed by the following model:

$$\Delta_{k,i,j} = \frac{1}{|S_{k,i,j}|} \sum_{s \in S_{k,i,j}} s \tag{2}$$

Where:

- $\Delta_{k,i,j}$ - most internal subclass centroid,
- $S_{k,i,j}$ set of segments, which are segments belong to: a class c_k, document length d_i and position p_j i.e. $S_{k,i,j} = \{s : \langle s, c_k, d_i, p_j \rangle \in S\}$,
- $|S_{k,i,j}|$ - number of segments for which it is built *k*-th centroid,
- $s = s_n$ - segment weights $w_{n,m}$ described by terms t_m in the vector space segment.

Based on the such defined centroid is calculated similarity measure (Euclidean d_E, Cosine d_C, Jacard d_J and Dice d_D) [7, 16]

$$sim(s, \Delta_{k,i,j}) = \begin{cases} sim(s, \Delta_{k,i,j})_E = d_E \\ sim(s, \Delta_{k,i,j})_C = d_C \\ sim(s, \Delta_{k,i,j})_J = d_J \\ sim(s, \Delta_{k,i,j})_D = d_D \end{cases} \tag{3}$$

The classification of the segment $s_x = s$ of unknown class c_x to one of the classes c_k is as follows:

$$s_x(d_i, p_j) = \arg_{k=1,\dots,|C|} \begin{cases} sim(s, \Delta_{k,i,j})_E = d_E \\ sim(s, \Delta_{k,i,j})_C = d_C \\ sim(s, \Delta_{k,i,j})_J = d_J \\ sim(s, \Delta_{k,i,j})_D = d_D \end{cases} \tag{4}$$

According to equation 4 is taken the maximum or minimum, depending on the metric used, the value of the similarity between the segment s_x and most internal subclass centroid $\Delta_{k,i,j}$.

The second modification of the classifier does not differ from the solutions described above. The difference lies in the construction of a classifier. Information about the report length d_i is not considered only position p_j of segment in the report itself is considered. This difference lies in the fact that *most internal subclass centroid* has the following form:

$$\Delta_{k,j} = \frac{1}{|S_{k,j}|} \sum_{s \in S_{k,j}} s \tag{5}$$

Where:

- $S_{k,j}$ - set of segments, which are segments belong to: a class c_k, and position p_j i.e. $S_{k,j} = \{s : \langle s, c_k, p_j \rangle \in S\}$,
- $|S_{k,j}|$ - number of segments for which it is built *k*-th centroid,
- $s = s_n$ - segment weights $w_{n,m}$ described by terms t_m in the vector space segment.

Further process classification is the same as presented in the above classifier (only centroid $\Delta_{k,i,j}$ in formulas 3 and 4 is converted to centroid from formula 5 i.e. $\Delta_{k,j}$).

2.2.2 Classifier based on local weighted nearest subclass centroid

The third modification Rocchio classifier, based on the on *local weighted nearest subclass centroid*, expressed by the following model:

$$\Delta_{w_{k,i,j}} = \frac{\Delta_k + \Delta_{k,i} + \Delta_{k,i,j}}{3} = \frac{\frac{1}{|S_k|}\sum_{s \in S_k} s + \frac{1}{|S_{k,i}|}\sum_{s \in S_{k,i}} s + \frac{1}{|S_{k,i,j}|}\sum_{s \in S_{k,i,j}} s}{3} \tag{6}$$

Where:

–$S_{k,i}$ - set of segments, which are segments belong to: a class c_k, and document length d_i i.e. $S_{k,i} = \{s : \langle s, c_k, d_i \rangle \in S\}$,
–$|S_{k,i}|$ - number of segments for which it is built k-th centroid.

Further process classification is the same as presented in the section 2.2.1 (only centroid $\Delta_{k,i,j}$ in formulas 3 and 4 is converted to centroid from formula 6 i.e. $\Delta_{w_{k,i,j}}$).

The fourth modification of the classifier does not differ from the solutions described above (equation 6). The difference lies in the construction of a classifier. Information about the report length d_i is not considered only position p_j of segment in the report itself is considered. This difference lies in the fact that *local weighted nearest subclass centroid* has the following form:

$$\Delta_{w_{k,j}} = \frac{\Delta_k + \Delta_{k,j}}{2} = \frac{\frac{1}{|S_k|}\sum_{s\in S_k} s + \frac{1}{|S_{k,j}|}\sum_{s\in S_{k,j}} s}{2} \qquad (7)$$

Where:

–$S_{k,j}$ - set of segments, which are segments belong to: a class c_k and position p_j i.e. $S_{k,j} = \{s : \langle s, c_k, p_j \rangle \in S\}$,
–$|S_{k,j}|$ - number of segments for which it is built k-th centroid.

Further process classification is the same as presented in the section 2.2.1 (only centroid $\Delta_{k,i,j}$ in formulas 3 and 4 is converted to centroid from formula 7 i.e. $\Delta_{w_{k,j}}$).

2.2.3 Classifier based on global nearest-weighted subclass centroid

The fifth modification Rocchio classifier, based on the *global nearest-weighted subclass centroid*, expressed by the following model:

$$\Delta_{wg_k} = \frac{\Delta_k + \Delta_{wg_{k,i}} + \Delta_{wg_{k,j}}}{3} = \frac{\frac{1}{|S_k|}\sum_{s\in S_k} s + \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|S_{k,i}|}\sum_{s\in S_{k,i}} s + \frac{1}{m}\sum_{j=1}^{m}\frac{1}{|S_{k,j}|}\sum_{s\in S_{k,j}} s}{3} \qquad (8)$$

Where:

–n - maximum length of the document in the class c_k,
–m - number of positions that may take a segment in the class c_k on grounds of the document d_i.

Further process classification is the same as presented in the section 2.2.1 (only centroid $\Delta_{k,i,j}$ in formulas 3 and 4 is converted to centroid from formula 8 i.e. Δ_{wg_k}).

The sixth modification of the classifier does not differ from the solutions described above (equation 8). The difference lies in the construction of a classifier. This difference lies in the fact that *global nearest-weighted subclass centroid* has the following form:

$$\Delta_{wg_k} = \frac{\Delta_k + \Delta_{wg_{k,j}}}{3} = \frac{\frac{1}{|S_k|}\sum_{s\in S_k} s + \frac{1}{m}\sum_{j=1}^{m}\frac{1}{|S_{k,j}|}\sum_{s\in S_{k,j}} s}{3} \qquad (9)$$

Further process classification is the same as presented in the section 2.2.1 (only centroid $\Delta_{k,i,j}$ in formulas 3 and 4 is converted to centroid from formula 9 i.e. Δ_{wg_k}).

2.3 Classification results

This section was collected, presented and discussed the best classification results for the classifiers (section 2.2). The research used the space of terms:

–the whole space of terms consisting 16,030 terms,
–the reduced space of terms in the text preprocessing. This preprocessing uses a lemmatization and n-grams database (10,475 terms) and doesnt use n-grams database (8215 terms).

Below the author presents the best indicators obtained with each classifier: k-nn, Bayesian and Rocchio with or without modification. All descriptions are listed together by weight binary terms. This summary presents the table 4.

Data that are presented in table 4 visualized using appropriate graphs (asterisk in the table indicates that the results are consistent). The author has created a separate chart of the F-measure. This chart presents a summary of the classification results using k-nn classifier. Author examined the weighted and unweighted version this classifier, depending on the number of equations describing a set of segments. These charts presented in figure 3. Author also created the recall and precision charts for the best coefficients obtained from the classification process using the selected classifiers. Figure 4 presents these charts.

Figure 3 shows a comparison of classification results obtained using k-nn classifier. This figure shows that the use of n-grams base much worse classification results. The set of segments consisting of 8215 or 1630 terms gives better classification results than a collection consisting of 10475 terms. The difference between these solutions is 4% coefficient of F-measure. Weighting for the k-nn classifier practically had no effect on the classification process. Thus, it is not influenced significantly its indicators. Significant impact on the classification of segments with a k-nn classifier was the measure, the type of similarity metrics. The figure shows that the Euclidean measure in each case, for each terms space, gives significantly worse results than Jacard, Cosine or Dice measure.

Table 4: The best indicators obtained with each classifier (weight binary terms). Source: [own elaboration]

Number of terms	Classifier	Precision	Recall	F-measure
16030	Unweighted *k*-nn, k = 9, Jacard similarity, Cosine similarity*	0.898	0.897	0.897
	Weighted *k*-nn, k = 8, Dice similarity, Jacard similarity*	0.902	0.901	0.901
	Rocchio, Jacard similarity, Cosine similarity*	0.839	0.829	0.830
	Naive Bayes	0.861	0.851	0.845
	Global nearest-weighted subclass centroid with use a d_i and p_j, Jaccard similarity, Cosine similarity*	0.840	0.832	0.832
10475	Unweighted *k*-nn, k = 3, Jacard similarity	0.858	0.852	0.854
	Weighted *k*-nn, k = 8, Jacard similarity	0.866	0.860	0.861
	Rocchio, Jacard similarity, Cosine similarity*	0.818	0.799	0.805
	Naive Bayes	0.834	0.820	0.817
	Global nearest-weighted subclass centroid with use a d_i and p_j, Cosine similarity, Jacard similarity*	0.815	0.799	0.802
8215	Unweighted *k*-nn, k = 7, Jacard similarity	0.895	0.894	0.894
	Weighted *k*-nn, k = 9, Jacard similarity	0.899	0.898	0.898
	Rocchio, Jacard similarity, Cosine similarity*	0.831	0.819	0.820
	Naive Bayes	0.883	0.880	0.881
	Global nearest-weighted subclass centroid with use a p_j, Cosine similarity*	0.834	0.824	0.825

Table 5: F-measure coefficients of Rocchio classifier and his for different weighting schemes terms i.e. binary, TF and TF-IDF. Source: [own elaboration]

Number of terms	Terms weight type	Classifier	F-measure
8215	Binarna	ck	0.82
		cp	0.825
	TF	ck	0.818
		cp	0.822
	TF-IDF	ck	0.861
		cp	0.869
10475	Binarna	ck	0.805
		cp	0.802
	TF	ck	0.799
		cp	0.8
	TF-IDF	ck	0.847
		cp	0.849
16030	Binarna	ck	0.83
		cp	0.833
	TF	ck	0.829
		cp	0.834
	TF-IDF	ck	0.87
		cp	0.878

Figure 4 shows the best coefficients obtained from the classification process using the selected classifiers. This figure shows that the use of *n*-grams base in the classification process makes the operation of all the classifiers is worse. Lemmatization process significantly affect the naive Bayes classifier. Improving the coefficient of F-measure is 3.5% for the set of segments containing 1630 features and 6.5% for the set of segments containing 1630 features.

The results of the classification process via Rocchio classifier and its author modifications at different weight schemes: binary, TF, TF-IDF has been put together. This summary presents figure 5.

In order to discuss the recall and precision charts, which is shown in figure 5, created an additional summary statistics. Table 5 presents these statistics. This table contains the best F-measure coefficients of Rocchio classifier (in the figure 5 and table 5 marked as c_k) and a classifier which was based on him (in the figure 5 and table 5 marked as c_p).

According to the data which are presented in table 5 and figure 5, the Rocchio classifier and its modifications work best when using TF-IDF weights. TF-IDF weights give the best classification results for a set of segments consisting of 16,030 and 8,215 terms. Rocchio classifier, which uses a collection of 16.030 terms, gives 87% coefficient of F-measure. This coefficient is 87.8% when using authorial modifications. So the quality of classification could be increased by 0.8%. Rocchio classifier using reduced terms space to 8,215 terms and using TF-IDF weight gave 86.1% coefficient of F-measure. This coefficient is 86.9% when using authorial modifications. So the quality of classification

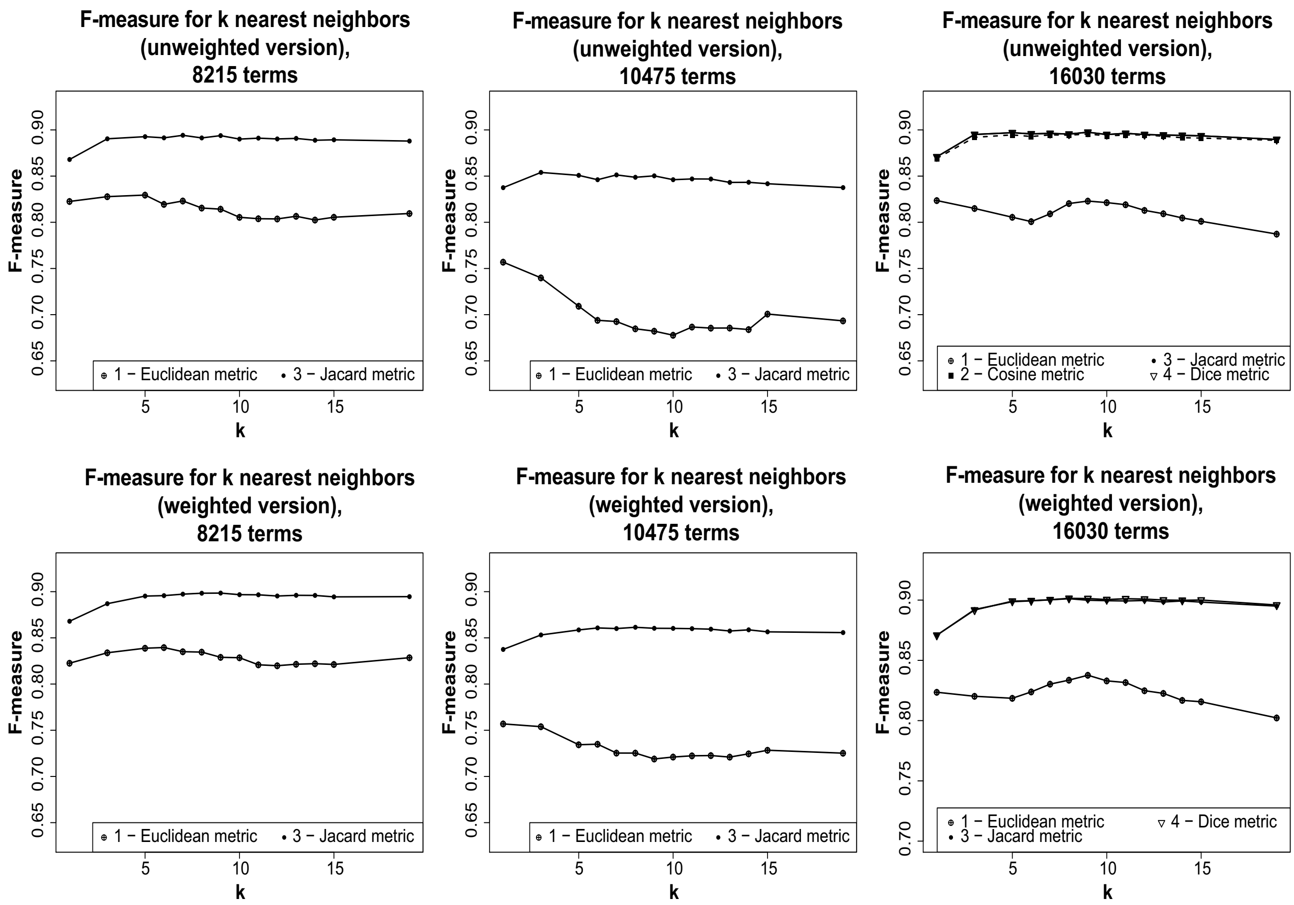

Fig. 3: Comparison chart of F-measure for *k*-nn classifier (weight binary terms). Source: [own elaboration]

could be increased by 0.8%. Reduce by half the number of terms does not significantly influence the degradation indicators of classification. This reduction influenced to reducing the effects of proposed modifications to the classification. Few substantial effect modification of the classification process can be seen when using binary and TF weights. In the case of Rocchio classifier using a set of 16,030 features, the F-measure coefficient of measurement for binary weight increased from 83% to 83.3% . So there was an increase by only 0.3%. While the weighting scheme TF for the same coefficient rose slightly by 0.5% from 82.9% to 83.4%. Using a set consisting of 8,215 terms obtained improved Rocchio classifier in a binary weighting scheme. Coefficient of F-measure increased by 0.5% from 82% to 82.5%. This coefficient for weighting scheme TF grew from 81.8% to 82.2%, and thus improved by less than 0.4%. Most insensitive to the modification of the classifier was a set of segments which uses a collection of 10,475 terms. This collection also gave the worst results of the classification.

3 Conclusions

In the article author presents the proposed segments classification process (semantic classification). The author has demonstrated the applicability of solutions for data mining (artificial intelligence, machine learning) to classification of segments that are part of the report. The results obtained from experiments, processing of Polish texts (segments), are broadly consistent with the results obtained in the world, i.e. with experiments on English texts [20, 15]. Using a simple linguistic filter (only *feature construction*), the author received a satisfactory classification results of up to 90% of correctly classified segments. The proposed modifications by the author Rocchio classifier have not worked to the end of the study. These modifications gave a slight improvement Rocchio classifier, or much worse classification parameters. The conclusion is that the class of segments cannot be modeled in this way, i.e. using additional meta information. However, author sees the possibility of applying meta-information in the classification using Bayesian networks. These networks allow the integration

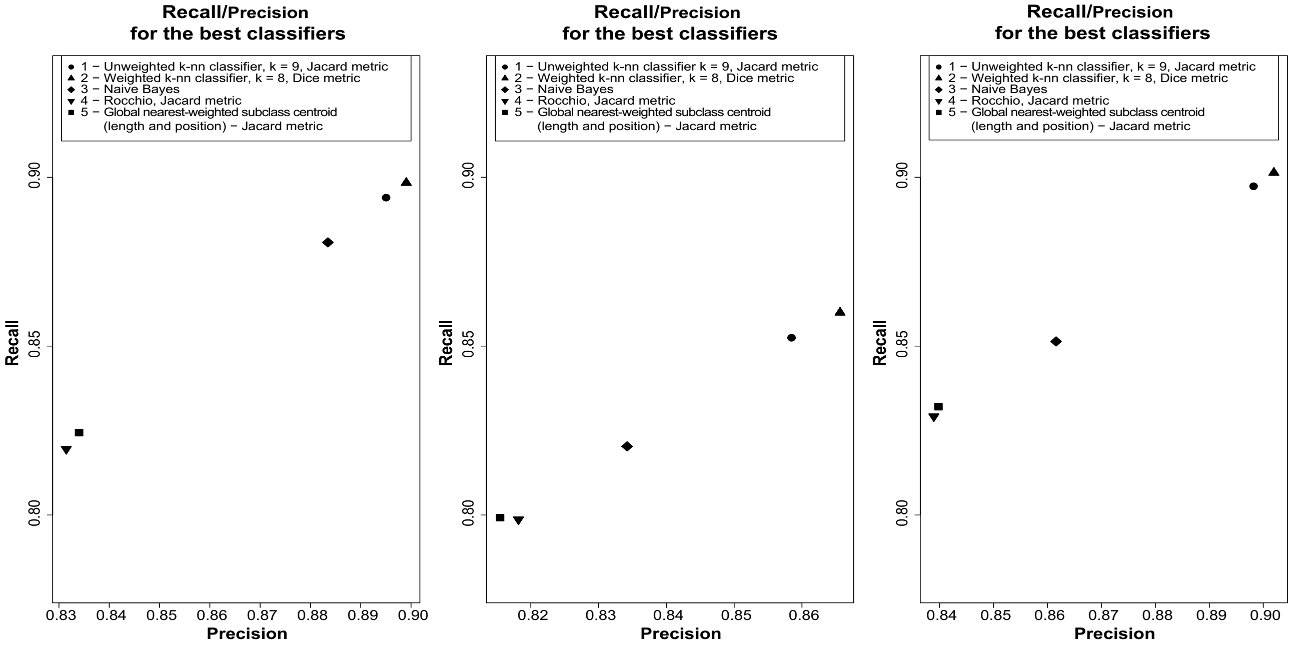

Fig. 4: The best coefficients obtained from the classification process using the selected classifiers (weight binary terms). Source: [own elaboration]

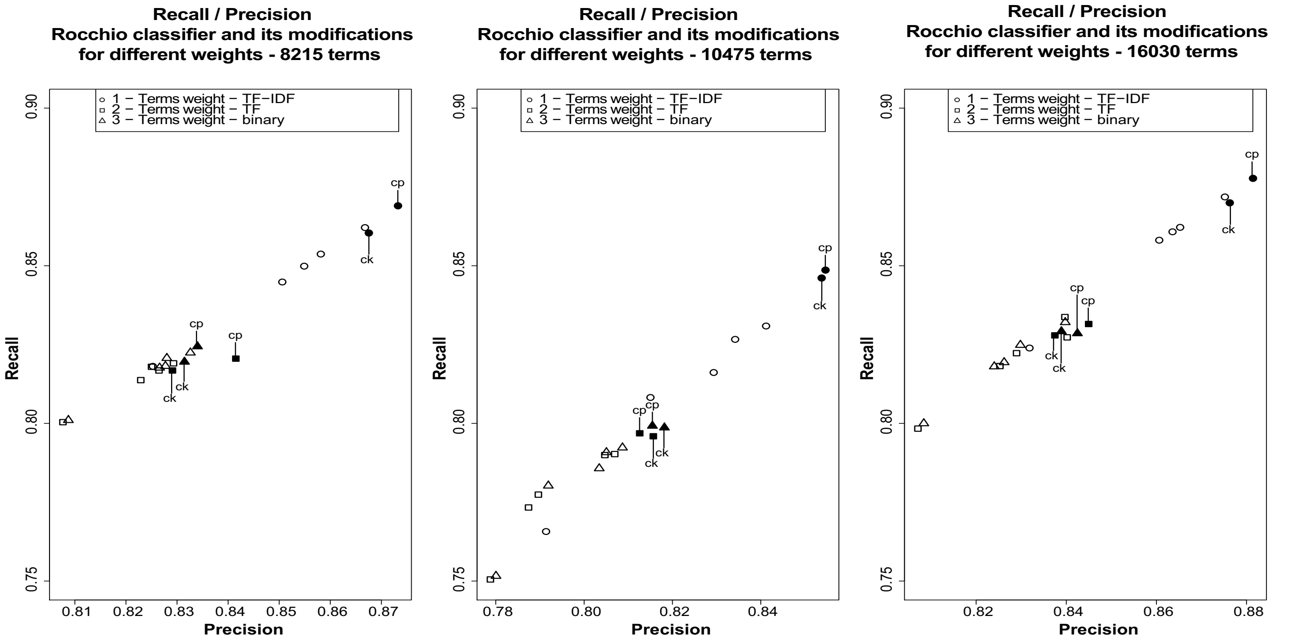

Fig. 5: The coefficients obtained from the classification process using the Rocchio classifiers and its modification. Source: [own elaboration]

of additional domain knowledge (meta information) to the process of inference [9].

Furthermore, research on Rocchio classifier and its modifications have shown that changing the term weight significantly improves the classification. Therefore it would be possible to expand studies on the use of mixed models and kernel estimators or normal distributions of term in a naive Bayes classifier. Despite the fact that classifiers based on nearest neighbors give the best results of their real implementation on the segments classification system is limited. This limitation is associated with the complexity of k-nn algorithms and classification time. Building a model based on the entire data set is an expensive process. For this reason it is recommended Naive Bayes and extending research on probabilistic methods of classification. Probabilistic model gives good results. These results are close to the k-nn classifier that use a *linguistic filter* with lemmatization terms. This filter should be followed to implement the element associated with the *feature selection*. This element is implemented to reduce the space of terms. The feature selection can be done using probabilistic ranking model like *entropy* or giving a set of attributes best describe the class like a *convex and piecewise linear* or another filter models [19,6].

Currently there are no documented reports of analysis. In the available literature of domain, the author could not find any processes or methods to process the reports. For these reasons can be considered that studies reports in the manner proposed by the author, are innovative in polish rescue fire service field. In the longer term research can yield for commander benefits such as structural CBR systems. Bases of these systems can subsequently be supplemented by information, coming also from the reports. Supplementation can be developed and implemented through information extraction system. The system realized the second stage of structuring (description of this stage was placed in the article appearing on the stage of review in the Polish Military University of Technology Bulletin).

References

[1] Abakus: System ewid99. [online]. URL http://www.ewid.pl. [Access date: 1 May 2009]

[2] Abakus: System ewidstat. [online]. URL http://www.ewid.pl. [Access date: 1 May 2009]

[3] Morfologik. [online]. URL http://morfologik.blogspot.com/. [Access date: 1 May 2011]

[4] Rozporzadzenie ministra spraw wewnetrznych i administracji z dnia 29 grudnia 1999 r. w sprawie szczegolowych zasad organizacji krajowego systemu ratowniczo-gasniczego. dz.u.99.111.1311 34 pkt. 5 i 6

[5] Arturo, M.R.: Automatic text categorization of documents in the high energy physics domain. Ph.D. thesis, Granada U., Granada (2006). Presented on 03 Mar 2006

[6] Biesiada, J., Duch, W., Kachel, A., Maczka, K., Palucha, S.: Feature ranking methods based on information entropy with parzen windows. In: Proceedings of the 9th International Conference on Research in Electrotechnologyand Applied Informatics (REI'05), 109–119 (2005). URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.60.4914

[7] Choi, S.S., Cha, S.H., Tappert, C.C.: A survey of binary similarity and distance measures. Journal on Systemics, Cybernetics and Informatics, **8**, 43–48 (2010).

[8] Chow, T.W., Zhang, H., Rahman, M.: A new document representation using term frequency and vectorized graph connectionists with application to document retrieval. Expert Systems with Applications, **36**, 12,023–12,035 (2009). URL http://dl.acm.org/citation.cfm?id=1598090.1598546

[9] Darwiche, A.: Modeling and Reasoning with Bayesian Networks, 1st edn. Cambridge University Press, New York, NY, USA, (2009).

[10] Dasgupta, A., Drineas, P., Harb, B., Josifovski, V., Mahoney, M.W.: Feature selection methods for text classification. In: Proceedings of the 13th ACM SIGKDD international conference on Knowledgediscovery and data mining, KDD'07, ACM, New York, NY, USA, 230–239 (2007). URL http://doi.acm.org/10.1145/1281192.1281220

[11] Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: Advances in knowledge discovery and data mining. chap. From data mining to knowledge discovery: an overview, American Association for Artificial Intelligence, Menlo Park, CA, USA, 1–34 (1996). URL http://dl.acm.org/citation.cfm?id=257938.257942

[12] Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: From data mining to knowledge discovery in databases. AI Magazine, **17**, 37–54 (1996)

[13] Grishman, R.: Computational linguistics: an introduction. Studies in natural language processing. Cambridge University Press, (1986). URL http://books.google.pl/books?id=Ar3-TXCYXUkC

[14] Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research: Special Issue on Variable and Feature Selection, **3**, 1157–1182 (2003). DOI 10.1162/153244303322753616. URL http://dx.doi.org/10.1162/153244303322753616

[15] Han, E.H.S., Karypis, G.: Centroid-based document classification: Analysis experimental results, 424–431 (2000).

[16] Kim, M.C., Choi, K.S.: A comparison of collocation-based similarity measures in query expansion. Information Processing and Management: an International Journal, **35**, 19–30 (1999). URL http://linkinghub.elsevier.com/retrieve/pii/S0306457398000405

[17] Krasuski, A., Maciak, T., Krenski, K.: Decision support system for fire service based on distributed database and case-based reasoning. Studies i logic grammar and rethoric, **17**, 159–169 (2009)

[18] Li, S., Xia, R., Zong, C., Huang, C.R.: A framework of feature selection methods for text categorization. In: Proceedings of the Joint

Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural LanguageProcessing of the AFNLP: ACL'09, Association for Computational Linguistics, Stroudsburg, PA, USA, **2**, 692–700 (2009). URL http://dl.acm.org/citation.cfm?id=1690219.1690243

[19] Lukaszuk, T.: Feature selection using cpl criterion functions. Zeszyty Naukowe Politechniki Bialostockiej. Informatyka, **4**, 85–95 (2009).

[20] Manning, C.D., Prabhakar, R., Hinrich, S.: Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA, (2008).

[21] Mark, H., Eibe, F., Geoffrey, H., Bernhard, P., Peter, R., H., W.I.: The weka data mining software: an update. ACM SIGKDD Explorations Newsletter, **11**, 10–18 (2009). URL http://doi.acm.org/10.1145/1656274.1656278

[22] Mironczuk, M., Maciak, T.: The project of hybrid decision support system for the fire service. Zeszyty Naukowe SGSP, **1**, 27–42 (2009). URL http://www.zn.sgsp.edu.pl/39/2.eps

[23] Mironczuk, M., Maciak, T.: The modified analysis fmea with the elements sfta in projects of hydrotehnical object information searsh system in nosql catalogue register. Studia Informatica, **32**, 155–177 (2011)

[24] Moens, M.F.: Information Extraction: Algorithms and Prospects in a Retrieval Context (The Information Retrieval Series). Springer-Verlag New York, Inc., Secaucus, NJ, USA, (2006).

[25] Mooney, R.J., Bunescu, R.: Mining knowledge from text using information extraction. SIGKDD Explorations Newsletter, **7**, 3–10 (2005). DOI http://doi.acm.org/10.1145/1089815.1089817. URL http://portal.acm.org/citation.cfm?id=1089817

[26] Neumann, G., Piskorski, J.: A shallow text processing core engine. Computational Intelligence, **18**, 451–476 (2002).

[27] Scrucca, L.: qcc: an r package for quality control charting and statistical process control. R News, **4/1**, 11–17 (2004). URL http://CRAN.R-project.org/doc/Rnews/

[28] Shin, K., Abraham, A., Han, S.Y.: Enhanced centroid-based classification technique by filtering outliers. In: TSD'06, 159–163 (2006).

[29] Solka, J.L.: Text data mining: Theory and methods. Statistics Surveys, **2**, 94–112 (2008). URL http://arxiv.org/abs/0807.2569

[30] Watson, I.: Applying Case-Based Reasoning: Techniques for Enterprise Systems. Morgan Kaufmann Publishers, (1997).

[31] Yi, J., Nasukawa, T., Bunescu, R., Niblack, W.: Sentiment analyzer: Extracting sentiments about a given topic usingnatural language processing techniques. In: Proceedings of the Third IEEE International Conference on Data Mining, ICDM'03, IEEE Computer Society, Washington, DC, USA, 427–434 (2003). URL http://dl.acm.org/citation.cfm?id=951949.952133

[32] Zhou, F., Yang, B., Li, L., Chen, Z.: Overview of the new types of intelligent decision support system. Information and Control, 1–4 (2008)

Marcin Mironczuk he is a graduate of Bialystok Technical University, Faculty of Electrical Engineering in Poland. The PhD degree received in Bialystok Technical University, Faculty of Computer Science. He is currently working in Poland Institute of Computer Science of the Polish Academy of Sciences. His research interests are in the areas of applied mathematics, data/text mining technique and information extraction.