

FORMALIZATION OF TECHNOLOGICAL KNOWLEDGE IN THE FIELD OF METALLURGY USING DOCUMENT CLASSIFICATION TOOLS SUPPORTED WITH SEMANTIC TECHNIQUES

The process of knowledge formalization is an essential part of decision support systems development. Creating a technological knowledge base in the field of metallurgy encountered problems in acquisition and codifying reusable computer artifacts based on text documents. The aim of the work was to adapt the algorithms for classification of documents and to develop a method of semantic integration of a created repository. Author used artificial intelligence tools: latent semantic indexing, rough sets, association rules learning and ontologies as a tool for integration. The developed methodology allowed for the creation of semantic knowledge base on the basis of documents in natural language in the field of metallurgy.

Keywords: Application of Information Technology to the Foundry Industry, Document Classification, Semantic Techniques, Knowledge formalization, Text mining

1. Introduction

Quality assurance is a key aspect of production and management in the steel industry. In an age of rapidly developing technology, an important element of the design and control of the process are information systems, including decision support tools [1-3]. The knowledge base of such systems consist of two types of formal resource-one of them is a set of decision rules [4,5], and the second-the repository of documents containing technological knowledge necessary for decision-making in emergency situations, in the non-routine activities [6,7]. Such documents that can be used, must be prepared ex ante-digitized, described, ready to search.

Human language is not the language understood by the machine, which is why the use of computational systems to select information requires the use of programming methods, which become a kind of bridge between man and machine. The methods used so far mainly consisted in comparing the searched term-key with the information stored in the knowledge base [8]. As the database grows, the amount of the results increases, making it difficult to reach the results of the "best" value from the point of view of the user.

To obtain the results of the content search closer to the user expectations one can use methods of an artificial intelligence: the method of latent semantic indexing [9], rough sets [10-12] and association rules learning [13,14]. These methods have one thing in common-to define the substantive content of the documents use patterns acquired through exploration of training set of documents. The result of these algorithms is to get the classes/

clusters of documents of common interest or content, which is an essential element in enhancing the quality of search in the knowledge base.

2. Investigation methods

A large group of collected resources are text documents in the domain of metallurgy. In addition to typical documents considered text documents, such as articles, notes, books (e-books), this group can also include Web pages, forms, and tables [15-17]. The exploratory analysis of text documents, known also as *text mining*, is a set of concepts, methods and, implemented in the form of software, algorithms for processing the text resources, leading to automation of processes related to documents prepared in natural languages. *Text mining* can also be defined as the process by which in the text resources is discovered the information previously unknown. As in data mining, also in text mining there is an attempt to explore the previously unknown regularities, rules, and patterns [18-20].

2.1. Text pre-processing

The research base is composed of text documents and the data not having a well-defined structure, written in natural language. The main problem that the designer of systems of content analysis must face is morphological analysis of the searched phrase. It has to do with the specific nature of the language-the

* AGH UNIVERSITY OF SCIENCE AND TECHNOLOGY, AL. A. MICKIEWICZA 30, 30-059 KRAKOW, POLAND

Corresponding author: regulski@agh.edu.pl

Polish language in particular-because many of the words are ambiguous and used depending on the context of the statement [6]. Therefore, simple lexical analysis fails and produces results far from the expected ones.

Stop list. In natural languages a large number of words in the text are, for example, conjunctions, pronouns and articles (in languages that have them). They do not contain information about the topics contained in a document, but only shape the course of the statement. They are found in nearly all documents, which makes their discriminant power very weak. A list of such words is called stop-list. Taking into account the words included in the stop-list leads to disturbances in the analysis of the document. The removal of such words allows reducing the dimensionality of the space, which leads to savings in memory and shortens the duration of making the algorithms.

Lemmatization consists in bringing different inflected grammatical forms to the basic form called lexeme. Lexeme means an abstract unit in linguistics specifying the lexical meaning and the set of all the grammatical functions, which it can perform. The canonical form of the lexeme is the lemma (the simplest of all possible forms-given in the dictionaries). The process of lemmatization is important in languages with rich inflection, which also include the Polish language. The application of the process of lemmatization may considerably limit the number of terms extracted from the document.

Stemming is the process of bringing the word to its root by eliminating the prefixes and suffixes. The root is part of the word, which in a unique way is identified with the lexeme. This operation contributes to a decrease in the data to be analyzed [21].

Vector Space Model is an approach to the issue of the representation of the document. This approach is quite popular in the algorithms of grouping. It is assumed that each document is represented by a vector of numbers that determine the occurrence of the keyword in the document. With respect to the m -element set of documents there is a matrix $M_{m \times n}$ (TFM, term frequency matrix) in which each document d_i is represented by the n -dimensional vector of the occurrence of a term t_i . In the vector space representation of text documents, t_i expresses the weight (a certain real number) that shows how many times a given term appears in the document or in the whole set of documents. The weight $w_{tf-idf}(1)$ assumes high values for those terms that occur in a small number of documents, the medium values when the term appears frequently in many documents and rarely in a few documents, and low values for the terms found in almost all documents. These properties contribute to the diagnosis of specific terms.

$$\begin{aligned} w_{tf-idf}(t_i, d) &= w_{tf}(t_i, d) * w_{idf}(t_i, d) = \\ &= w_{tf}(t_i, d) * \log(N/n_j) \end{aligned} \quad (1)$$

where: w_{tf} – term frequency, t_i – number of occurrences of the term in the document d ; N – is the number of documents in a collection; n_j – is the number of documents that contain the phrase t_j .

2.2. Latent Semantic Indexing (LSI)

Latent semantic analysis is a theory and mathematical method, whose mission is the extraction of knowledge about the contextual meaning of words from large text bodies, help in solving the problems of synonymy (several words with the same meaning) and polysemy (one word with many meanings) in systems of information extraction.

LSI (Latent Semantic Indexing) algorithm uses a factor analysis, a method that allows reducing the large number of variables (terms from term frequency matrix-TFM) to a number of uncorrelated factors. Factor analysis is conducted with SVD decomposition (Singular Value Decomposition) which essence is to compute the singular value decomposition of the matrix A (containing vector representations of documents i.e. TFM) of dimensions $m \times n$ (it can be assumed without the loss of generality that $m > n$). Proceeding in this way reduces the dimensionality of the matrix of the terms, and hence the redundancy in the description of the documents. Importantly, this method entails only small loss of information. There are new expressions (uncorrelated factors)-*pseudowords*, which contain in themselves the meaning of words relating to the same issues (e.g. phrases such as ingot, mould, metallic die, cast and mould core define the casting issues). More details related to the discussed method can be found in the publications [9,22].

2.3. Rough sets

The aim of the rough set theory is to create a new mathematical model describing imprecise and inaccurate data. The advantage of the rough set theory, in the case of data analysis, is the possibility to search the relationships between the attributes that characterize objects and to generate rules that describe the possible relationships between them. According to the traditional concept a set is precisely defined by its components. In the case of a rough set theory we do not have a clear understanding of its elements, and the set is described by attributes placed in the decision table [23]. This method of defining a set implies the potential for ambiguity, hence the rough set is composed of the lower and upper approximations. All objects decision table (elements of the set) are determined by a binary *Indiscernibility relation* (IND), which is key in rough set theory [24]. Its definition is as follows: Let it be a decision table $DT = U, C, D, V, f$, where $C \cup D = Q$ and any subset of attributes $P \subseteq Q$. Indiscernibility relation is:

$$IND(P) = \{(x, y) \in U : \forall a \in P \rightarrow a(x) = a(y)\} \quad (2)$$

The relationship defines the objects (elements) that have the same values for all attributes in the subset. In other words, we cannot clearly distinguish between data objects based on separate attributes. In addition, this ratio divides us considered a collection of a group called disjoint classes of abstraction. The fact that we strive to create the most general rules of decision, the best solution is to use a set of *reducts* of the decision table, not

the entire set of conditional attributes. *Reduct* is the minimum subset of the set of conditional attributes C , which preserves the distinguishing of objects relative to a set of decision attributes D . It is defined as follows:

$$RED(C) = \left\{ X : X \subseteq C \rightarrow \gamma_{IND(C)}(D) = \gamma_{IND(X)}(D) \right\} \quad (3)$$

It is a set, which cannot be reduced without losing the identification of objects. The minimum subset of attributes is created in the goal of creating the least amount of decision rules [25].

2.4. Association rules learning

Association rules learning is one of the methods for data mining, which is based on creating sets of association rules describing the collection. Association analysis is one of the main methods used in text mining because of the usefulness of the method in the processing of nominal variables. Apriori algorithm and all its modifications are the best known implementation of association rules learning. Apriori algorithm is based on a determination of *frequent keyword sets* (keyword combinations) and generating rules from them in the second step of the algorithm. The *frequent keyword set* is a set of common terms whose *support* (the number of occurrences of the phrases set in relation to the total number of transactions-documents) exceeds the required level. Each set has calculated support. If it reaches a minimum value set by the user, the frequent set creates a rule. Each of the frequent set that meet the support is re-used to create more and more of k -elements frequent sets that are checked again in the context of the support. Apriori algorithm is the simplest realization of the idea of associative analysis, but its weakness is the computation time, hence the need for more efficient algorithms as AprioriTid or algorithm FP Growth, which, by introducing additional database structure enables the omission of certain transactions [26]. All these algorithms have one goal-to determine what words have the property of “co-existence” in the individual documents, and on this basis to define groups of documents and the content.

2.5. Ontology

Ontologies are taken from the philosophy with full consciousness of the objective they are supposed to meet - a simplified model of the reality. An *ontology* is a model of a fragment of domain knowledge consisted of a set of concepts and a set of predicates: properties of concepts being unary predicates and a set of relationships between concepts being binary predicates. An ontology describes objects and relationships, resulting from the knowledge agreed by the group rather than individual experience [27-30]. An *ontology's domain* is a range of knowledge concerning one or more issues that are in a scope of interest because of the purpose of the model. The *domain ontologies* refer to specific fields, these are ontologies for the different fields

of knowledge, adapted to functional needs and requirements of the application, in which they are to be used. Thus there is a need to create tools for the integration of ontologies [31-32]. Ontology is not a database schema, but a simplification is also possible: an ontology for the repositories of knowledge is what the diagram of entity is for a database; it is a schema, a model of a certain field of knowledge, readable by both computers and people. Creating an unified, formal language of related-concepts definition and its further sharing with a number of users is a first and most important goal of ontologies [33-35].

3. Investigation results

The methods described in the preceding paragraphs were used in the creation of a system for classifying and semantic integrating documents. This system is a result and an extension of former studies of the Author described in [22]. The created program includes five components: (1) latent semantic indexing module, (2) rough sets document classification module, (3) apriori document classification module, (4) system of semantic integration of documents, (5) ontology integration module. The premise of the system is automatic scanning and cataloguing of documents using the described algorithms in such a way that they can be integrated later with the ontological model. The product of the system operation is a file with the ontology enriched with integrated documents. Components of the system are shown in figure 1.

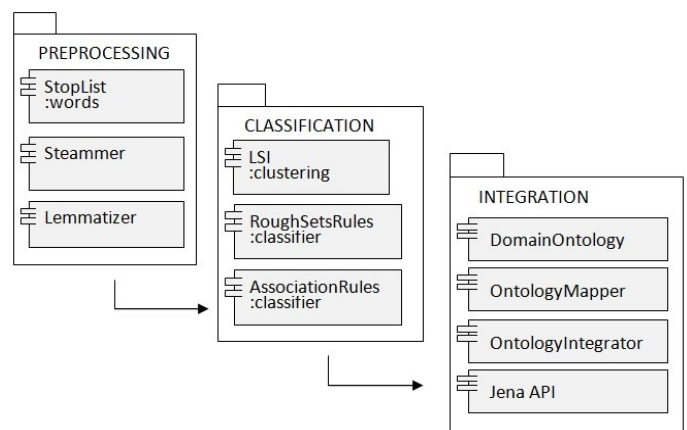


Fig. 1. Components of the system for classifying and semantic integrating documents

RoughSetsRules:classifier. The module for classifying documents based on the theory of rough sets on the input has a database entry, which is a specially designed decision table generated on the basis of a set of documents – Polish-language publications in the field of metallurgy. The decision table contains the conditional attributes (keywords) and a decision attributes (class of documents), under which the system performs an analysis of the newly added document. The block called **RoughSetsRules:classifier** include all the necessary algorithms for defining the *cores*, *reducts* and the *rules* that generated the

decision. To obtain the most general set of rules reducts of the decision table should be calculated. The module uses the method for determining reducts based directly on the approximation of the rough set so that the computational complexity of the algorithm used is smaller.

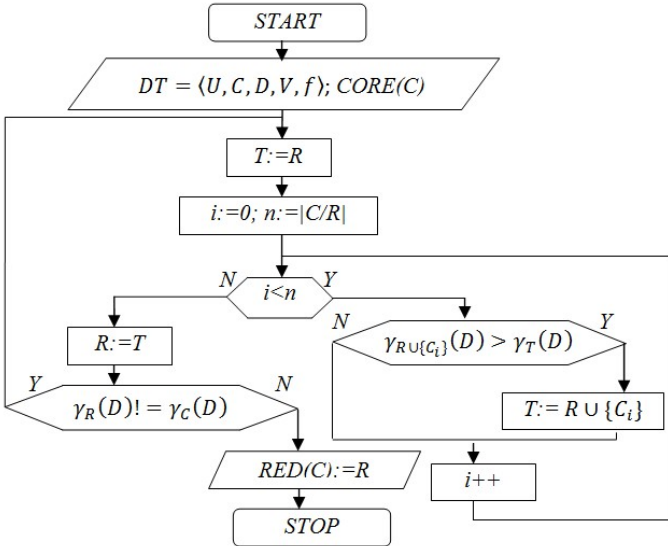


Fig. 2. The algorithm for creating reducts for the input decision table

Figure 2 presents a scheme of an algorithm for creating reducts for the input decision table. The algorithm assigns elements that are a core to the set of reducts because these are the attributes that affect the decision certainly. Then goes a block of instructions until the condition:

$$\gamma_R(D) \neq \gamma_C(D) \quad (4)$$

If the above inequality is true, or if the quality of the approximation of set of reducts is different from the quality of the approximation of all the conditional attributes, then the conditional block is executed. After checking all the attributes, elements of the calculated subset are added to reducts and then the main condition (4) is checked. When it is false-algorithm exits and results on output with calculated set of $RED(C)$, being a minimum set of attributes for the decision table DT [11]. The final stage is to determine the classification decision rules. Based on the determined reducts of decision table, columns that the set of $RED(C)$ does not contain are removed. Decision rules are presented in the form of implications, making it easy to attach to a decision depending on the value generated reducts. Example of generated rule might be: *IF 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0 THEN "heat treatment"* where a string of bits indicates the occurrence of keywords from the decision table and the THEN clause is followed the conclusion being a class which will be assigned the document.

AssociationRules:classifier module is based on four phases: (1) Preparation of data for learning; (2) Load data and their extraction, (3) Application of the algorithm TF-IDF / Apriori and the *Association rule learning*, (4) Presentation of results.

The user specifies: (1) *length filter* of words handled by the program. If after the extraction process the word will be shorter

than the size indicated by the user it would not be added to the database. (2) When too strong factor *min-frequency* will eliminate the least frequently occurring words making use of the following algorithms may be shortened, but it can also spoil the results, if the keyword appears in the document is rare, so use this parameter with caution. (3) Determination of the *support* affects the result Apriori algorithm, which, depending on this factor creates a frequent keyword set. (4) *Confidence* determines the credibility of the rule set, which is the result of the algorithm Apriori. (5) *Support tf-idf* narrows words appearing in the documents such that there are at least 0.00-0.10 results obtained after normalization *tf-idf*; setting the indicator to 0: not filtered at all, and at least 0.01: rejects the word in small number of times in each of the documents in the group. Classification conducts by using generated for each category association rules to group "unknown" documents. The documents are grouped by scanning them in sets of words occurring in every single association rule. The more rules appear in the collection of disaggregated words, the program increases the probability factor for the document that can belong to a group of documents in which it was previously found association.

LSI:clustering module. The components responsible for processing documents (LSI) and their integration with ontology (semantic integration of documents **OntologyMapper**) work closely together. For their operation, modules that are responsible for the analysis of documents and their integration do not demand from the user making any major operations. In the automatic way they assign documents to be catalogued to ontology and write the modified ontology to a new file. More about the operation of these modules, the reader would find in the position [22]. For the example, to the class called *Decarburization* has been assigned the document called *Fundamentals of heat treatment*. In addition, as a result of the operation, a new ontology called *NewOntology* was created, while the base ontology, i.e. *ontology_cast iron*, has not been overwritten as a result of the program operation.

Managing classifications from different domain ontologies may be supported by the module **OntologyIntegrator**. It provides solutions for the comparison and integration of the two existing ontology into one. An important role for the work of the algorithm plays a *ontologies convergence factor* specifying the extent to which ontologies should be similar to each other, to allow for their integration. It is a factor set by the user. The algorithm has three modes: (1) *Class Only*: Ontologies are compared with each other on the basis of class names, this is the easiest and fastest method of analysis, but its drawback is the inability to detect changes in the properties of classes or change the number of objects which are not classes. (2) *Standard Algorithm*: analyzing ontologies compares not only the names of the classes, but which are also included in these properties. This allows a broad comparison of ontology. (3) *Full*: algorithm counts and compares the value of all the triples <Subject, predicate, object>, from which the ontology is built. In this mode, the algorithm compares each part of both ontology. As the example of ontology integration two ontologies – white cast iron and grey cast iron ontologies – containing a small amount of common classes, but having common mechanical properties are not possible to

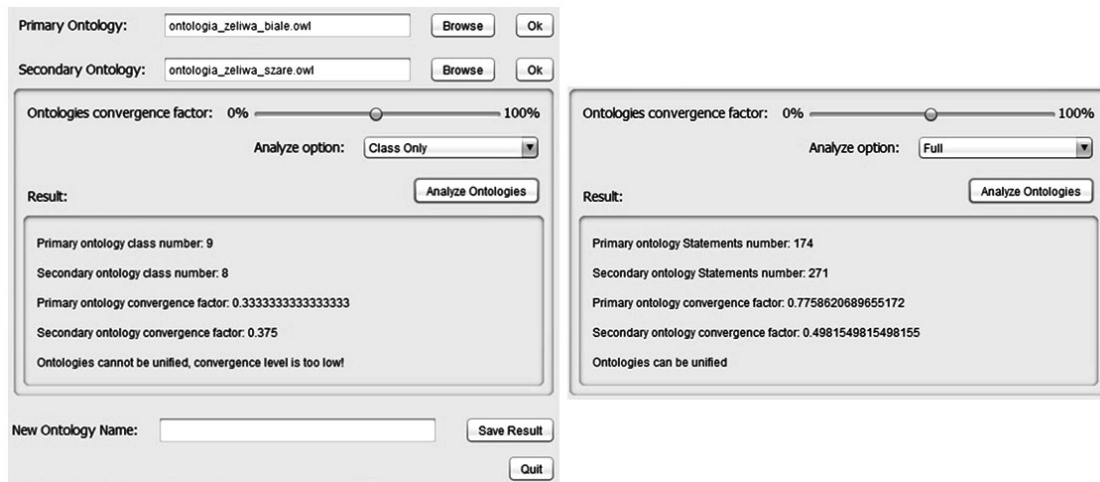


Fig. 3. Example of operation with OntologyIntegrator module

integrate on the basis of names of classes only. Because of the common characteristics of classes analyzed it was possible to integrate with option of class&properites (2: *Standard Algorithm*) mode or with full&triple (3: *Full*) mode of operation. Example is presented in the figure 3.

The results of the analysis indicate the need for a user raised awareness about the actions taken by him. Of great help here is certainly a clear definition of objectives and knowledge of the contents of the compared ontologies.

4. Conclusions

The advantage of search for documents based on classifiers over the methods based on keywords is finding documents indirectly related, expressed in words other than the query but referring to a similar subject. It also allows the creation of rankings of items found, so that the user receives the results sorted based on relevancy of the search.

Issues related with data integration and interchange of data among heterogeneous programming artifacts are today very modest due to its importance in complex and distributed systems. Ontologies are able to solve problems related with integration of knowledge, and even more: could help in classification when the accessible knowledge is incomplete and inconsistent [36].

Evaluation of the quality of search on the basis of classifiers depends on many factors, including the number of documents in the training set, the similarity of subject matter of these documents and specified classes, the number of classes, and finally, the quality of the documents. Using a smaller amount of data with quite dissimilar types of categories allows you to adjust searched documents from almost one hundred percent accuracy. Using a very large number of learning documents but equally more categories can be seen that the results are much worse, especially if the categories are similar to each other.

The use of three algorithms for classification and clusterization of documents obtained meaningful search quality. While the use of ontologies to integrate document repository lets you

standardize and universalize categorization of documents. Possibility of ontology integration provides functional support and managing the document repository that has the ability to provide decision support.

Acknowledgements

The work has been supported by the Polish Ministry of Science and Higher Education-AGH University of Science and Technology Funds No. 11.11.110.300.

REFERENCES

- [1] E. Nawarecki, S. Kluska-Nawarecka, K. Regulski, Multi-aspect character of the man-computer relationship in a diagnostic-advisory system, *Human-computer systems interaction: backgrounds and applications 2*, eds. Z.S. Hippe, J.L. Kulikowski, T. Mroczek, 2012 Berlin; Heidelberg: Springer-Verlag.
- [2] J. David, P. Svec, R. Frischer, R. Garzinova, The Computer Support of Diagnostics of Circle Crystallizers; *Metalurgija* **53** (2),193-196 (2014).
- [3] P. Malinowski, J.S. Suchy, J. Jakubski, Technological knowledge management system for foundry industry, *Archives of Metallurgy and Materials* **58** (3), 965-968 (2013).
- [4] Z. Gronostajski, M. Hawryluk, M. Kaszuba, M. Marciniak, A. Niechajowicz, S. Polak, M. Zwierzchowski, A. Adrian, B. Mrzyglod, J. Durak, The expert system supporting the assessment of the durability of forging tools, *International Journal Of Advanced Manufacturing Technology* **82** (9-12), 1973-1991 (2016). DOI: 10.1007/s00170-015-7522-3
- [5] A. Maciol, P. Maciol, S. Jedrusik, J. Lelito, The new hybrid rule-based tool to evaluate processes in manufacturing, *International Journal Of Advanced Manufacturing Technology* **79** (9-12), 1733-1745 (2015).
- [6] C.D. Manning, P. Raghavan, H. Schutze. *An Introduction to Information Retrieval*. 2008.

- [7] D. Wilk-Kołodziejczyk, A. Opalinski, E. Nawarecki, S. Kluska-Nawarecka, Exploration of Web resources in the domain of metal processing technologies, *Metalurgija* **55** (1), 127-130 (2016).
- [8] R. Agrawal, R. Srikant, Fast algorithms for mining association rules in large databases. in: *Proceedings of the 20th International Conference on Very Large Data Bases, VLDB*, p. 487-499, Santiago, Chile, (1994).
- [9] S. Deerwester, S. Dumais, T. Landauer, G. Furnas, R. Harshman, Indexing by latent semantic analysis, *Journal of the American Society of Information Science* **41** (6), 391-407 (1990).
- [10] D. Miao, Q. Duan, H. Zhang, N. Jiao, Rough set based hybrid algorithm for text classification, *Expert Systems with Applications* **36** (5), 9168-9174 (2009).
- [11] Z. Pawlak, Rough set theory and its applications, *Journal of Telecommunications and Information Technology* **3**, 7-10 (2002).
- [12] Y. Leung, M. Fischer, W.Z Wu, J.S. Mi, A rough set approach for the discovery of classification rules in interval valued information systems, *International Journal of Approximate Reasoning* **47** (2), 433-246 (2008).
- [13] R.K. Roul, S.K Sahay, An Effective Approach for Web Document Classification using the Concept of Association Analysis of Data Mining, *International Journal of Computer Science Engineering and Technology* **3** (10), 483 (2012).
- [14] Y. Hirate, H. Yamana, Generalized Sequential Pattern Mining with Item Intervals, *Journal Of Computers* **1** (3) (2006).
- [15] J. Jakubski, P. Malinowski, S.M. Dobosz, K. Major-Gabryś, ANN Modelling For The Analysis Of The Green Moulding Sands Properties, *Archives of Metallurgy and Materials* **58** (3), 961-964 (2013).
- [16] K. Smyksy, E. Ziółkowski, R. Wrona, M. Brzeziński, Performance evaluation of rotary mixers through monitoring of power energy parameters, *Archives of Metallurgy and Materials* **58** (3), 911-914 (2013).
- [17] I. Olejarczyk-Wożeńska, A. Adrian, H. Adrian, B. Mrzygłód, Parametric representation of TTT diagrams of ADI cast iron, *Archives of Metallurgy and Materials* **57** (2), 981-986, (2012).
- [18] K. Regulski, D. Szeliga, J. Kusiak, Data Exploration Approach Versus Sensitivity Analysis for Optimization of Metal Forming Processes, *Key Engineering Materials* **611-612**, 1390-1395 (2014).
- [19] Z. Glowacz, Recognition of Acoustic Signals of Loaded Synchronous Motor Using FFT, MSAF-5 and LSVM, *Archives Of Acoustics* **40** (2), 197-20, (2015). DOI:10.1515/aoa-2015-0022
- [20] A. Glowacz, Z. Glowacz, Recognition of thermal images of direct current motor with application of Area Perimeter Vector and Bayes Classifier, *Measurement Science Review* **15** (3), 119-126 (2015). DOI: 10.1515/msr-2015-0018
- [21] M. Berry, S. Dumais, G. O'Brien, Using Linear Algebra for Intelligent Information Retrieval, *SIAM Review* **37** (4), 573-595 (1995). DOI: 10.1137/1037127.
- [22] S. Kluska-Nawarecka, K. Regulski, M. Krzyżak, G. Leśniak, M. Gurda, System of semantic integration of non-structuralized documents in natural language in the domain of metallurgy, *Archives of Metallurgy and Materials* **58** (3), 927-930 (2013). DOI: 10.2478/amm-2013-0103.
- [23] R.W. Świdniarski, Rough sets methods in feature reduction and classification, *International Journal of Applied Mathematics and Computer Science* **11** (3), 565-582 (2001).
- [24] R. Colomo-Palacios, I. González-Carrasco, J.L. López-Cuadrado, A. García-Crespo, ReSySTER: A hybrid recommender system for Scrum team roles based on fuzzy and rough sets, *International Journal of Applied Mathematics and Computer Science* **22** (4), 801-816 (2012).
- [25] S. Kluska-Nawarecka, D. Wilk-Kołodziejczyk, K. Regulski, G. Dobrowolski, Rough sets applied to the RoughCast system for steel castings, *Intelligent Information and Database Systems*. in: *Springer Lecture Notes in Computer Science*, 6592/2011, 52-61, (2011). DOI: 10.1007/978-3-642-20042-7-6
- [26] M.J. Zaki, SPADE: An efficient algorithm for mining frequent sequences, *Machine Learning* **42**, 31-60, (2001).
- [27] S. Staab, R. Studer, *Handbook on Ontologies*, 2004 Springer-Verlag, Berlin.
- [28] V. Avram, D. Rizescu, A Domain Ontology For Data Collections Of The Accounting System, in: *International Conference On Informatics In Economy*, Eds. Boja C., Batagan L., Doinea M., et al., 347-351, Romania 2013.
- [29] J. Broekstra, M. Klein, S. Decker, D. Fensel, F. Harmelen, I. Horrocks, *Enabling knowledge representation on the Web by extending RDF Schema*, Stanford University, 2000 USA.
- [30] S. Grimm, P. Hitzler, A. Abecker, *Knowledge Representation and Ontologies*, *Semantic Web Services: Concepts, Technologies and Applications* **3**, 51-106 (2007).
- [31] A. Maciol, R. Wrona, A. Stawowy, P. Maciol, An attempt at formulation of ontology for technological knowledge comprised in technical standards, *Archives of Metallurgy And Materials* **52** (3), 381-388 (2007).
- [32] M.N. Ahmad, R.M. Colomb, M.S. Abdullah, *Ontology-Based Applications for Enterprise Systems and Knowledge Management*, 294-320 (2013). DOI: 10.4018/978-1-4666-1993-7
- [33] N. Konstantinou, D.E. Spanos, M. Chalas, E. Solidakis, N. Mitrou, *VisAVis: An Approach to an Intermediate Layer between Ontologies and Relational Database Contents*, in: *Web Information Systems Modeling*, Luxemburg 2006.
- [34] R. Poli R, *Theory and Applications of Ontology*, *Computer Applications*, 3-12 (2010).
- [35] Z. Xu, S. Zhang, Y. Dong, Mapping between Relational Database Schema and OWL Ontology for Deep Annotation, in: *2006 IEEE/WIC/ACM International Conference on Web Intelligence*, 548-552, Washington, DC, USA 2006.
- [36] S. Kluska-Nawarecka, D. Wilk-Kołodziejczyk, K. Regulski, Practical aspects of knowledge integration using attribute tables generated from relational databases, in: *Semantic Methods for Knowledge Management and Communication*, **381**, 13-22, Springer 2011, Berlin.