How Specific Can We Be with k-NN Classifier?

KAROL DRASZAWKA
Gdańsk University of Technology
Faculty of Electronics
Telecommunications and Informatics
Narutowicza St. 11/12, 80-233 Gdańsk
POLAND
kadr@eti.pg.gda.pl

JULIAN SZYMAŃSKI
Gdańsk University of Technology
Faculty of Electronics
Telecommunications and Informatics
Narutowicza St. 11/12, 80-233 Gdańsk
POLAND
julian.szymanski@eti.pg.gda.pl

Abstract: This paper discusses the possibility of designing a two stage classifier for large-scale hierarchical and multilabel text classification task, that will be a compromise between two common approaches to this task. First of it is called *big-bang*, where there is only one classifier that aims to do all the job at once. *Top-down* approach is the second popular option, in which at each node of categories' hierarchy, there is a flat classifier that makes a local classification between categories that are immediate descendants of that node. The article focuses on evaluating the performance of a k-NN algorithm at different levels of categories' hierarchy, aiming to test, whether it will be profitable to make a semi-big-bang step (restricted to a specified level of the hierarchy), followed by a middle-down more detailed classification. Presented empirical experiments are done on Simple English Wikipedia dataset.

Key–Words: Hierarchical categorization, k-NN, Multi-label classification, Large Scale Classifier

1 Introduction

One of the biggest challenges in the field of data mining is text classification [1]. Given a text written in natural language, text classification aims to assign one or more category labels that best describe the content of the text. The problem is challenging because the most known data mining techniques require data in the form of vectors of numerical features and it is difficult to represent natural language using such a restricted representation without loosing much information. Nevertheless, a simple Bag-of-Words representation with properly chosen feature weighting scheme, combined with statistical methods like LSA, pLSA or LDA [2, 3, 4] is often satisfactory, when the number of documents per subject category in the training set is large enough.

Nowadays however, the challenge is set four steps further.

- First, is the implementation of text classification task for a large scale repositories. This means that the number of documents to classify is counted in thousands, if not millions, and so the number of categories. As an example, the problem of automatic classification of English Wikipedia articles involves processing the repository that contains over 3 millions of documents and over 300 thousand categories.

- Second, the cardinality of classes' elements is unequal. Such unbalance distribution stems from 'horizontal differences', i.e. unrelated topic areas of various sizes, as well as 'vertical differences', which means varying specificity of topics and subtopics – categories can be organized in a hierarchical structure.

- Third, the task of text classification is often multi-label, that means a document can be associated with more than one category.

- Fourth, as there are large number of classes, the classification task should take into account relations between categories that form a structure that organize them. This is not taken into consideration in a typical classification, where classes are treated as independent from one another.

Among many attempts coping with such a large scale, unbalanced, multi-label and hierarchical text

ISBN: 978-960-474-379-7

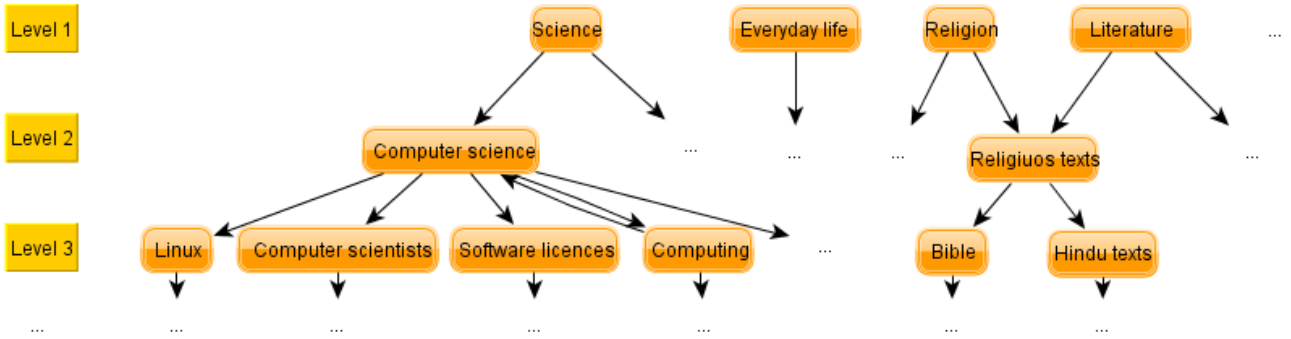


Figure 1: A small snippet from pseudo-hierarchical structure of Simple English Wikipedia categories with depth level indication. It presents one of many cycles (between *Computer Science* and *Computing* categories) that have to be cut in order to obtain a real hierarchy. An example of a category (*Religious texts*) with multiple supercategories is also shown.

classification problems [5, 6, 7], one of the most promising solutions, in respect to performance quality and robustness, is the use of two-stage classification [8]. In the first stage, a fast classifier performs a rough classification returning a list of categories, that have relatively high probability of being in the set of correct labels for a given document. After that, the second stage is used to fine-tune the previous rough estimate employing some more sophisticated and computationally more demanding classification algorithms.

If it comes to the choice of a first-stage classifier, a modified version of a well known *k* nearest neighbor algorithm is among the most reasonable ones. It does not require learning phase and it can have fast implementation using inverted indexes [9] and/or some parallelization techniques [10].

In this article, we present our experiments with a multi-label version of k-NN classifier aiming at measuring its classification performance at different level of thematic categories' specificity. The next section describes the problem. Section 3 explains the method that was used to evaluate k-NN. Results of that evaluation are given in section 4, after which the last section contains some conclusions and final thoughts.

2 Problem description

If categories are organized in a hierarchical-like structure, such as shown in Fig. 1, the *specificity* of a category can be understood as the depth level of this category in a directed acyclic graph (DAG) of the hierarchical structure. For example, category "Linux" from level 3 of the hierarchy is more specific than category "Computer Science" (lev. 2) which in turn is more specific than category "Science" (lev. 1). Of course, defining specificity of a category in terms of its position in a more or less arbitrary hierarchy could lead to paradoxical and even wrong assessments (especially when some branches in a hierarchy are much deeper than other), but this approach can be seen as a first rough approximation.

In hierarchical classification, there are two frequently employed approaches [11]:

1. *Big-bang approach*, in which there is only one global classifier, that tries to assign to a test document labels from all the levels of specificity at once.

2. *Top-down approach*, where the task is decomposed into smaller, 'flat' classification problems, at each level of a hierarchy, so that there are many 'local' classifiers involved.

As many studies show, the results of a top-down strategy is often impaired by the so called 'error propagation' effect, i.e. the fact that errors made by classifiers at higher levels of a hierarchy are repeated at lower levels as well. On the other hand, the downside of a big-bang approach is that one global classifier is hard to train and this results in poor classification performance.

ISBN: 978-960-474-379-7

As indicated above, one of the promising attempts to large scale text documents classification is the use of a two stage classifier, especially with k-NN-like algorithm in its first phase. In the context of the two approaches to hierarchical classification, the roles of system's two stages can also be of two types. In the first one, k-NN classifier tries to select a reasonable number of the most promising candidate categories from all the levels of specificity, and the second stage is used to validate those choices by making a binary decision about each candidate category, whether it should be added to final output or not. This is a two-staged big-bang approach to large-scale classification. The other one is to restrict the categories from which the first stage classifier can choose candidates only to those with low specificity. This is done in order to minimize classification errors at the first stage, so that the second stage can be used not to validate previous choices, but to give more precise description of a document by incorporating local classifiers. This is therefore partially hierarchical top-down approach.

That second view reveals the fact that a two-staged system can be a good architecture for a classifier that incorporates a policy that would be a golden mean between two extremes, i.e. big-bang and fully local. In the first stage, k-NN performs a big-bang classification, but only to a limited level of specificity, so that its performance is satisfactory. Next, in the second stage local classifiers make detailed classification in a top-down fashion, but because part of the work is done by k-NN, the problem of error propagation would be limited.

The problem is then to examine how specific can k-NN be, i.e. to which level of categories' specificity the classification performance of k-NN is reasonably enough to be used as a first, big-bang step in the two-staged classification system. Such system aims to be a balance between big-bang and top-down approaches to hierarchical classification.

3 Evaluation method

To answer the above question, we performed experiments on the text corpus extracted from the whole Simple English Wikipedia[1] processed with Matrix'u application [12]. All articles are represented using a standard bag-of-words technique with supervised *Confidence Weight* feature weighting [13]. We did some initial filtering, which includes removal of categories that have less than 5 articles, cutting all the cycles that appear in the category structure and removal of stop words and words that appeared only once in the whole corpus. After that, the dataset consists of 55637 articles and 5679 categories. The categories form a DAG with 8 supercategories as the root nodes (level 1 of specificity) and other ones being their descendants up to 13 levels of specificity.

The global classifier which was evaluated at different category levels is our implementation of the 2012 Pascal Challenge on Large Scale Dowinning algorithm [5]. In this paper, we did not aim at improving the performance of it (one way of such improvement is published in [14]), and used the original parameters of the classifier, which were not modified (most importantly: k parameter of the k-NN is set to 30, and α parameter of the scoring phase is set to 3.0).

The classifier was employed once, in a leave-one-out fashion on the whole dataset. All the results presented in the next section rely on evaluating these k-NN predictions. This evaluation was performed using macro- and micro-averaged label based measures (precision, recall and their harmonic mean – F1-score) [15].

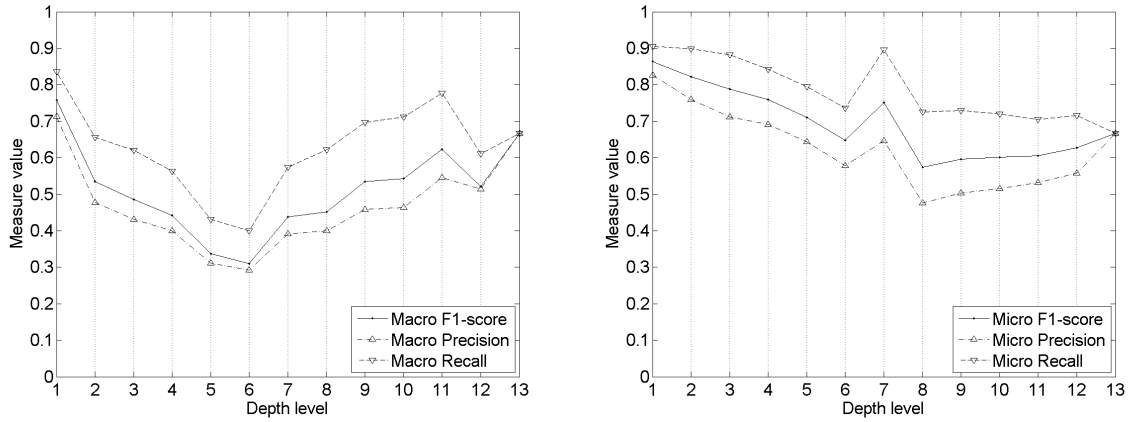
In the calculation of the classification performance at a given hierarchy level, only categories that are exactly at the level under investigation are treated as relevant. The decisions made by the k-NN classifier about associations with categories that are out of this set are not passed to evaluation measures. At each level, classifier decisions regarding each of 55637 articles are evaluated. The classification works with the assumption that when an object belongs to some class it also belongs to all its ancestor classes. Therefore, if the k-NN assigns a category (or many categories) to a document, it is automatically also assigned to all supercategories of that category (categories).

4 Results and discussion

Figure 2 presents results given by the described evaluation method. Initially, the results indicated by macro-averaged label based measures (fig. 2a) seem to be surprising. In its left part, the plot is as expected: with the increase of categories specificity, the quality of classification is getting worse, each measure indi-

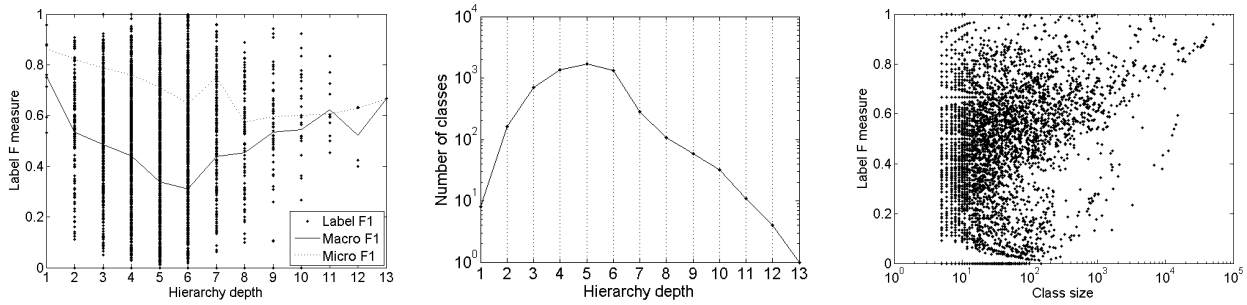
[1] *http://simple.wikipedia.org*

ISBN: 978-960-474-379-7



(a) Macro-weighed measures. (b) Micro-weighed measures.

Figure 2: K-NN performance depending on categories' hierarchy depth.



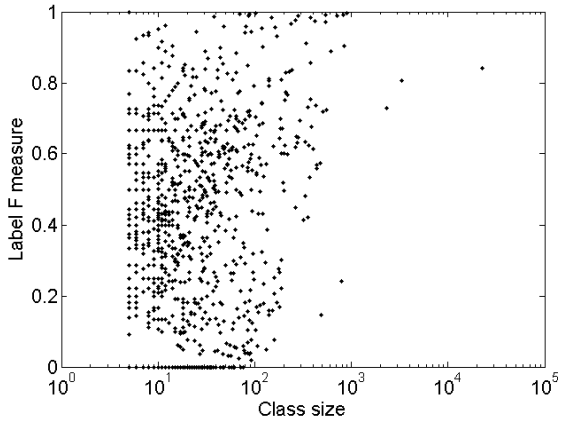
(a) (b) (c)

Figure 3: Details of the results: (a) F1-scores of each class per hierarchy depth with macro- and micro-averages of them, (b) the number of categories per hierarchy depth, and (c) F1-scores of each class per its size (log-scale on the horizontal axis).

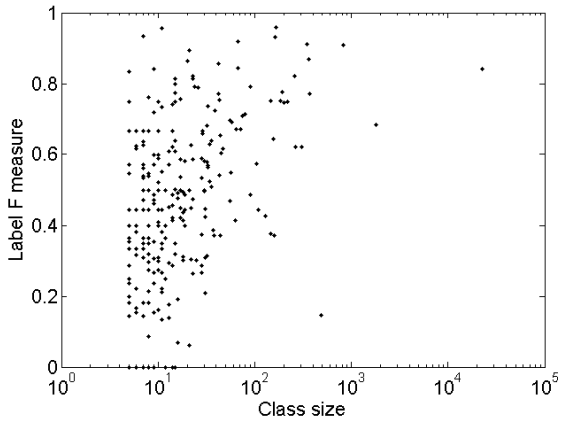
cates this. However, after the 6th level of depth, the classification quality increases, so that more specific categories seem to be predictable better. The same effect can be found by looking at micro-averaged measures (fig. 2b), although here it is less apparent. The second observation is that the micro-averaged measures have higher values than the macro-averaged. The third thing worth noticing is the pick at 7th level of micro-averaged measures. To explain these phenomena, more detailed analysis of the results has to be done.

Additional characteristics of obtained results are presented in fig. 3. Each point in fig. 3a corresponds to the value of F1-score calculated for each category separately. Macro-averaged and micro-averaged F1-score values are plotted here once again to show the differences between them. Macro-averaging is simply the arithmetic average of all classes' F1-scores at a given hierarchy depth. Micro-averaging involves averaging the number of individual classifier decisions (i.e. about assigning a label to an article or not) and calculating F1-score based on that averages. Because *true negatives* do not impact the F-measure and this is the most common type of classifier decisions for small classes (most objects should *not* be assigned to such classes and a classifier correctly does not do this), and because *true positives* and *false negatives* do impact the F-measure and these are the most frequent classifier decisions when classifying to large classes, the micro-averaged label-based F1-score has the property

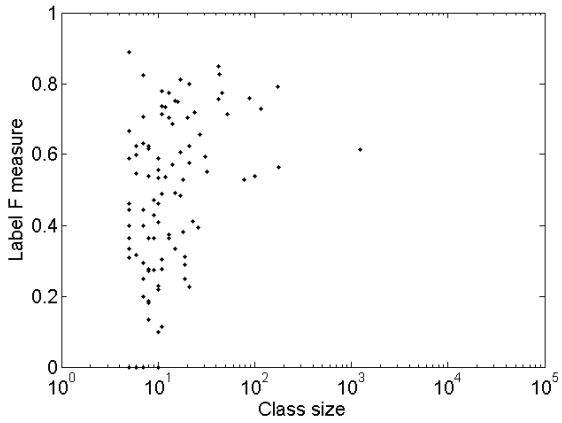
ISBN: 978-960-474-379-7



(a) Level 6 classes.



(b) Level 7 classes.



(c) Level 8 classes.

Figure 4: F1-scores of each class per its size (log-scale on the horizontal axis): (a) classes from level 6, (b) classes from level 7, (c) classes from level 8.

of being determined largely by classifier performance on classes that have big sizes. The performance of k-NN classifier for big classes, which are present especially at levels 1 to 7, is relatively good (above 0.8 for the largest, as is depicted in fig. 3c), and this is the reason why the micro-averaged measures at that levels indicates much better classification performance than macro-averaged measures.

Depending on the depth level, the number of categories changes significantly. This change is visible in fig. 3a as much more densely located points at levels 4 to 6 than at other levels, but is additionally shown in fig. 3b. The main 8 categories from the first level are gradually branching, initially causing the number of categories to grow exponentially. However, most of category branches goes not deeper than 5th or 6th level. After that level, the number of categories drastically decrease, with only one category at the last 13th level.

It can be seen that the macro-averaged measures are mostly influenced by the number of classes, that k-NN has to choose between. If there are many classes (as on levels 4 to 6), the classification is obviously more difficult and the results are worse. This also affects micro-averaged metrics, but here the effect is less harmful, because micro-averaging is less dependent on the number of classes (many of which are of small size), but on the performance on the biggest ones.

To explain why there is a pick in the micro-averaged performance measures at 7th level of the hierarchy (which is, interestingly, precisely the middle depth level), detailed scatter plots are shown in fig. 4. This is the same type of illustration as presented in fig. 3c, but the classifier performance for categories only from levels 6, 7 and 8 are pointed out. It can be seen that level 6 contains a big number of classes. Although there are some very large classes and their F1-scores are high, most of them have small size and, moreover, substantial part of them has very low F1-score, even 0.0. This biases micro-averaging towards low value. Level 7 still contains big classes (esp. the one with over 20000 articles) with high F1-scores, and, at the same time, has many fewer small, mistakes-generating classes. This is why the averaged result is much higher at that level. On the other hand, level 8 has even smaller number of small categories that k-NN has problems with, but here there are no big and easy to classify categories, that pull up the micro-

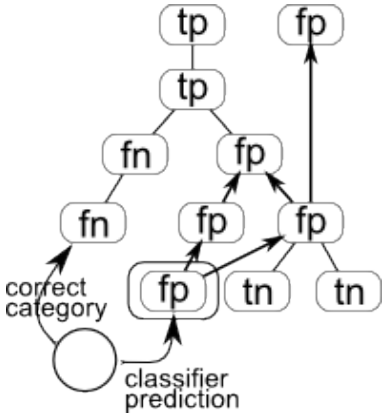


Figure 5: Upward error propagation example: *fp* – false positive error, *fn* – false negative error, *tp* – true positive decision, *tn* – true negative decision.

averaged measures at previous, higher levels. The middle depth level of the hierarchy is then a "golden middle" that already does not have too many small and difficult categories (many hierarchy branches are ended here), but still does contain large and easy categories.

Upward error propagation

There is one more, very important negative effect that takes place, when a *big-bang* classifier is used to large scale hierarchical classification task. It can be called *upward error propagation*, and should be seen as a counterpart of the well known error propagation effect in the local, top-down approach. Upward error propagation occurs when a global classifier, such as our k-NN, makes a mistake at some level x. Then (fig. 5), because of the assumption that if a document belongs to a category, it automatically belongs to all the ancestors of that category, it is propagated upwards to higher levels $x-1$, $x-2$ etc. up to the level, at which one of the original categories of the document has the common ancestor node with the mistaken category. In the worst case, the error can be propagated to the root level 1. This propagation effect is even more damaging when categories can have more than one direct supercategories, as it is in our case of Wikipedia categories.

Described effect is very harmful to the tested k-NN and Wikipedia dataset. The most extreme example of its influence is the case of an article titled *'Rye House railway station'* classification, which originally is assigned to a third level category *'Transportation stubs'*, and therefore is connected also with ancestors of that class: *'Transportation'* and *'Everyday life'*. k-NN classifier, as it has generally problems with 'stubs', because of their very short articles, assigns wrongly the label *'Human behavior'* from 12th level. This false positive decision was propagated all the way up through forking paths to roots of the hierarchy ending with 275 (!) false positives to that article.

5 Conclusions

There are two interesting findings from presented experiments.

- First, it is not the case that the classification of text documents using k-NN classifier is best when classes are of general subject and then monotonically decreasing. Hence, we cannot simply tell, up to which level of specificity it is desirable to classify using first stage classifier, and then validating and precising the results with a second stage classifier. It can be seen however, that maybe the most promising results are at the middle level of class specificity.

- The second thing is the existence of the effect, named *upward error propagation*, which is a challenging problem for those who prefer to use 'big bang' classifiers instead of top-down ones, arguing that they do not suffer from standard error propagation effect. We can say that both attempts to large scale hierarchical classification have the same problem of the influence between decisions made at one level of a hierarchy and decisions at other levels, only the direction of error propagation changes.

In the future, we plan to make more experiments aiming at establishing those preliminary conclusions. The use of other, even bigger data set, like the normal English Wikipedia can be used. We should also try to incorporate other feature weighting schemes to find how those affect the general behavior of the classification system. And, most importantly, we must develop the method, used as a second stage classifier or an additional, middle-stage, to compensate or minimize the results of upward error propagation. For that purpose,

ISBN: 978-960-474-379-7

we plan to use optimalization methods similar to proposed in [16, 17].

Acknowledgements: This work was done within grant "Modeling efficiency, reliability and power consumption of multilevel parallel HPC systems using CPUs and GPUs" sponsored by and covered by funds from the National Science Center in Poland based on decision no DEC-2012/07/B/ST6/01516.

References:

[1] M. Ikonomakis, S. Kotsiantis, V. Tampakas, Text classification using machine learning techniques *WSEAS Transactions on Computers*, Vol:4:8, 2005 ,pp. 966–974.

[2] S. T. Dumais, Latent semantic analysis, *Annual review of information science and technology*, 38(1), 2004, pp. 188–230.

[3] T. Hofmann, Probabilistic latent semantic analysis, *In Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers Inc., 1999, pp. 289–296.

[4] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation, *The Journal of Machine Learning Research*, 3, 2003, pp. 993–1022.

[5] X. L. Wang, H. Zhao, B. L. Lu, Enhanced K-Nearest Neighbour Algorithm for Large-scale Hierarchical Multi-label Classification, *Proceedings of the Joint ECML/PKDD PASCAL Workshop on Large-Scale Hierarchical Classification*, 2011.

[6] X. Han, J. Liu, Z. Shen et. al., An optimized k-nearest neighbor algorithm for large scale hierarchical text classification, *Proceedings of the Joint ECML/PKDD PASCAL Workshop on Large-Scale Hierarchical Classification*, 2011.

[7] H. H. Malik, Improving hierarchical SVMs by hierarchy flattening and lazy classification. *Proceedings of the Large-Scale Hierarchical Classification Workshop (ECIR 2010)*, Milton Keynes, UK, 2010.

[8] G.-R. Xue, D. Xing, Q. Yang, Y. Yu, Deep Classification in Large-scale Web Hierarchies, *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, 2008, pp. 619–626.

[9] F. Scholer, H.E. Williams, J. Yiannis, J. Zobel, Compression of inverted indexes for fast query evaluation *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, ACM, 2002, pp. 222–229.

[10] P. Czarnul: Modeling, run-time optimization and execution of distributed workflow applications in the JEE-based BeesyCluster environment, *The Journal of Supercomputing*, Springer, 2010, pp. 1–26.

[11] A. Sun, E-P Lim, Hierarchical Text Classification and Evaluation, *Proceedings of the 2001 IEEE International Conference on Data Mining (ICDM 2001)*, 2001, pp. 521–528.

[12] J. Szymański, Wikipedia articles representation with matrixu, *Proceedings of the Distributed Computing and Internet Technology*, Springer LNCS, 2013, pp. 500–510.

[13] P. Soucy, G. W. Mineau, Beyond TFIDF weighting for text categorization in the vector space model, *Proceedings of the International Joint Conference on Artificial Intelligence*, Edinburgh, 2005, pp. 1130–1135

[14] K. Draszawka, J. Szymanski, Thresholding Strategies for Large Scale Multi-Label Text Classifier, *The 6th International Conference on Human System Interaction (HSI)*, Sopot, Poland, 2013, pp. 350–355.

[15] G. Tsoumakas, I. Vlahavas, Random k-labelsets: An ensemble method for multilabel classification, *The 18th European Conference on Machine Learning (ECML)*, Warsaw, Poland, 2007, pp. 406–417.

[16] J. Balicki: Numerical experiments on Pareto-optimal task assignment representations by Tabu-based evolutionary algorithm, *WSEAS Transactions on Information Science and Applications*, World Scientific and Engineering Academy and Society (WSEAS), Vol. 5:5, 2008, pp. 695–705.

[17] J. Balicki: An adaptive quantum-based evolutionary algorithm for multiobjective optimization, *WSEAS Transactions on Systems and Control*, World Scientific and Engineering Academy and Society (WSEAS), Vol. 4:12, 2009, pp. 603–612.