Meta-learning Method for Automatic Selection of Algorithms for Text Classification

Karol Furdík, Ján Paralič, Gabriel Tutoky
Faculty of Electrical Engineering and Informatics
Technical University of Košice
Letná 9, 04200 Košice, Slovakia
kfurdik@stonline.sk, jan.paralic@tuke.sk, gabriel.tutoky@gmail.com

Abstract. *The paper presents a meta-learning approach for textual document classification task and an automatic selection of the best available algorithm for creation of classifiers. After brief introductory description of principles of document preprocessing, creation and evaluation of the classifiers, the meta-learning approach is presented as a method for automatic selection of the most appropriate classifier algorithm for creation of binary classifiers. Designed method, based on the modification of MUDOF (Meta-learning Using Document Feature Characteristics) algorithm, is described together with its implementation using the JBowl Java library. Finally, the experimental results achieved by the meta-learning algorithms as well as their comparisons with traditional ways used for text classification are presented.*

Keywords. Meta-learning, Classification, MUDOF algorithm, Text Mining

1 Introduction

The classification, also sometimes referenced as the categorization, is a widely used method for data analysis [4]. It is based on the supervised learning, where the goal is to distribute the objects from an input data set to the pre-defined categories. The input data set contains a sub-set of training examples, i.e. the objects categorized in advance; these training examples are processed by statistical or machine-learning algorithms to produce the so-called classification model. The resulting model can then be applied on the rest of the input set to classify the objects without known relation to the categories.

If the objects in the input data set are textual documents (or can be transferred into a textual form, e.g. by enhancing the non-textual objects by a textual description) and the content of the texts is relevant to the distribution among categories, then the process of classification is rather specific. A phase of pre-processing and text analysis is needed to identify the most relevant words, sentences, or text fragments, which have major impact to the inclusion of the text as whole to the given categories. It also affects a selection of proper classification algorithm and its settings [1].

Since the text categorization method enables to distinguish the textual documents according to the content (i.e. the meaning, topics), it was identified as a promising approach to support the semantic annotation in the virtual collaborative environment, namely in the Knowledge Practices Laboratory (KP-Lab, http://www.kp-lab.org) integrated FP6 EC project. In the KP-Lab system, the learning or working materials are semantically annotated by means of ontologies and are collaboratively investigated by students or workers in the virtual shared space. The learning materials (e.g. documents of various formats, multimedia files, concept maps, etc.) always contain textual information – directly in their contents, or indirectly in the textual descriptions given by users. Analyzing these textual descriptions, the classification procedure supports a semi-automatic annotation of the learning materials by organizing them into pre-defined categories. The conceptual framework as well as the structure of a domain

of discourse in the virtual shared space is given by various types of ontologies – e.g. taxonomies, concept maps, or domain ontologies. It means that the ontology concepts, used for semantic annotation of learning materials, can stand for classification categories. The training set is created from already annotated materials within the scope of the discourse. Insertion of a new learning material or maintenance of a domain of discourse in the virtual shared space is a typical task where the classification can be used very effectively. The design and implementation of the text mining services (including both classification and clustering) for the KP-Lab system is described in more details in [3].

However, practical implementation of the classification services in the KP-Lab system led to the necessity of enhancements of the classification procedure itself. Originally, the classification was designed as a semi-automatic procedure, where the users (learners, students) were responsible for selection of proper classification model, algorithms, text pre-processing methods, and optionally also to restrict the training set. Assuming that learners are not experts in the field of text mining, it was hard for them to select the optimal settings and the requirement was to try to investigate the classification settings automatically, from global characteristics of the input data set. It resulted in a design of the meta-learning method for selection of classification algorithms, which we describe in greater details within the next sections of this paper.

2 Classification, basic principles

The classification belongs to one of the basic approaches in predictive data mining. In the case of text classification, it is an approach for specific knowledge extraction from textual documents. The process of classification consists of two phases [4]:

1. Construction of the classifier;
2. Usage of the classifier.

Basic functional blocks and components used in these two phases are depicted on Figure 1.

In the *first phase*, a given set of training examples (i.e. a set of already categorized text documents) is processed to create the classifier as a model of the data behavior. In the *pre-processing* step, the terms are extracted from the text of documents, and the whole input set is transformed into a vector representation [1]. A term-document matrix of m x n dimension is produced, where m is the number of indexing terms and n is the number of documents in the training set. The number of indexing terms and corresponding dimension m of the matrix can be reduced by various pre-processing and text analysis methods as e.g. tokenization, stop-words elimination, stemming and lemmatization, term clustering (LSI), etc. [1], [5].

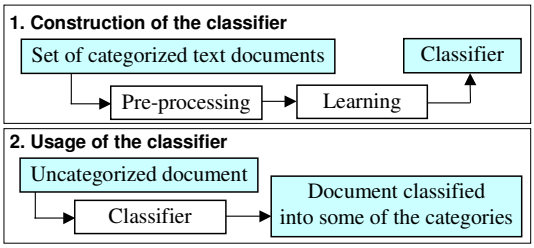

Figure 1. Two phases of the classification process

In the step of *learning*, various learning algorithms based on the statistical and heuristic techniques are used for processing the vector representation of the training set. Selection of proper algorithm and settings its optimal parameters is usually preformed manually and requires expert knowledge as well as an experience in the field of text mining. This is especially the point where the meta-learning approach (described in the next section) can help and select the most appropriate classification algorithm with respect to the characteristics of the training data set. In this paper, we will focus on the following algorithms:

- Linear classifiers: *Perceptron*, *Support Vector Machine* (*SVM*),
- Methods based on a recursive division of the space of documents into a set of disjunctive areas: *Decision trees*, *Decision rules*,
- Methods based on the instances: *k Nearest Neighbors* (*kNN*).

Besides the methods mentioned above, there exist also multiplicative algorithms as exponential gradient descent and Winnow algorithm, regression models as logic regression and linear discriminative analysis, probability models (Bayesian networks), neural networks, and methods based on instances or prototypes (Rocchio) [1].

For the construction of the classifier, it is assumed that every training example, i.e. an a-priori categorized document, belongs to one or more of the pre-defined categories. So all documents assigned to category i ($i \in \langle 1,N \rangle$, where N stands for the number of the categories) create a subset $D_i \subset D$, where D is the whole

training set. Let the $C = \{c_1, c_2, ..., c_N\}$ be a set of all available categories. If a document is categorized into more than one category, then it will belong to all of the corresponding subsets D_i. These newly created subsets D_i can be used for learning of so-called *binary classifiers*, i.e. the classifiers that are able to distinguish the documents of one category from the documents belonging to all of the rest of the categories.

For the creation of a single binary classifier for corresponding subset D_i, the task is to find the approximation of an unknown function $\Phi_i : D \times c_i \rightarrow \{true, false\}$. For d_{ij}, the Φ_i has the value *true* if the document d_{ij} belongs to the category c_i; otherwise, the value of Φ_i is *false*. The function $\hat{\Phi}_i : D \times c_i \rightarrow \{true, false\}$, which is the approximation of Φ_i, is specified by means of the selected learning algorithm and is called as *binary classifier*. This way, each category can have its own binary classifier; the union of these classifiers for all the categories forms the resulting classifier – so-called *classification model*, which implicitly describes the set of pre-defined categories.

The resulting classification model is used in the *second phase* for a prediction of the target attributes (i.e. the categories), which are identified for the "new" (i.e. unknown, a-priori uncategorized) documents from the input set. The input document is processed by all the binary classifiers $\hat{\Phi}_i$ from the classification model. If the value of $\hat{\Phi}_a$ is *true*, then the document is assigned as belonging into the category c_a. If there is no category for which the $\hat{\Phi}_i$ returns the value *true*, then the document is assigned as unclassified. Finally, a set of categories, predicted for the input document, is generated as a result of the classification procedure.

Quality of the classification can be evaluated using the *testing set* of documents. Similarly as the training set, the testing set contains the documents already (a-priori) categorized into the pre-defined categories. The testing documents are classified regularly, using the produced classification model (Figure 1). The results are then compared with the a-priori categorization for each testing document. This comparison is performed by a set of statistical measures; the most frequently used indicators are the *precision*, *recall*, and *combined effectiveness measure F1*. These measures can be combined into one global measure for the space of all categories by *micro averaging* and *macro averaging* methods [4], [5]. We will use these measures to evaluate the results of our experiments (section 4).

3 Meta-learning

Implementation of the classification procedure in practice requires the selection of proper algorithm in the phase of classifier creation, namely in the learning step. In the KP-Lab system, as well as in similar user-oriented systems, the selection of classification algorithms can be done in advance, manually by system developers. But this may not work well for all categories and all new documents to be classified. Alternatively, the meta-learning approach can be used to automate the selection of the algorithms separately for each of binary classifiers, according to the specific characteristics of the training set of documents, thus resulting into more adaptive and flexible classification procedure. This approach does not require any additional effort from user side for controlling the classification process and provides higher quality of the classification results.

The meta-learning approach is based on a design of an adaptive system, which can increase its effectiveness based on the feedback from previous "experiences", i.e. on the evaluation of the examples processed in past [6]. Selection of the best learning strategy, most suitable for particular problem, is a generalization based on accumulating experience on the performance of multiple applications, strategies, or algorithms [7]. In the domain of text classification, the meta-learning approach is able to select the most appropriate and the most effective classification algorithm according to the characteristics of the training set (as e.g. term or category distribution, average length of documents, etc.). To achieve this selection, there is a need to create the decision mechanism (meta-model) in the first step and then to use it in the second step for creation of new classifiers (cf. first phase of general text classification process, presented on Figure 1).

The process of the meta-learning approach applied in text classification for construction of classifiers consists again of the two phases, as depicted on Figure 2:

1. Construction of the meta-model;
2. Usage of the meta-model for selection of algorithms and for creation of classifiers.

First phase of the meta-model construction can be further divided into the two steps:

- Specification of feature characteristics for training documents;
- Learning of the meta-model (meta-classifier).

The feature characteristics can be obtained from the training set for each of categories and can be expressed as a vector $F_i = (f_{i1}, f_{i2}, \ldots, f_{il})$. These vectors can then be used in the step of meta-model learning for selecting the most appropriate algorithm for particular categories.

The meta-model learning is usually based on prediction of an optimization parameter, given by comparison and evaluation of the feature characteristics F_i with the values of effectiveness, i.e. with the classification errors obtained from applying pre-defined classification algorithms on the training and testing set. The meta-model can then be constructed from these values using a regression analysis or a meta-classification procedure.

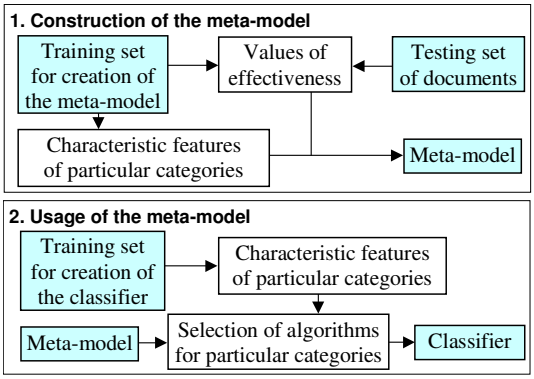

Figure 2. Meta-learning approach, two phases

Second phase of the meta-model usage is rather simple, where the feature characteristics are obtained from unknown (uncategorized) input documents and are processed in the same way as in the phase of meta-model construction. The meta-model is then able, according to the feature characteristics of the new documents, to select and propose the most suitable classification algorithm for creation of the resulting classifier.

In the KP-Lab, we adopted the MUDOF (Meta-learning Using Document Feature Characteristics) algorithm [8], based on the multiple regression analysis of feature characteristics obtained from the training set of text documents. We have implemented the MUDOF algorithm as an extension of the JBowl library [9], which was selected as the main implementation platform for the solution of the classification services for the KP-Lab system [3]. In addition, after a set of initial experiments, we have enhanced the MUDOF algorithm itself in several ways, producing a modified version where the meta-model learning step is based on the meta-classification procedure, using the kNN classification algorithm. The modified algorithm, referenced as MUDOF_K (i.e. MUDOF with kNN meta-learning) was also implemented into the JBowl. A set of experiments were performed to compare the effectiveness and results of the original MUDOF_R (i.e. MUDOF based on regression analysis) with MUDOF_K and with the traditional classification using the five pre-defined algorithms (see section 2). Some of these experiments are described and discussed in the section 4 below.

The MUDOF proposes a set of nine ($l = 9$) feature characteristics [8], from which we have selected the following:
- *AvgTopInfoGain*, average information gain of the best t terms of a given category. The information gain of individual terms is computed for each of categories, average is then counted from t terms with the highest information gain.
- *PosTr*, number of positive examples in the training set for given category.
- *AvgTermVal*, average weight of document's terms for given category. The average weight of terms for a single document is computed at first; then the weight is computed for all the positive examples of a given category.
- *NumInfoGainThres*, number of terms for which the information gain value exceeds a globally specified threshold.
- *AvgDocLen*, average length of a document for given category. The document's length is computed as a number of all the indexed terms in a document. The average is obtained by computing the length for all the positive examples for given category.

The selection of these feature characteristics was accomplished according to the experimental results. The above mentioned five characteristics were selected as the most representative, with the most significant influence on the selection of algorithms. In addition, the characteristics *PosTr* and *NumInfoGainThres* were modified into the form of a ratio or an average value, since it provides a more adequate description of global characteristics over particular categories. The modifications were as follows:
- *PosTr*, ratio of positive and negative examples in the training set for given category.
- *NumInfoGainThres*, ratio of the number of terms with the information gain over the threshold to the number of all the terms.

The MUDOF algorithm requires a division of the training set into two sub-sets [8]:
- training set for meta-model (*TM*),
- training set for classification model (*TC*).

The characteristic features for the categories can then be obtained from *TM* and *TC* as two separate data sets. The *TM* data are used for an estimation of the regression model parameters, and the data from *TC* are used for prediction of the classification error for particular algorithms used within the binary classifiers for the given categories. The algorithm with the lowest estimation of the classification error on a category is then returned as the best (optimal) and will be used for the construction of binary classifier of this category.

The MUDOF algorithm uses a prediction of the classification error for a given category, based on the characteristic features of the training documents belonging to this category. In the MUDOF_R algorithm, the regression model describing the relations between characteristic features and classification algorithms is created in the phase of learning. A goal is to obtain the $(\hat{\beta}_{jk})$ parameters for each of the algorithms. Our implementation of the MUDOF_R algorithm can be described in the following steps:

A. Meta-model construction:
Input: *TM*, *TC*, set of available classification algorithms *A*, set of categories *C*.
1. While (there is an algorithm in *A*)
2. Take an algorithm ALG_j from *A*
3. For each (category c_i from *C*)
4. Apply ALG_j on *TM* for c_i and obtain the binary classifier CF_{ij}
5. Apply CF_{ij} on *TC* for c_i and obtain the classification error e_{ij}
6. Make the logarithmic transformation of e_{ij}: $y_{ij} = \ln \frac{e_{ij}}{1 - e_{ij}}$, where y_{ij} is the response variable
7. End For
8. Estimate the $(\hat{\beta}_{jk})$ parameters of regression model for ALG_j using response variable y_{ij} and vector of feature characteristics F_{ik} (on the *TM* training set)
9. End While

B. Usage of the meta-model:
10. For each (category c_i from *C*)
11. While (there is an algorithm in *A*)
12. Take an algorithm ALG_j from *A*
13. Estimate the classification error e_{ij} using the $(\hat{\beta}_{jk})$ and corresponding F_{ik} (on the *TC* set)
14. If the e_{ij} is minimal, then the ALG_j is the best for category c_i
15. End While
16. End For

The designed modification of the MUDOF_K differs from the MUDOF_R by the meta-model learning method. Instead of linear regression, the MUDOF_K uses the classification approach, based on the kNN method. Main advantage of the proposed modification is the possibility of incremental learning of the meta-model. This feature is especially helpful in the systems like KP-Lab, where the input data set is updated rather frequently and the changes should be reflected in the meta-model.

4 Experiments

The original meta-learning algorithm MUDOF_R, based on the regression model, as well as the MUDOF_K modification, based on the kNN classification method, were both implemented as an extension of the JBowl library. The implementations were then tested in a set of experiments to prove the concept of automatic creation of classifiers by the meta-learning approach and to evaluate the quality of the resulting classification procedure.

4.1 Preparation of the testing data

The experiments were accomplished on the *Reuters-21578* [10] and *20 Newsgroups* [11] document sets. The Reuters-21578 contains 10.788 documents distributed into 90 categories. For the experiments, the document set was divided into the following subsets:
- training set (*TR*): 7.769 documents,
- testing set (*TE*): 3.019 documents.

For the meta-learning, the *TR* was further divided into the training sets for meta-model and for classifier:
- *TM*: 3.815 documents,
- *TC*: 3.961 documents.

The Reuters-21578 set is not very well balanced; it has a high variability of the documents distribution towards the categories. It contains categories with about 1.500 positive examples, as well as about 30 categories with less than 10 documents.

The 20 Newsgroups contains 19.997 documents distributed into 20 categories. The 20 Newsgroups set is well balanced and has low data variability, since almost equal number (about 1.000) of documents belongs into each of the categories. For the experiments, we have divided the 20 Newsgroups set into the following subsets:
- training set (*TR*): 10.025 documents,
- testing set (*TE*): 9.972 documents.

4.2 Experiment 1, single data set

First experiment was focused on testing of the meta-learning approach on a single data set. The goal was to prove the hyphothesis that the meta-learning provides a better effectiveness and quality of the resulting classifier in comparison with the several pre-defined classification algorithms. This experiment was performed on the Reuters-21578 document set.

The *effectiveness* of the classification was evaluated by the *F1* quality measure mentioned in the section 2 above. The integrated measure *Macro F1*, which combines precision and recall over whole testing set, was used as the main quality measure for the experimental results. The MUDOF_K and MUDOF_R algorithms were compared with basic classification algorithms as *Decision Trees*, *Decision Rules*, *SVM*, *Perceptron*, and *kNN*. Resulting values of the quality measures are listed in Table 1, graphical comparison of the *Macro F1* measure is depicted on Figure 3. Macro F1 has been chosen because it is the most decriptive effectiveness measure for unballanced document sets (like Reuters-21578).

Table 1. Single data set, quality measures

Statistics	MUDOF_K	MUDOF_R	Dec. Trees
Micro Precision	0,808	0,869	0,790
Micro Recall	**0,860**	0,820	0,793
Micro F1	0,833	0,844	0,792
Macro Precision	0,567	0,556	0,521
Macro Recall	**0,520**	0,502	0,503
Macro F1	**0,543**	0,527	0,511

Statistics	Dec. Rules	SVM	Perc.	kNN
Micro Precision	0,792	**0,932**	0,885	0,852
Micro Recall	0,801	0,785	0,794	0,792
Micro F1	0,796	**0,852**	0,837	0,821
Macro Precision	0,499	**0,580**	0,556	0,496
Macro Recall	0,492	0,369	0,356	0,384
Macro F1	0,495	0,451	0,434	0,433

The results demonstrate that the MUDOF algorithms, using the meta-learning approach, are able to provide higher values of the resulting effectiveness, expressed by the Macro F1 measure. For the macro measure, the MUDOF has similar results as Decision Trees and Rules. However, the MUDOF has better results for the Micro measure. The *SVM*, *Perceptron*, and *kNN* have similar and slightly better (in case of *SVM*) results as MUDOF for the Micro measures, but the MUDOF is better in the results for Macro measures. Percentage increase of the MUDOF algorithms in comparison with the globally best basic algorithm, i.e. Decision Trees, was 4,1% for 3,1% for the Macro F1 measure.

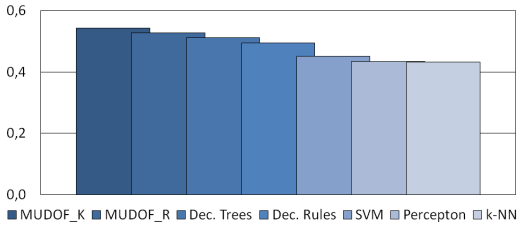

Figure 3. Single data set, comparison of the Macro F1 measure for algorithms

As a part of the first experiment, we have also investigated how the selection of a classification algorithm for particular categories influences the effectiveness of the resulting classifier. The global effectiveness depends on the ability to select the best algorithms for particular category in the meta-learning phase. This selection is highly dependent on the method how the algorithms are selected as well as on the adequate description of categories by their characteristics.

The threshold cases IDEAL (for each category the best classifier would be chosen) and ¬IDEAL (for each category the worst classifier would be chosen) were introduced to border the space of possible values of the effectiveness, which can be obtained by combining binary algorithms for particular categories. The selection of algorithms provided by the meta-learning can then be compared in relation to these threshold values. In addition, the threshold case AVERAGE was also included in the comparison. This value, expressing the average effectiveness of the five basic algorithms, can be used as minimal constraint for the effectiveness reached by meta-learning approach. In other words, the meta-learning approach should provide the values of effectiveness which are somewhere between the AVERAGE and IDEAL

thresholds. The values of the basic quality measures for the threshold cases evaluated for the Reuters-21578 data set are listed in Table 2.

The results demonstrate that both MUDOF implementations achieved high increase of effectiveness in comparison with the ¬IDEAL threshold case: 22% for MUDOF_R, and 23,6% for MUDOF_K. In comparison with the AVERAGE, the increase was 6,2% and 7,7%, resp. However, the decrease of the effectiveness was achieved for the IDEAL threshold: 11,3% for MUDOF_R, and 9,8% for MUDOF_K. Despite the overall increase in comparison with basic algorithms, there is still some space available here for improving the effectiveness of the meta-learning.

Table 2. Single data set, quality measures for thresholds

Statistics	IDEAL	AVERAGE	¬ IDEAL
Micro Precision	0,928	0,851	0,755
Micro Recall	0,844	0,792	0,687
Micro F1	0,884	0,819	0,720
Macro Precision	0,716	0,532	0,360
Macro Recall	0,580	0,422	0,267
Macro F1	0,641	0,466	0,307

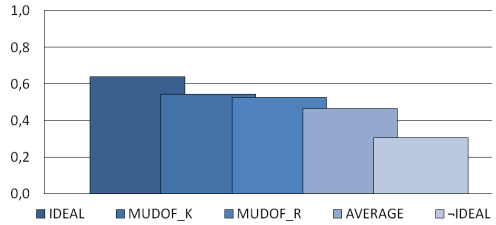

Figure 4. Single data set, comparison of the Macro F1 measure for MUDOF algorithms and thresholds

4.3 Experiment 2, two data sets

In the second experiment, the goal was to test the usability of the meta-leraning approach on two different sets of documents, achieving a situation where meta-model has been trained on different dataset than it has been tested later on. Meta-model learning phase was performed on the Reuters-21578 document set (we have used the same meta-model as in previous example), and the resulting classifier was constructed on the 20 Newsgoups data. The process of obtaining results and their evaluation is the same as in the first experiment. Values of the effectiveness measures for the experiment with two data sets are presented in Table 3, graphical comparison of the Macro F1 measure is depicted on Figure 5.

It follows from the achieved results that the *SVM* algorithm is the best for the ballanced data of the 20 Newsgroups. All the algorithms except *Perceptron* have almost equal results, no significant improvement was achieved by applying the meta-learning (Figure 5). In the case of ballanced data, a single algorithm can be selected as the best – in our case it is the *SVM*. The meta-learning approach is able to assure that the resulting effectiveness will be "close" to the best, and avoid a selection of the algorithms with bad effectiveness (Perceptron, in our case).

Table 3. Two data sets, quality measures

Statistics	MUDOF_K	MUDOF_R	Dec. Trees
Micro Precision	0,824	0,899	0,892
Micro Recall	**0,894**	0,871	0,873
Micro F1	0,857	0,884	0,883
Macro Precision	0,830	0,896	0,891
Macro Recall	**0,895**	0,869	0,875
Macro F1	0,861	0,882	0,883

Statistics	Dec. Rules	SVM	Perc.	kNN
Micro Precision	0,892	**0,961**	0,286	0,838
Micro Recall	0,873	0,843	0,383	0,847
Micro F1	0,883	**0,898**	0,328	0,843
Macro Precision	0,891	**0,958**	0,782	0,845
Macro Recall	0,875	0,844	0,384	0,848
Macro F1	0,883	**0,897**	0,515	0,846

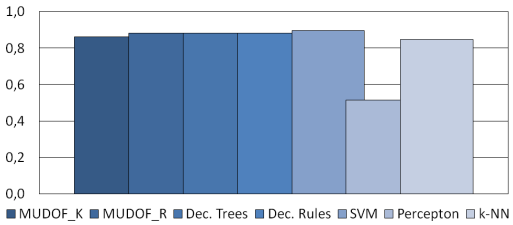

Figure 5. Two data sets, comparison of the Macro F1 measure for algorithms

The values of effectiveness measures for the threshold cases are presented in Table 4, graphical comparison of the thresholds and MUDOF algorithms by means of the Macro F1 measure is depicted on Figure 6.

Table 4. Two data sets, quality measures for thresholds

Statistics	IDEAL	AVERAGE	¬ IDEAL
Micro Precision	0,959	0,774	0,689
Micro Recall	0,900	0,764	0,021
Micro F1	0,928	0,767	0,041
Macro Precision	0,960	0,873	0,690
Macro Recall	0,906	0,765	0,359
Macro F1	0,932	0,805	0,472

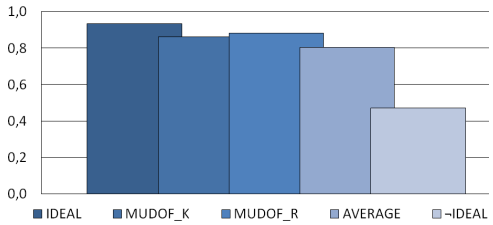

Figure 6. Two data sets, comparison of the Macro F1 measure for MUDOF algorithms and thresholds

The results show that the effectiveness of both MUDOF implementations is above the AVERAGE threshold for the F1 measure for both micro and macro averaging. The increase of the Macro F1 measure is 7,7% for MUDOF_R and 5,6% for MUDOF_K. It proves the legitimity of applying the meta-learning also for a set of two (or more) data sets, and even for differently ballanced data sets.

5 Conclusions

The presented meta-learning approach towards the text classification seems to be a suitable method for support of automatic classification in user-oriented systems. The original MUDOF meta-learning algorithm, based on the linear regression, was modified and adapted using the *kNN* classification method for meta-model creation. Both algorithms were tested on the Reuters-21578 and 20 Newsgroups document sets and the results indicate that the meta-learning increases effectiveness and quality of the results. Comparison with the ideal threshold values shows that there is still some space for further improvements, especially in the case of balanced training sets. However, the proposed meta-learning approach can be considered as a technology, which enables automatic and adaptive text classification, increases quality of the classification results, and can be effectively used in the user-oriented systems in practice.

6 Acknowledgments

The work presented in this paper was supported: by European Commission DG INFSO under the IST program, contract No. 27490; by the Slovak Research and Development Agency under the contracts No. APVV-0391-06 and RPEU-0011-06; and by the Slovak Grant Agency of Ministry of Education and Academy of Science of the Slovak Republic under the contract No. 1/4074/07.

The KP-Lab Integrated Project is sponsored under the 6th EU Framework Programme for Research and Development. The authors are solely responsible for the content of this article. It does not represent the opinion of the KP-Lab consortium or the European Community, and the European Community is not responsible for any use that might be made of data appearing therein.

References

[1] Bednár P.: **Authomatic classification of texts based on the content** (in Slovak), Concept of PhD. thesis, TU Košice, Slovakia, 2004.

[2] Bednár P., Butka P., Paralič J.: **Java Library for Support of Text Mining and Retrieval**, Proc. of the conference Znalosti 2005, Stará Lesná, Slovakia, 2005, pp. 162-169.

[3] Furdík K., Paralič J., Smrž P.: **Classification and automatic concept map creation in eLearning environment**, Proc. of the conference Znalosti 2008, Bratislava, Slovakia, 2008, pp. 78-89.

[4] Paralič, J.: **Knowledge Discovery in databases and texts**, Habilitation thesis, Technical University of Kosice, Slovakia, 2003.

[5] Sebastiani F.: **Machine Learning in Automated Text Categorization**, ACM Computing Surveys, Vol. 34, Iss. 1, New York, USA, 2002, pp. 1-47.

[6] Vilalta R., Drissi Y.: **A Perspective View And Survey Of Meta-learning**, AI Review, Vol. 14, No. 2, Springer Netherlands, 2002, pp. 77-95.

[7] Vilalta R., Giraud-Carrier Ch., Brazdil P.: **Meta-Learning: Concepts and Techniques**, The Data Mining and Knowledge Discovery Handbook, Springer US, 2005, pp. 731-748.

[8] Wai L., Kwok-Yin L.: **A meta-learning approach for text categorization**, Proc. of the 24th ACM SIGIR conference, New Orleans, USA, 2001, pp. 303-309.

[9] Bednár P.: **JBowl, Java library**, available at `http://sourceforge.net/projects/jbowl/`, Accessed: 12th May 2008.

[10] Lewis D.: **Test data Collection Reuters-21578**, available at `http://www.daviddlewis.com/resources/testcollections/reuters21578/`, Accessed: 12th May 2008.

[11] The UCI KDD Archive: **20 Newsgroups**, University of California, Irvine, 1999, available at `http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html`, Accessed: 12th May 2008.