DOI:10.2298/CSIS100420034V

Ontology-based multi-label classification of economic articles

Sergeja Vogrinčič[1] and Zoran Bosnić[2]

[1] Jožef Stefan International Postgraduate School,
Jamova 39, 1000 Ljubljana, Slovenia
sergeja.sabo@mps.si
[2] University of Ljubljana, Faculty of Computer and Information Science
Tržaška cesta 25, 1000 Ljubljana, Slovenia
zoran.bosnic@fri.uni-lj.si

Abstract. The paper presents an approach to the task of automatic document categorization in the field of economics. Since the documents can be annotated with multiple keywords (labels), we approach this task by applying and evaluating multi-label classification methods of supervised machine learning. We describe forming a test corpus of 1015 economic documents that we automatically classify using a tool which integrates ontology construction with text mining methods. In our experimental work, we evaluate three groups of multi-label classification approaches: transformation to single-class problems, specialized multi-label models, and hierarchical/ranking models. The classification accuracies of all tested classification models indicate that there is a potential for using all of the evaluated methods to solve this task. The results show the benefits of using complex groups of approaches which benefit from exploiting dependence between the labels. A good alternative to these approaches is also single-class naive Bayes classifiers coupled with the binary relevance transformation approach.

Keywords: ontology, multi-label classification, machine learning, text categorization, economics, document classification.

1. Introduction

Classification of textual data has become increasingly important during the last decade, along with its many applications on the World Wide Web. People are using intelligent agents to find content of their interest as well as the articles pertaining to their research fields. Traditionally, the librarians, authors and field experts were in charge of categorizing documents with keywords, numerical classifications and other metadata which is used to summarize the documents' contents as well as to enable more efficient document retrieval through keyword search.

To ease the task of the document classification and retrieval in various domains, the usage of automatic approaches is welcome. By replacing the tedious work of manually categorizing documents, automatic document

classification can utilize computer resources to perform the task more efficiently. In addition to more efficient execution of the task, the automatic approach can be in practice used to categorize large sets of documents which have not been annotated in the past, hence enabling their automatic retrieval which was not possible so far.

In this paper, we address a problem of automatic document classification, focusing on the economic domain in particular. Given a corpus of economic scientific papers' abstracts, we aim at finding the answers to the following questions:

1. How to define the possible classes for the documents in the corpus, having no prior knowledge about their contents? We approach this challenge by utilizing a tool for semi-automatic ontology generation, using which we detect and define the most notable concepts in the domain, represented by the corpus. In the following, we use these concepts as possible document classes.

2. What are the appropriate classification algorithms, suitable for this task? In real-life scenarios, documents are frequently annotated with more than a single category. With our aim of classifying such document, multi-label classification approaches, which can predict many classes simultaneously, are required. In addition to relevance of multiple document categories, a hierarchical dependence of the categories may exist in a domain. Based on these facts, in our experimental work we comparatively evaluate: (i) adapted single-class classifiers, coupled with a data transformation approach which enables predicting multiple classes simultaneously [1], (ii) specialized models for multi-label prediction and (iii) a specialized hierarchical multi-label classification model and a ranking model.

We expect that the performance of the machine learning approach should justify the usage of the automatic document classification approach in this domain. Additionally, we also expect the specialized models to perform better than the adapted single-class classifiers.

The paper is organized as follows. In Section 2, we summarize the related work on ontologies (tools, economic ontologies), multi-label classification and text classification approaches. In Section 3, we describe our construction of the ontology of economics using a semi-automatic ontology tool and other text mining software. In Section 4, we empirically evaluate the performance of various types of the multi-label prediction approaches and present their results. We conclude the paper in Section 5, where we present the ideas for further work as well.

2. Related work

As noted in the Introduction, in our work we combine the field of semi-automatic ontology generation with the field of supervised multi-label classification in machine learning. In the following we present the related work in both fields.

2.1. Ontology tools and economic ontologies

Ontology is a formal representation of the knowledge by a set of concepts within a domain and the relationships between those concepts. It is used to reason about the properties of that domain, and may be used to describe the domain. Ontologies have become increasingly important research topics in many areas, dealing with a particular domain structure, categories, entities and their inter-relations. Ontology consists of *concepts*, their *hierarchical relations*, their *additional arbitrary relations*, and *axioms*. Additionally, it may also contain other *constraints* and *functions* [2]. The connection between nodes within ontology can be represented by a graph, namely ontology chart.

Ontology creation and management tools. Ontologies can be constructed, structured and managed either manually or using a certain degree of automatism [3]. Among the latter techniques, we can distinguish between semi-automatic and fully automatic methods for construction, maintenance and evolution strategy of the ontology, depending on the interaction they require from the user. The most of the methods are semi-automatic, being implemented as specialized ontology tools.

Currently, there are many such tools for managing an ontology, such as *Protégé* [4], *OntoEdit* [5], *Ontolingua ontology development environment* [6], *Chimaera* [7] and *OntoGen* [8, 9]. OntoGen is a semi-automatic and data-driven ontology editor focusing on editing of topic ontologies. The system combines the text mining techniques with an efficient user interface to bridge the gap between the complex ontology editing tools and the domain experts who are constructing the ontology. Due to efficient integration of text mining techniques with the ontology authoring, we chose OntoGen as a tool for creation of the economic ontology, which we describe in Section 3.

Ontological representations of economics. Although some works on economic ontologies exist, none of them systematically covers the whole field of economics, but focuses mainly on the ontological representation of some economic sub-area pertaining to the article topic. Zuniga [10] deals with the ontology of economic objects. The author describes economic categories and laws that provide the conditions for settling objectively whether individuals' views about an instance of any category indeed correspond to that category. Blomqvist [3] deals with a fully automatic construction of enterprise ontologies using design patterns and deals with the creating an enterprise ontology for an automotive supplier. Siricharoen [11] uses the economic domain to illustrate to the economists how ontologies work, as well as to guide the computer software developers to understand the basic concepts of economy. There are also additional works in this field, approaching the economics through the views of a philosophical science, social science and politics [12, 13].

As shown in the following section, in our work we construct a new representation of the economic domain, based on the corpus of economic documents. Since the ontology is built using the underlying economic

documents, it will include the concepts from those documents which are suitable for their further categorization.

2.2. Multi-label classification in machine learning

A large body of research in supervised learning deals with the analysis of a single label data, where training examples are associated with a single label *l* from a set of disjoint labels *L* (i.e. a single-label classification). However, nowadays training examples in most of application domains are associated with a set of labels $Y \subseteq L$, being therefore multi-labeled. Learning from such examples and predicting their labels therefore calls for multi-label prediction approaches [1,14].

The two major tasks in supervised learning from multi-label data are: multi-label classification (MLC) and label ranking (LR). MLC [14] is concerned with learning a model that outputs a bipartition of set of labels into relevant and irrelevant with respect to a query instance. LR [15], on the other hand, is concerned with learning a model that outputs an ordering of the class labels, according to their relevance to a query instance. In certain classification problems, the labels belong to a hierarchical structure, then we call the task hierarchical classification. If each example is labeled with more than one node of the hierarchical structure, then the task is called the hierarchical multi-label classification [14].

Many real-world problems in the areas of text mining, semantic annotation of images and videos [16, 17], web page categorization [18], music categorization [19], bioinformatics (gene functional analysis, functional genomics) [18, 20, 21, 22], and many others, are multi-label problems. This has attracted attention from many researchers who were motivated to find a number of new applications to solve these problems.

2.3. Text classification

Text classification (also known as text categorization), where each document may belong to several topics (or labels, keywords, categories, classes), is the task of building learning systems capable of classifying text documents into one or more predefined categories or subject codes. Textual data, such as documents and web pages, are frequently annotated with more than a single label. The categorization of textual data is perhaps the most dominant multi-label application. One of the well-known approaches to solve the problem of text classification is BoosTexter proposed by Schapire and Singer [23], which is extended from the ensemble learning method AdaBoost. A Bayesian approach to multi-label document classification proposed by McCallum [24] combines a mixture probabilistic model and the EM algorithm. Ueda and Saito [25] proposed two parametric mixture models (PMMs) for multi-label text classification, where basic assumption under PMMs is that multi-labeled text

has a mixture of characteristic words, appearing in single-labeled text that belong to each category of the multi-categories.

The classification of the textual data from the domain of economics, which is the focus of our paper, is certainly a case of a multi-label classification which has not received much systematic attention so far.

3. Ontology-based assignment of document classes

To construct the ontology of the economic domain, we created a corpus of economic documents. With the aim to include such texts which contain representative phrases and words for this domain, we decided the corpus to include the abstracts of the published papers in the distinguished economic journals (such as *Quarterly Journal of Economics, Journal of Economic Literature, Journal of Economic Perspective, Econometrica etc.*). We used the JSTOR (http://www.jstor.org/) online service, which includes contents of over thousand academic journals and other scholarly content, to collect 1015 such abstracts. Although short in size, we expect the abstract to hold enough of condensed information required to categorize the paper into the appropriate categories.

3.1. Construction of the economic ontology

As mentioned in Section 2.1, we used OntoGen to partially make the construction of the ontology automatic. OntoGen [8,9,26] is a semi-automatic and data-driven ontology editor. The system combines text-mining techniques with a user interface to ease and integrate automatic analysis of texts and ontology construction based on the found important keywords and concepts. The authors [26] define the system to be *semi-automatic* (it suggests concepts, relations between them, visualizes instances within a concept and provides a good overview of the ontology through concept browsing, while the user is always in full control of the system's actions and can accept or reject the system's suggestions) and *data-driven* (most of the aid provided by the system is based on the underlying data provided by the user typically at the beginning of the ontology construction; the data provided as a document corpus serves for an automatic extraction of instances for the concept and relation learning).

After creating a document corpus for OntoGen, we utilized the following functionalities of OntoGen to construct the economic ontology using the:

1. k-means clustering (unsupervised learning): we automatically generated a list of possible sub-concepts for concepts of interest by using k-means clustering. We performed the clustering many times for different possible numbers of clusters (sub-concepts) and selected the most

reasonable grouping[1]. The sub-concept generation was repeated recursively to create the sub-sub-concepts etc., until the desired granularity of the concepts was reached.

2. querying for a particular concept and active learning: by providing a set of keywords describing the sought concept, the system followed by asking a series of yes/no questions, if particular documents belonged to the concept in question. The system automatically identified the documents that corresponded to the topic, and the selection was further refined by the user-computer interaction through an active learning loop, using a machine learning technique for a semi-automatic acquisition of the user knowledge. The questions were chosen from the instances on the border between being relevant to the query or not and were therefore most informative to the system. The system than refined the suggested concept after each our reply, and we decided when to stop the process based on how satisfied we were with the suggestions. After the concept was constructed, it was added to the ontology as a sub-concept of the selected concept.

3. visualization-based assignment to concepts: OntoGen has a functionality to visualize a document corpus. In the document space the similar documents are visualized as the neighboring points, while the less similar are visualized more apart from each other. By interactively selecting very dense subgroups of the documents in the visualized space and analyzing their keywords, we were able to assign those document groups to a concept in the ontology.

4. manual assignment of documents to concepts: in the last step of ontology construction we manually analyzed which document corresponded to each concept in the ontology, and fine-tuned the document categorizations by: categorizing documents to some other concepts, categorizing documents to additional (more than one) concepts or categorizing documents which were not categorized so far.

After performing the semi-automatic construction steps (1 and 2 above), we therefore additionally improved the ontology by performing manual steps (3 and 4 above). As a result, the hierarchy of concepts (ontology) shown in Figures 1 and 2 was obtained. As shown in the both figures, we developed the hierarchy to the maximum depth of three levels. The lowest level contains 16 different concepts, corresponding to 16 different classes (not considering their parent concepts). Note that a particular document could had been assigned to different categories (using concept querying, visualization-based selection and manual assignment), thus the same document belongs to more than one class which is a scenario for the multi-label classification.

[1] according to the first author's judgment, who is a bachelor of economics

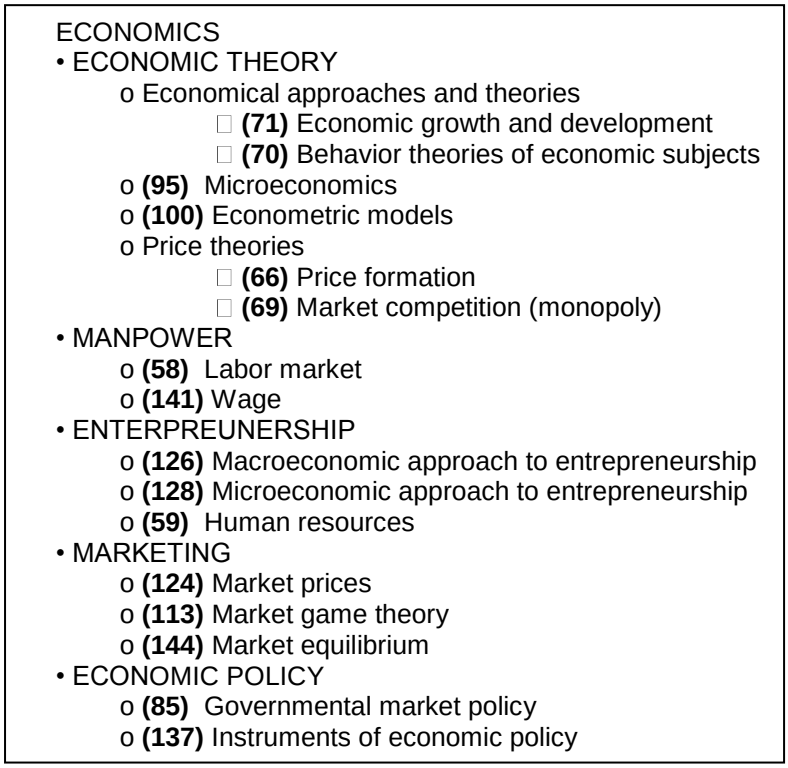

Fig. 1. The hiearchy of the concepts in the ontology of the economic domain. The numbers of the bottom-level concepts denote the number of documents (of total 1015), associated with that category (since each document can be associated to many concepts, the sum of these numbers is higher than 1015).

3.2. Dataset for multi-label learning

After creating the ontology of economics, OntoGen allowed us to export the constructed hierarchy, along with the information which document belongs to which concepts as a set of Prolog[2] clauses. In parallel, we used the R Project for Statistical Computing software [27] to transform the document corpus into the example-attribute relational dataset, suitable for machine learning.

[2] Prolog is a general purpose declarative logic programming language associated with artificial intelligence and computational linguistics. Prolog has its roots in formal logic; the program logic is expressed in terms of relations, represented as facts and rules.

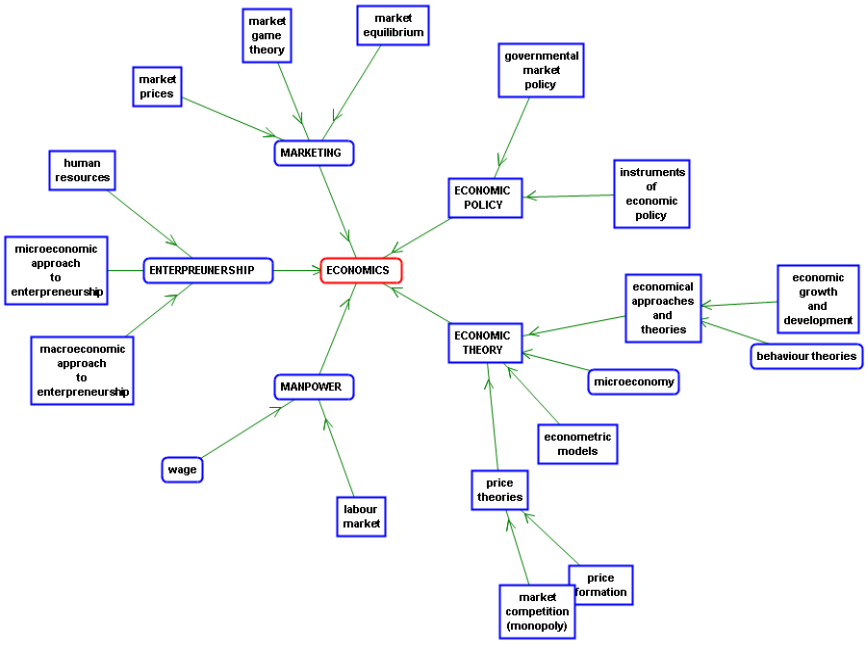

Fig. 2. Visualization of the construncted ontology

For our purpose, we used the R text mining library *tm* [28]. For each document we performed the typical preprocessing operations: converting to lower case, removal of punctuation and numbers, removal of stopwords (common words such as *and*, *the*, *of*, etc.), stemming (removal of different suffixes to keep only the root of the word).

For each document we computed the attributes, representing each appearing word by using the popular weighting schema, the normalized term (word) frequency TFIDF [29] which is based on the logic that the word is more important if it appears several times in a target document, and if it appears in less documents in the corpus:

$$tfidf\,(w) = tf\,(w) \cdot \log\left(\frac{N}{df\,(w)}\right) \tag{1}$$

where *tf(w)* denotes the term frequency (number of word occurrences in a document), *df(w)* denotes the document frequency (number of documents containing the word), and *N* denotes the number of all documents.

The computed attributes were combined with the corresponding document classes which were exported from OntoGen as described in the first paragraph. After performing the feature selection using the information gain measure, we selected the 100 best evaluated attributes out of 4991 total, yielding the final dataset ready for the multi-label classification.

In the following, we tested the performance of various multi-label classification models on the obtained dataset. Note that the ontology generation tool was only used to automatically label the documents in a corpus, and that the further testing scenario will independently assure unbiasedness of the testing procedure by cross-validating the dataset. In addition, the authors have taken care that the developed ontology covers all general fields of economics, independently of the underlying documents' contents.

4. Multi-label classification

In this Section we test the prediction performance of various multi-label classification approaches, including hierarchical multi-label classifier and classification ranking algorithm. We present and compare the empirical results of the evaluated methods, achieved using the tenfold cross-validation evaluation of the models. We implemented the experiment in the Weka [30] environment, using the additional library Mulan [1] which is an open source Java library for multi-label learning.

In the following, we will use $L = \{\lambda_j, j = 1 \dots q\}$ to denote a finite set of labels in a multi-label learning task, and $D = \{(\boldsymbol{x_i}, Y_i), i = 1 \dots m\}$ to denote a set of multi-label examples, where $\boldsymbol{x_i}$ denotes a feature vector and $Y_i \subseteq L$ the set of labels of the i-th example. To evaluate the performance of the tested methods, we observed measures of their classification accuracy (CA) and the average precision (AP). The CA measure is in the context of multi-label classification a very strict measure, as it requires the predicted set of labels to be an exact match to the true label set:

$$ClassificationAccuracy = \frac{1}{m}\sum_{i=1}^{m} \#(Z_i = Y_i) \tag{2}$$

where $\#(condition)$ returns the number of occurrences for which the condition holds, Y_i denotes a set of true labels and Z_i denotes a set of predicted labels for the i-th example.

The AP is a ranking measure which evaluates the average fraction of labels that were ranked above a particular label $\lambda \in Y_i$ which actually are in Y_i:

$$AveragePrecision = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{Y_i}\sum_{\lambda \in Y_i}\frac{|\{\lambda' \in Y_i : r_i(\lambda') \leq r_i(\lambda)\}|}{r_i(\lambda)} \tag{3}$$

where m denotes the number of examples, Y_i a set of labels of the i-th example and $r_i(\lambda)$ a ranking of the label λ which was predicted for the i-th example.

While the classification accuracy measures the number of exact matches between the sets of the true and the predicted labels, the average precision is a softer measure. High average precision can be interpreted as an indicator

that among the predicted labels, the true labels were given more priority (in terms of the predicted higher rank or class probability) than the irrelevant labels.

4.1. Transformation to a single-label classification problem

The group of methods which transform the learning set to traditional single-label classification task can be applied to any multi-label classification problem. They transform the learning task into one or more single-label classification tasks in combination with which one can use an arbitrary single-label classifier.

Among possible dataset transformation practices we used the following three approaches [1]:

- instance copy transformation (CO) which replaces every multi-label example (x_i, Y_i) with $|Y_i|$ examples (x_i, λ_i), for every $\lambda_i \in Y_i$. A single-label classifier that outputs class probability distributions can afterwards be used to learn the ranking and predict the relevant labels for a query instance. To output a bipartition of the relevant and irrelevant labels and thus solve a multi-label classification problem, a threshold needs to be applied to the predicted label probability scores. In our experimental work, the default selected probability threshold was 0.5. In cases where probabilities of all labels were less than 0.5, the most probable class was output as relevant, regardless of its probability score.
- label powerset (LP) which considers each unique set of labels that exists in a multi-label training set as one of the classes of a new single-label classification task. Given a new instance, the single-label classifier of LP outputs the most probable class, which is actually a set of labels.
- binary relevance (BR) which learns q binary classifiers, one for each different label in L. It transforms the original dataset into q datasets $D_{\lambda_j}, j = 1 \dots q$ that are intended to perform binary classification tasks for each label in $L = \{\lambda_j, j = 1 \dots q\}$. The transformed datasets contain all examples of the original dataset labeled positively if the label set of the original example contained λ_i and negatively otherwise. For the classification of a new instance BR outputs the union of the labels λ_i that are positively predicted by the q classifiers.

Figure 3 provides an example which illustrates the effect of the above transformations (CO, LP and BR) on an example dataset. Note that the LP transformation results in a transformed dataset, where each example is labeled with a *single* label which represents a combination of all possible labelings in the label powerset. Additionally, note that the BR transformation results in a set of binary classification datasets on which a separate binary classifier is afterwards trained.

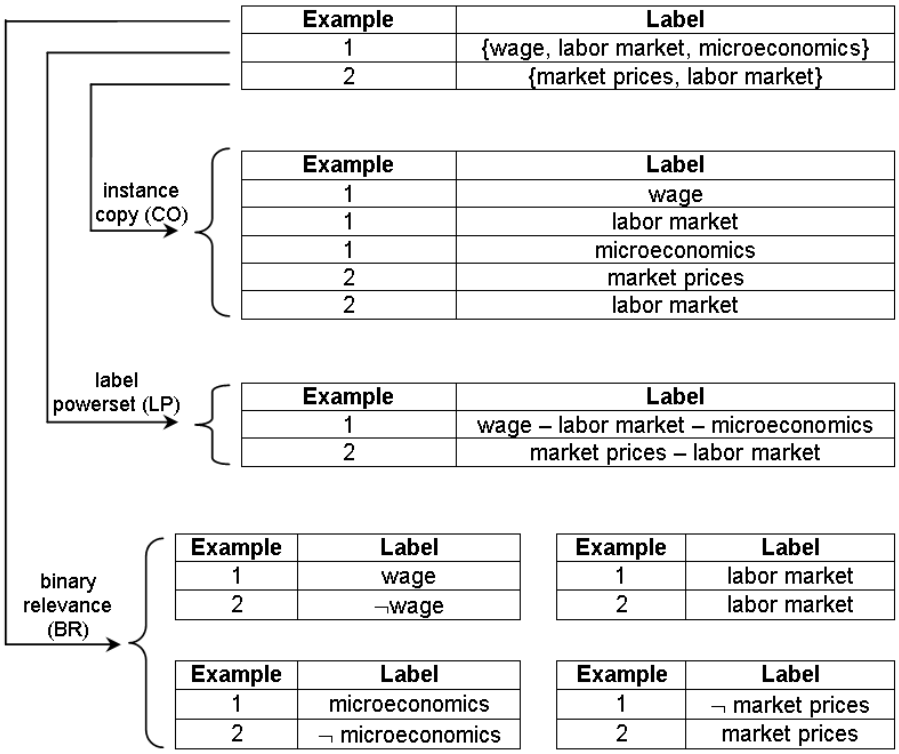

Fig. 3. Illustration of applying data transformation techniques (instance copy, label powerset and binary relevance) on an example multi-label dataset which contains two examples (in our case paper abstracts where each paper is labeled with a set of labels).

Since the above problem transformation approaches can be used with an arbitrary classifier that can output a probability distribution over classes, we coupled each of these approaches with one of the four following classification models:

- support vector machines (SVM) [31]: SVM implementation in Weka which uses a third-degree radial basis function kernel, γ*=1/(number of attributes)*, parameter *C=1* and sequential minimal optimization training,
- decision tree (DT) [32]: implementation of a recursive partitioning tree in Weka (J48) which splits nodes by examining the normalized information gain (difference in entropy) of attributes. Each leaf is assigned a majority class of the examples in the leaf,
- k-nearest neighbors (kNN): an implementation of the instance-based lazy learning algorithm IBk in Weka which uses 5 nearest neighbors for the

classification of a new example. Similarity of two examples is computed based on the Euclidean distance (all attributes are continuous),

- naive Bayes (NB) classifier: multinomial variant of the common naïve Bayes probabilistic classifier.

By combining each of the three transformation approaches with each of the four classifiers we therefore tested 12 different scenarios. Since the tested single-class models return the probabilities over class distributions, the labels can be ranked according to their predicted probability. In addition to the classification accuracy (Eqn. 2), their order was therefore evaluated using the average precision (Eqn. 3) as well. The experimental results are shown in Table 1.

Table 1: Performance of four different classifiers (decision tree – DT, support vector machine – SVM, naive Bayes – NB, k-nearest neighbors – kNN) coupled with three problem transformation approaches (binary relevance – BR, label powerset – LP, copy transformation – CO). The best achieved individual and average results are denoted by underlining. The table shows the average values of the performance measures and standard deviations.

CLASSIFICATION ACCURACY				
	BR	**LP**	**CO**	**average**
DT	0.282±0.024	0.311±0.040	0.318±0.048	**0.304**
SVM	0.053±0.010	0.225±0.034	0.289±0.046	**0.189**
NB	0.346±0.014	0.352±0.023	0.391±0.035	**0.363**
kNN	0.195±0.042	0.288±0.039	0.282±0.047	**0.255**
average	**0.219**	**0.294**	**0.320**	
AVERAGE PRECISION				
	BR	**LP**	**CO**	**average**
DT	0.531±0.022	0.397±0.040	0.450±0.040	**0.459**
SVM	0.255±0.019	0.374±0.035	0.434±0.035	**0.354**
NB	0.618±0.024	0.475±0.019	0.527±0.027	**0.540**
kNN	0.525±0.038	0.275±0.023	0.431±0.033	**0.410**
average	**0.482**	**0.380**	**0.461**	

We can see from the table that, on the average, the highest CA was achieved using the copy transformation approach (0.320) and the highest AP using the binary relevance transformation (0.482). Among the average performance of the classifiers, the naive Bayes performed the best with the average CA of 0.363 and the average AP of 0.540. By analyzing the results of each combination individually, we can see that the naive Bayes classifier achieved the best CA (0.391, coupled with CO transformation) and AP (0.618, coupled with BR transformation).

Note that classifying into the majority class (the most documents are labeled with a single category {*econometric_models*}) would give a default classification accuracy of 0.080. This means that the usage of all but one tested model-transformation combinations (namely, SVM-BR), outperforms

this default accuracy and shows a potential of using the machine learning algorithms to solve this task.

In the following Section 4.2, we move on to specialized multi-label classification models that can learn from and predict the data in their original multi-label form.

4.2. Multi-label models

The second group of methods extends the specific learning algorithms in order to handle multi-label data directly. In the previous section, we tried to solve multi-label problem by transforming the dataset to enable usage of the single-label models. However, these kinds of methods do not consider the correlations between the different labels. In the field, several approaches especially designed for multi-label learning tasks have been proposed, among which we evaluate the following two:

- back-propagation multi-label neural network (BP-MLL) [33]: This model is derived from the popular back-propagation algorithm through replacing its error function with a new function defined to capture the characteristics of multi-label learning, that is, the labels belonging to an instance should be ranked higher than those not belonging to that instance. The neural network was trained by epoch and contained one hidden layer of 20 neurons,
- multi-label k-nearest neighbors algorithm (ML-kNN) [18]: an adapted *k*-nearest neighbors lazy learning algorithm which retrieves the *k* nearest examples and aggregates the label sets of these examples by combining the statistical information gained from the neighboring instances. The algorithm uses normalized Euclidean distance as a distance function and parameter *k=10*.

The accuracy and the average precision of these two approaches are shown in Table 2.

Table 2: Results of multi-label models BP-MLL and ML-kNN, hierarchical model HOMER and ranking model CLR. The cells contain the average CA and AP values and their standard deviations.

	BP-MLL	**ML-kNN**	**HOMER (hierarchical)**	**CLR (ranking)**
Classification Accuracy	0.207±0.079	0.202±0.031	0.466±0.028	0.365±0.021
Average Precision	0.433±0.107	0.576±0.032	0.627±0.023	0.683±0.012

The results show that the ML-kNN achieves comparable CA to BP-MLL classifier, but outperforms it with better AP (0.576 compared to 0.433). If we compare these results to the ones in the previous subsection, we can see that the CA of the single-label naive Bayes classifier is on the average still better

than CA of any tested multi-label model. However, both multi-label models achieve slightly higher CA than the SVM on the average.

As for the AP, the multi-label classification approach seems to benefit from exploiting the dependence of the labels. Namely, ML-kNN achieves higher AP (0.576) than any of the single-class models on the average (the highest AP was on the average achieved using the naive Bayes – 0.540). However, note that the combination of the naive Bayes and the binary relevance transformation approach still achieves better AP of 0.618.

4.3. Hierarchical and ranking models

In our domain, there is a hierarchical dependence of the categories. This calls for evaluation of the additional models which are able to consider the relations between various labels (classes) in the hierarchy. As only 16 bottom-level ontological concepts were used so far (see Figures 1 and 2), to apply such classification method we need to expand the dataset we used so far with the additional classes, corresponding to the parent concepts in the ontological hierarchy (first-level and second-level concepts). Doing this, we introduce additional 7 classes, giving altogether 23 classes which are given to the learning algorithm along with the information about their inter-relations.

In this section, we evaluate a couple of methods that focus on dealing with problems with large number of labels by decomposing the original multi-label classification problem into a series of simpler problems [34]:

- HOMER (Hierarchy Of Multi-label classifiERs) [35,36] approach decomposes the problem into a hierarchy of simpler problems, where each problem uses a reduced number of possible labels. The hierarchical structure of the labels is obtained by applying recursive clustering to the initial set of labels. The main idea is the transformation of a multi-label classification task with a large set of labels L into a tree-shaped hierarchy of simpler multi-label classification tasks by recursively partitioning the set of labels into a number of nodes using a balance clustering algorithm. Then it builds one multi-label classifier at each node apart from leaves, following the hierarchical binary relevance approach [1]. A calibrated label ranking algorithm was used as an underlying multi-label classifier, and the number of selected clusters was 3.
- The Calibrated Label Ranking approach (CLR) [37] interprets a multi-label problem as a special case of a preference learning problem. Besides using an underlying classifier to rank the labels according to their prediction score, it also uses an additional neutral label which represents a breaking point of the ranking into relevant and irrelevant sets of labels. The binary models that learn to discriminate between the virtual label and each of the other labels correspond to the models of binary relevance. This way CLR can be used to perform the multi-label ranking task. In our experiments, the underlying binary models were built using the J48 decision trees.

The CA and the AP of HOMER and CLR are shown in Table 2. The results show that HOMER outperforms CLR in terms of its CA, but performs worse than CLR in terms of its AP. Compared to the results of the multi-label models (BP-MLL and ML-kNN) we can see that both, HOMER and CLR, achieve better CA and AP. Based on these results we can conclude that the learners from this section benefit from the information on hierarchical class structure in our problem domain.

Additionally, compared to the results of the single-class learners, applied to the transformed datasets (see Table 1), we can see that the CA and AP of HOMER are higher than the average CAs and APs of all models. However, the individual AP denoting the combination of naive Bayes and binary relevance transformation method seems to be only slightly worse than AP of HOMER and CLR. The visual comparison of performance of all tested models is shown in Fig. 4.

5. Conclusion

In the paper, we focused on the task of document classification in the field of economics, proposed an approach to this task and empirically evaluated how the selected machine learning methods are successful performing it.

We approached to the problem of **assigning the classes to documents** in the corpus by collecting a database of 1015 economic paper abstracts and semi-automatically constructing an ontology. We used an ontology tool which, during the construction, automatically distributed the documents among the concepts being recognized from the underlying text documents. Noting a scenario for the **multi-label classification approaches** we evaluated three groups of approaches to classify the documents: transformation to single-class problems, specialized multi-label models, and hierarchical/ranking models.

Classification accuracies of all tested classification models compared to the default majority classifier indicate that there is a potential for using the majority of the evaluated methods to solve this task. An exception to this rule is the SVM-BR model/transformation combination, which performed the worst classification accuracy of 0.053. The hierarchical multi-label classifier HOMER achieved the highest accuracy of 0.466. Given that classification accuracy in the context of multi-label classification is a very strict measure, which measures only the ratio of the exact matches between the true and the predicted labels, we can interpret this as a good result. By analyzing the performance using the average precision which is a less strict measure in terms of requiring the equivalence of the predicted and true label sets, the ranking approach CLR performed the best (average precision of 0.683).

The results show the benefits of using more complex (multi-label models, hierarchical and ranking) approaches, since they benefit from exploiting dependence between the labels. However, a good comparably-performing

alternative to these approaches is a single-class naive Bayes classifier which performed comparably to the best more complex approaches.

Our ideas for the further work include:

1. it shall be considered, how the described approached of ontology-based document classification can be further automatized, not requiring the interaction of the user,
2. alternative feature selection approaches shall be tested, along with the effect of using different numbers of selected attributes,
3. the usefulness of this approach shall be tested in other domains and using alternative document corpora (including larger documents instead of abstracts).

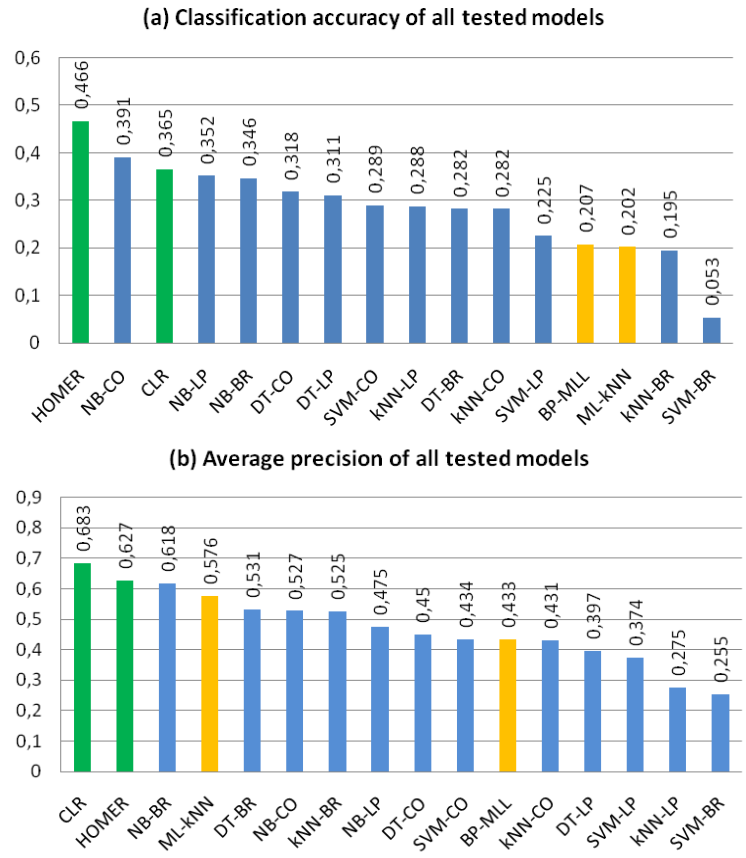

Fig. 4. Graphical representation of the obtained results: classification accuracies (above) and average precisions (below) of all tested models. The results are ranked in the decreasing order of the performance measure.

References

1. Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining Multi-label Data. In: Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook, 2nd edition, Springer, Heidelberg. (2010)
2. Noy, N. F., McGuinness, D. L.: Ontology Development 101: A Guide to Creating Your First Ontology. Knowledge Systems, AI Laboratory, Stanford University, No. KSL-01-05. (2001)
3. Blomqvist, E.: Fully Automatic Construction of Enterprise Ontologies Using Design Patterns: Initial Method and First Experiences. Lecture Notes in Computer Science, Vol. 3761. Springer-Verlag, Berlin-Heidelberg, 1314-1329. (2005)
4. Stanford Center for Biomedical Informatics Research: Protégé [Online]. Available: http://protege.stanford.edu/ (current March 2010)
5. Sure, Y., Angele J., Staab, S.: OntoEdit: Guiding Ontology Development by Methodology and Inferencing. In: Proceedings of the Confederated International Conferences on the Move to Meaningful Internet Systems CoopIS DOA and ODBASE 2002, Lecture Notes in Computer Science, Vol. 2519. Springer-Verlag, 1205-1222. (2002)
6. Farquhar, A., Fikes, R., Rice, J.: Tools For Assembling Modular Ontologies in Ontolingua. In: Proceedings of 14th American Association for Artificial Intelligence Conference (AAAI-97). AAAI/MIT Press., Menlo Park, CA, 436-461. (1997)
7. Stanford University: Chimaera [Online]. Available: http://www-ksl.stanford.edu/software/chimaera/ (current March 2010)
8. Fortuna, B., Grobelnik, M., Mladenić, D.: Semi-automatic Data-driven Ontology Construction System. In: Proceedings of the 9th International multi-conference Information Society IS-2006, Bohanec, M. et al. (eds.), Ljubljana, 223-226. (2006)
9. Fortuna, B., Grobelnik, M., Mladenić, D.: OntoGen: Semi-automatic Ontology Editor [Online]. Available: http://ontogen.ijs.si/ (current March 2010)
10. Zuniga, G. L.: An Ontology of Economic Objects. American Journal of Economics and Sociology, Vol. 2, No. 58, 299-312. (1999)
11. Siricharoen, W. V., Puttitanun, T.: Creating Ontology Chart Using Economy Domain Ontologies. International Journal of Digital Content Technology and its Applications, Vol. 3, No. 3, 74-80. (2009)
12. Mäki, U.: The Economic World View: Studies in the Ontology of Economics. Cambridge University Press, UK. (2001)
13. Fullbrook, E.: Ontology and Economics: Tony Lawson and His Critics. Routledge, Abingdon, UK. (2009)
14. Tsoumakas, G., Katakis, I.: Multi-label Classification: An Overview. International Journal of Data Warehousing and Mining, Vol. 3, No. 3, 1-13. (2007)
15. Vembu, S., Gärtner, T.: Label Ranking Algorithms: A Survey. In Fürnkranz, J., Hüllermeier, E. (eds.): Preference Learning. Springer-Verlag. (to appear 2010)
16. Yang, S., Kim, S.-K., Yong M. R.: Semantic Home Photo Categorization. In: Circuits and Systems for Video Technology, IEEE Transactions on, Vol. 17, No. 3, 324-335. (2007)
17. Qi, G. J., Hua, X. S., Rui, Y., Tang, J., Mei, T., Zhang, H. J.: Correlative Multi-label Video Annotation. In: MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia, New York, USA, 17-26. (2007)
18. Zhang M.-L., Zhou Z.-H.: ML-kNN: A Lazy Learning Approach to Multi-Label Learning. Pattern Recognition, Vol. 40, No. 7, 2038-2048. (2007)
19. Li, T., Ogihara, M.: Detecting Emotion in Music. In: Proceedings of the International Symposium on Music Information Retrieval, Washington D.C., USA, 239-240. (2003)

20. Schietgat, L., Blockeel, H., Struyf, J., Džeroski, S., Clare, A.: Decision Trees for Hierarchical Multilabel Classification: A Case Study in Functional Genomics. Lecture Notes in Computer Science, LNAI, Vol. 4213, 18-29. (2006)
21. Elisseeff, A., Weston, J.: A Kernel Method for Multi-labeled Classification. In: Advances in Neural Information Processing Systems 14, 681-687. (2002)
22. Clare, A., King, R. D.: Knowledge Discovery in Multi-label Phenotype Data. In: Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD 2001), Freiburg, Germany, 42-53. (2001)
23. Schapire, R. E., Singer, Y.: Boostexter: A Boosting-based System for Text Categorization. Machine Learning, Vol. 29, No. 2/3, 135-168. (2000)
24. McCallum, A.: Multi-label Text Classification with a Mixture Model Trained by EM. In: Working Notes of the AAAI`99 Workshop on Text Learning, Orlando, Florida. (1999)
25. Ueda, N., Saito, K.: Parametric Mixture Models for Multi-labeled Text. In: Becker, S., Thrun, S., Obermayer, K. (Eds.): Advances in Neural Information Processing Systems, Vol. 15, MIT Press, Cambridge, MA, 721-728. (2003)
26. Fortuna, B., Grobelnik, M., Mladenić, D.: OntoGen: Semi-automatic Ontology Editor. In: HCI International 2007, Beijing, China. Lecture Notes in Computer Science, Vol. 4558. Springer-Verlag, Berlin-Heidelberg, 309-318. (2007)
27. R Development Core Team: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. (2006).
28. Feinerer, I., Hornik, K., Meyer, D.: Text Mining Infrastructure in R. Journal of Statistical Software, Vol. 25, No. 5, 1-54. (2008)
29. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information Processing & Management, Vol. 24, No. 5, 513-523. (1988)
30. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I. H.: The WEKA Data Mining Software: An Update. SIGKDD Explorations, Vol. 11, Issue 1, 10-18. (2009)
31. Christiannini, N., Shawe-Taylor, J.: Support Vector Machines and Other Kernel-Based Learning Methods, Cambridge University Press. (2000)
32. Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J.: Classification and Regression Trees. Wadsworth International Group, Belmont CA. (1984)
33. Zhang M.-L., Zhou Z.-H.: Multi-label Neural Network with Applications to Functional Genomics and Text Categorization. IEEE Transactions on Knowledge and Data Engineering, Vol. 18, No.10, 1338-1351. (2006)
34. Tsoumakas, G., Mencia, L. E., Katakis, I., Park, S.-H., Fürnkranz J.: On the Combination of two Decompositive Multi-label Classification Methods. Workshop on Preference Learning, ECML PKDD 09, Hullermeir, E., Furnkranz, J. (Ed.), Bled, Slovenia, 114-133. (2009)
35. Tsoumakas, G., Katakis, I., Vlahavas, I.: Effective and Efficient Multilabel Classification in Domains with Large Number of Labels. In: Proceedings ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08), Antwerp, Belgium. (2008)
36. Koller, D., Sahami, M.: Hierarchically classifying documents using very few words. In: Proceedings of the 14th International Conference on Machine Learning (ICML-97), Nashville, 170-178. (1997)
37. Brinker, K., Hüllermeier, E.: Case-based Label Ranking. In: Machine Learning: ECML 2006, Vol. 4212, 566-573. (2006)

Sergeja Vogrinčič obtained her M.Sc. degree from Jožef Stefan International Postgraduate School, Ljubljana, Slovenia, in 2010. She has been employed at the same establishment, since 2006. Her research interests include machine learning, classification and ontology construction.

Zoran Bosnić obtained his M.Sc. and Ph.D. degrees from the University of Ljubljana, Faculty of Computer and Information Science (Slovenia) in 2003 and 2007, respectively. Since 2006 he has been employed at Faculty of Computer and Information Science and currently works as an assistant professor in the Laboratory of Cognitive Modelling. His research interests include artificial intelligence, machine learning, regression, and reliability estimation for individual predictions, as well as applications in these areas.

Received: April 20, 2010; Accepted: September 15, 2010.