elpub2002 Proceedings — J. A. Carvalho et. al. (Eds.) — VWF Berlin, 2002

Use of Text Mining Methods in a Digital Library[1]

Jiří Hynek[1] and Karel Ježek[2]

[1] inSITE, s.r.o., Knowledge Management Integrator
Rubesova 29, 326 00 Pilsen, Czech Republic
jiri.hynek@insite.cz
[2] Dept. of Computer Science & Engineering, University of West Bohemia
Univerzitní 22, Pilsen, Czech Republic
jezek_ka@kiv.zcu.cz

Abstract. The article deals with use of Itemsets classifier based on inductive machine learning in the context of digital library environment. We provide a brief description of a real-world digital library implemented at a power utility. Its implementation and operating experience have motivated our research in inductive machine learning methods for text mining described in the paper.
Being inspired by mining of association rules, we have developed a new categorization method named "Itemsets classifier". By performing various experiments we have proved its ability to surpass some well-known categorization methods, both in terms of precision/recall and efficiency. As the task of classification is closely related to clustering, we have integrated the principles of Itemsets method into a new document-clustering algorithm as well. We are also presenting other Itemsets classifier applications in unsolicited mail filtering and enhancement of the Naïve Bayes classifier. Main ideas and experimental results are presented in the paper.

1 Motivation – Digital Library and Classification Problems

Our research in text document classification based on inductive machine learning in the past few years was motivated by labor-intensive manual document classification in a digital library (part of the enterprise knowledge portal) we are operating

[1] Partly supported by grant No. MSM235200005

for Zapadoceska energetika, a.s., a regional power utility stationed in Pilsen, Czech Republic. The library currently serves more than 1,000 users and its taxonomy contains six hierarchies with more than 50 topics each. Accuracy of manual categorization of new entries (namely in case of informal documents, such as newsgroups entries) is not absolutely crucial (as we can afford some reasonable error rate). In addition, majority of staff members is using the system via search portal (part of the knowledge portal) based on our Uniseek search engine, neglecting the taxonomy structure at all.

The above-mentioned digital library includes multi-lingual documents categorized into a tree-like taxonomy implemented by means of a relational database. Library items are treated as objects interconnected by semantic links (e.g. between a document and a topic, or between two documents, etc.). There are currently more than ten thousand documents.

As the library is highly domain-focused, topics overlap, which makes automatic document categorization rather difficult. In addition, ever-growing number of documents makes us introduce new topics into the present topic hierarchy. Reclassification of formerly categorized documents is therefore necessary. We have recognized the necessity of automatic machine processing, as the volume of work makes manual classification prohibitive.

Most documents are in Czech, which is Slavonic language with typically rich morphology. In order to improve classification results (precision and recall), we are performing sophisticated dictionary-based morphological normalization (stemming) in addition to applying manually created stop-list.

2 Principles of Itemsets Classifier

We have described the principles of our novel approach to document classification using inductive machine learning in [3], [4] and [5]. Shortly, dimensionality of the original feature space is reduced substantially by selecting the so-called characteristic itemsets[3] describing each class in the topic hierarchy (taxonomy). Each category can be characterized either by a fixed or variable[4] number of itemsets. Documents to be classified are then compared with characteristic itemsets of each class and consequently categorized upon performing some weighing and thresholding. We have achieved very goods results (precision and recall over 90%) for extremely short documents (less than 30 to 50 significant terms), using 1-itemsets only. As we can conclude from our practical tests, *Itemsets classifier* can be well trained on classes containing more than 50 documents.

[3] Various term- and itemset weight computations take place in order to avoid selecting itemsets with low discriminatory power.

[4] Some form of normalization during classification phase is required.

Our further research will be focused on combining filter-based feature space reduction (as described above) with wrapper-based approach, i.e. optimizing the number (and/or selection) of characteristic itemsets associated with taxonomy categories in line with classification results, applying genetic algorithms and treating Itemsets classifier as a parametric black box.

3 Application of Itemsets Classification Method – Document Clustering

Cluster analysis can be helpful for solving a classification problem when little or nothing is known about the taxonomy structure of a document collection[5]. It can be also beneficial when the collection is a dynamic one and new categories must be added continuously or existing categories merged. We have proposed a new clustering algorithm using the *Itemsets classifier* to improve results of traditional clustering methods, using CLUTO (*CLUstering TOol*) package as a basis of our application. CLUTO is a software package for clustering low and high dimensional datasets and for analyzing the characteristics of various clusters. For more information on CLUTO, see `http://www-users.cs.umn.edu/~karypis/cluto`.

The main idea of *Itemsets clustering* is as follows: We start document clustering process with a traditional method suitable for clustering high dimensional patterns. Clusters thereby created are marked as *regular* or *non-regular*, depending on their size. Regular clusters (i.e. clusters of satisfactory size) are used for training the *Itemsets classifier* and producing characteristic itemsets for each of regular clusters. New documents are assigned to regular clusters by the *Itemsets classifier* if they fit to these. Otherwise, documents are assigned to *non-regular* clusters using a traditional clustering method. New clusters can be created as well. When a non-regular cluster becomes a *regular* one due to its increased size, Itemsets classifier must be trained on the new regular cluster. Time requirements are not extremely crucial (albeit reasonable, as indicated by our experiments), as the whole process is performed off-line.

We have developed an iterative non-hierarchical clustering algorithm that is partly based on novel *itemsets classification* approach. As we are currently optimizing the application, results will be published later. The first tests indicate improvements in the order of several per cent compared to CLUTO algorithm alone.

[5] It is often the case that formal documents (such as directives, standards, reports, etc.) are used for taxonomy building by a clustering algorithm, whereas informal documents (such as news, newsgroup entries, email messages, etc.) are later categorized by means of some classification algorithm (trained on the taxonomy already created).

4 Anti-Spam Filter Based on Itemsets Classifier

Commercial as well as freeware anti-spam programs facilitate spam filtering based, for example, on sender's address (user's name, domain name), subject of email, message text, or message headers (such as *Cc*, *priority*, *Reply-to*, etc.) or other parameters. An extensive set of filters is usually predefined by software authors. Users of these applications can provide sophisticated setup information to tune up anti-spam filtering process. Spam killing is then based more or less on occurrence of specific keywords (predefined or selected by users) in various sections of email messages.

Our anti-spam filter, however, is based on inductive machine learning approach. *Itemsets classifier* is trained on a sufficiently large set of unsolicited emails without user's assistance. Upon the inductive machine learning phase, incoming messages are classified into two topics – *non-spam* and *spam*. Further classification into user's email folders can be performed as well, either using *Itemsets classifier*, keyword matching, from-address matching, or any other approach.

Results of Itemsets inductive learning method can be also used to integrate characteristic itemsets thereby found to specify user-configurable keywords in any commercial anti-spam filter.

We have performed various tests to find out whether the *Itemsets classifier* would be convenient for automating time-consuming categorization of e-mail messages. The method was tested on a collection of both legitimate and unsolicited messages, dubbed *CIV corpus*[6] (provided by CIV of the University of West Bohemia, Pilsen, Czech Republic) containing e-mail messages from electronic conferences. We have perused these corpora and randomly chosen 1,375 messages written mainly in the English language (approximately 10 % of messages was manually identified as spam). The length of messages is specified in the table below.

Table 1. Length of email messages in CIV corpus

	minimum	maximum	average
all terms	7	2 777	180
significant terms	7	1 799	136

The average length of 136 significant terms is favorable for *Itemsets classifier*, which is ideal for categorizing (extremely) short text documents. Ten-fold cross-validation was applied to reduce random variations and the effect of attribute-set

[6] CIV donated six corpora containing a few thousand e-mail messages dispatched by users to electronic conferences.

size, application of stemming and stop-list. Legitimate-to-spam messages' cost ratio was also taken into account. Efficiency of our spam filter was assessed using the table below.

Table 2. Assessing the quality of itemsets-based anti-spam filter

	Is it really a spam?	
Identified as a spam?	Yes	No
Yes → Delete	a^{OK}	b^{error}
No → Keep	c^{error}	d^{OK}

Relevance factor applicable to **b** must be several orders of magnitude (e.g. λ = 1,000) higher than that of **c** (i.e. we desire to prevent deleting legitimate messages). In other words, blocking a legitimate message is λ-times more costly than misclassifying a spam message. *Classification accuracy* (i.e. filter accuracy) can be defined as (a+d) / (a+b+c+d). As we are performing mono-classification, we can also define the *error rate* as the proportion of incorrectly classified email messages (equal to *1 – accuracy*). Let's also define *Precision* as a / (a+b) and *Recall* as a / (a+c).

The table below shows results (i.e. classifier's *error rate*) achieved on the CIV corpus when applying *Itemsets classifier* (using 1-itemsets only):

Table 3. Error rate of Itemsets classifier applied as a spam filter

Itemsets	1.52 %
Itemsets + stemmer	2.18 %
Itemsets + stop-list	1.30 %
Itemsets + stemmer + stop-list	1.96 %

The above results were achieved with inductive learning based on 200 features. Figure 1 depicts dependence of filter accuracy on the number of features used.

Upon applying larger itemsets, time requirements increased tremendously, while achieving only very little improvement (0.1 - 0.2 %).

We must stress the importance of λ (classifying a legitimate message as spam is λ-times more costly than classifying spam as a legitimate message) in view of classification *precision* and *recall* (see fig. 2 and 3).

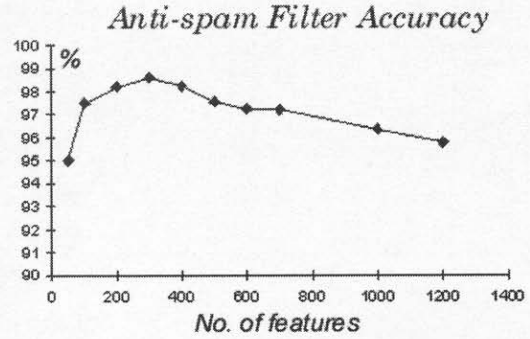


Fig. 1. *Anti-spam filter accuracy vs. number of features used in inductive machine learning process.*

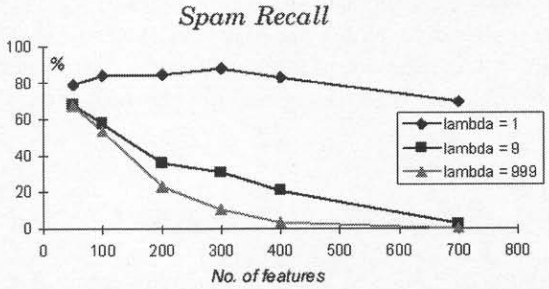


Fig. 2. *Spam recall vs. number of features for various λ settings.*

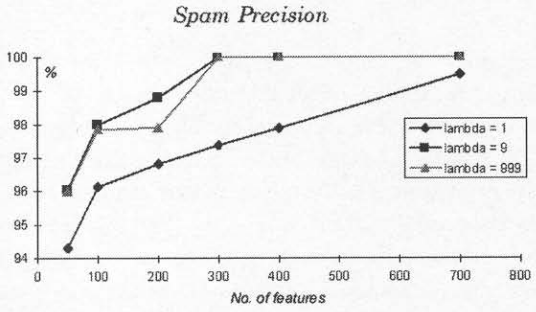


Fig. 3. *Spam precision vs. number of features for various λ settings.*

An anti-spam filter was tested, for example, by Androtsopoulos [6] using Naïve Bayes classifier, with the best results of spam *recall* approximately 84 % and spam

Table 4. Best average results obtained for various values of λ.

λ	No. of features	Spam recall (%)	Spam precision (%)
1	300	**88.00**	**97.38**
9	50	68.00	96.05
999	50	67.00	96.00

precision approximately 91 %. We have to note that a corpus of personal e-mail messages and spam messages was used and corpus size (in the number of e-mail messages) was comparable with ours.

Androtsopoulos also tried functions of Microsoft Outlook 2000 providing anti-spam filter based on keyword patterns. There are 58 patterns, looking for particular keywords in the body or header fields of messages (e.g. "Body contains" ',000' AND Body contains "!!" AND Body contains "$""). In this case, spam *recall* was approx. 53 % and spam *precision* approx. 95 %.[7]

5 Modification of Naïve Bayes Classifier Using Itemsets

The Naïve Bayes classifier is based on a simplifying assumption of conditional dependence of attribute values of the target value. In other words, conjunction is represented by a product of probabilities of individual attribute values, i.e.: $P(a_1, a_2, ...a_n \mid v_j) = \prod_i P(a_i|v_j)$, where v_j represents a category in the taxonomy structure.

In order to determine the probability of $P(a_i|v_j)$ we will use the following formula[8] for estimating probability of terms in document collection, i.e. $P(a_i \mid v_j = \frac{n_i+1}{n+|vocabulary|}$ where $|vocabulary|$ is the total number of distinct significant terms in the collection of training data, n is the total number of word positions in all training instances with the target value of v_j, and n_i is the number of a_i occurrences over these word positions. By analogy, $P(\Pi_i|v_j) = \frac{n_i+1}{n+|itemsets|}$ represents probability of occurrence of frequent itemset Π_i in documents in category v_j, where n indicates the number of occurrences of frequent itemsets of the same size as Π_i in documents in class v_j, and n_i is the number of occurrences of Π_i among these n occurrences.

[7] More information on spam filtering can be found at http://spam.abuse.net or http://www.junkemail.org. Various anti-spam filters are freely available on the Internet, e.g. http://spamkiller.com, http://spammotel.com, http://www.hms.com/spameater.asp, or http://www.mailwasher.net.

[8] This approach is also referred to as *Laplace smoothing.*

Traditional *Naïve Bayes classifier* is then defined as:

$$v_{NB} = \arg\max P(v_j) \prod_i P(a_i|v_j)$$

We have proposed *Itemsets modification* as follows: Instead of working with word attributes (a_i), we may use frequent itemsets Π of various size to compute $v_N B$. Over the course of classification, we will utilize frequent itemsets that have been determined for each class. This leads to significant feature space reduction. Only frequent itemsets of a given class contained in the instance A being classified are utilized. The formula is therefore changed to

$$v_{NBCI} = \arg\max P(v_i) \prod_i P(\Pi_i^{v_j}|v_j)^{wf_l} \ , \ L = \left|\Pi_i^{v_j}\right| ,$$

where $\Pi_i^{v_j}$ represents i-th frequent itemset of the class v_j, that also occurs in the instance (document) being classified, wf_L represents weight factor applicable to corresponding L-itemsets (weight ranging from 0 to 1). An itemset Π was declared frequent in the class v if its $P(\Pi|v)$ exceeded some *minsupport* value. We have named the resulting classifier *NBCI* (Naïve Bayes Combined with Itemsets).

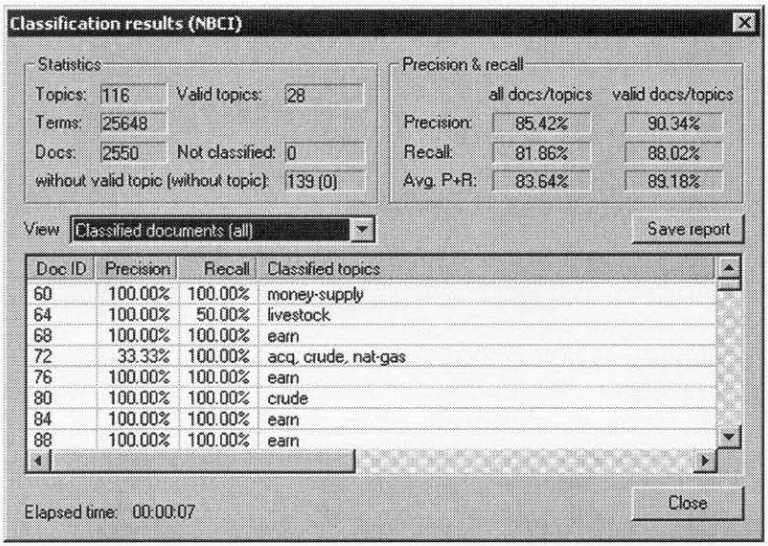


Fig. 4. *An example output by NBCI classifier.*

We have used *Reuters-21578* collection containing newswire stories for testing both Naïve Bayes and NBCI classifiers. 10 largest topics were used for testing (representing more than 80 % of all documents, with at least 150 documents per category).

We have achieved excellent results while using NBCI classifier to categorize documents into two largest topics only (F_1 approx. 98 %). Therefore, it is likely that the method could be used successfully as a part of an anti-spam filter.

Table 5. NBCI results achieved using different partitioning of Reuters-21578 collection

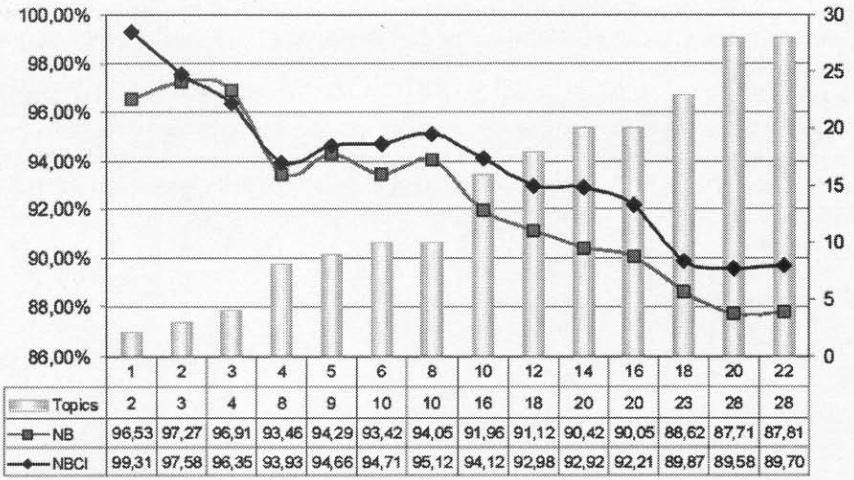
Topics used for categorization by NBCI	**Precision [%]**	**Recall [%]**	**F_1 measure**
All 116	89.91	86.14	**87.98**
28 topics with at least 50 documents (coverage of 94.7 % of all documents)	91.85	88.74	**90.27**
10 largest topics (coverage of more than 80 % of documents)	94.54	93.14	**93.84**
earn and acq topics (approx. 58 % of all documents)	97.95	97.75	**97.85**

Needed to say that usage of itemsets larger than 1 does not result in improving classification results of NBCI. Moreover, generation of itemsets of larger size is very time-consuming. Similar conclusion can be applied to the original *Itemsets classifier* as such.

Comparing Naïve Bayes classifier with NBCI approach, we can see that both approaches are based on the same algorithm. The novelty of NBCI is vested in different pre-processing of entry data and alternate computation/use of weights during both training and classification phase. When using 1-itemsets only, NBCI inductive learning phase is faster compared to NB. On the other hand, classification phase is quicker in case of NB, as Naïve Bayes uses document terms directly, whereas NBCI must look for itemsets contained in documents to be classified.

Figure 5 depicts average of *Precision* and *Recall* of both NB and NBCI algorithms with the increasing number of Reuters data sets used for training the classifier (curves can be decreasing, because adding a new data set can result in integrating additional "not-sufficiently-trained" topics to categorization process). NBCI outperforms Naïve Bayes by several per cent.

Modification of Naïve Bayes classifier using itemsets is also described by Meretakis and Wüthrich [7]. *Large Bayes classifier* proposed Meretakis and Wüthrich is reduced to Naïve Bayes classifier when all itemsets are of size one only (i.e. no feature space reduction takes place). Support of itemsets is determined with respect to their occurrence in the whole document collection (using F as a global set of all interesting and frequent itemsets), not a particular class (our approach). They



	1	2	3	4	5	6	8	10	12	14	16	18	20	22
Topics	2	3	4	8	9	10	10	16	18	20	20	23	28	28
NB	96,53	97,27	96,91	93,46	94,29	93,42	94,05	91,96	91,12	90,42	90,05	88,62	87,71	87,81
NBCI	99,31	97,58	96,35	93,93	94,66	94,71	95,12	94,12	92,98	92,92	92,21	89,87	89,58	89,70

Fig. 5. *Comparison of Naïve Bayes classifier with NBCI depending on the number of Reuters data sets used.*

work with the largest possible itemsets, leaving out shorter itemsets contained in larger ones.

6 Conclusions

We have proved viability of *Itemsets classifier* by its integration into several experimental applications. Some areas of its use and practical results are mentioned in the paper. Our further research will be focused namely on optimization of inductive machine learning process based on itemsets and practical implementation of this method in other applications, such as document clustering and expert finding.

References

1. Agrawal et al.: *Advances in Knowledge Discovery and Data Mining*, MIT Press 1996, 307-328
2. Dumais S., Platt J., Heckerman D., Sahami M.: *Inductive Learning Algorithms and Representations for Text Categorization*, CIKM 98, Bethesda MD, U.S.A.
3. Hynek J., Ježek K.: *Automatic document classification using Itemsets method, its modifications and evaluation*, Proceedings of the Annual International Database Conference Datakon 2001, Brno, Czech Republic, October 2001, ISBN: 80-227-1597-2
4. Hynek J., Ježek K., Rohlík O.: *Short Document Categorization – Itemsets Method*, International Conference PKDD 2000, Lyon – France, September 2000

5. Hynek J., Ježek K.: *Document Classification Using Itemsets* Proceedings of International Conference MOSIS 2000, Rožnov pod Radhoštěm, Czech Republic, May 2000, ISBN: 80-85988-45-3
6. I. Androutsopoulos, J. Koutsias, K. V. Chandrinos and C. D. Spyropoulos, *An Experimental Comparison of Naive Bayesian and Keyword-Based Anti-Spam Filtering with Personal E-mail Messages.* In Belkin, N. J., Ingwersen, P. and Leong, M.-K. (Eds.), Proceedings of the 23^{rd} Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2000), Athens, Greece, pp. 160-167, 2000.
7. Meretakis D., Wüthrich B.: *Extenting Naïve Bayes Classifiers Using Long Itemsets*, KDD-99, San Diego, California, 1999, pp. 165-174