# Short Document Categorization - Itemsets Method

**Jiri Hynek**[*]
jiri.hynek@insite.cz

**Karel Jezek**[**]
jezek_ka@kiv.zcu.cz

**Ondrej Rohlik**[**]
rohlik@kiv.zcu.cz

**Abstract**: The essential point of this paper is to develop a method for automating time-consuming document categorization in a digital library. The method proposed in this paper is based on itemsets, extending traditional application of the apriori algorithm. It is suitable for automatic categorization of short documents (abstracts, summaries) impeding usage of repeated occurrence of terms, such as in term-frequency-based methods. The paper presents basic principles of this method as well as preliminary results of an on-going research. The method is designed to fit to an extensive commercial application.

**Keywords**: itemset, classification, class generation, cluster, clustering, apriori algorithm, document similarity, document categorization, electronic library, digital library.

## 1 Introduction

Creating a digital library represents a challenging task, requiring considerable financial as well as human resources. Documents in the library are mostly subject to copyright and one must pay for having them stored in the library. Document categorization requires a domain expert deciding on appropriate topic class or classes.

Abstracts of technical articles are mostly freely accessible on the web. It is therefore possible to create an extensive library of such abstracts. Users of the library can then make a request to buy a full copy of a document or its translation. The task of document searching in electronic library is similar to the one of categorization, being solved by means of similar principles.

Digital library used while implementing the classifier represents a real library in a commercial environment. Majority of its documents focus on electrical engineering, electricity market, transmission and distribution of electricity and telecommunications. Its current parameters are as follows:

| | |
|---|---|
| The number of documents (abstracts) in the library | 2,200 |
| The number of distinct significant terms | 16,700 |
| Average length of a document in the library[1] | 87 |
| The longest document in the library[1] | 440 |
| The shortest document in the library[1] | 19 |
| The number of topics (classes) | 92 |
| Average number of classes a document is classified to | 4 |
| The number of stop words in the stop list | 142 |
| The most frequent significant term | „energy“ |

[*] *inSITE, s.r.o.*, Knowledge Management Integrator, Rubesova 29, 301 53 Plzen, Czech Republic

[**] Department of Computer Science, the University of West Bohemia, Univerzitni 22, Plzen, Czech Republic

[1] Expressed in the number of significant terms

Upon implementing a simple stemming engine, the number of distinct significant terms has been reduced by 43 %. The volume of index files has instantly dropped by 10 %. Although the stemming method used is a very simple one (cutting off the longest possible word endings), it proved very efficient in the commercial environment. We are designing a more sophisticated alternative of the stemming engine, utilizing complete corpus of the Czech language, including terms with irregular declensions.

By leaving out insignificant terms (contained in the stop-list), the number of words in the digital library dropped by 25 %. We have not observed any variation of this ratio in the long-term.

Digital library is highly specialized and some of its classes tend to overlap. Mostly (92 %) it is not possible to classify a document to one class only. Most documents are categorized into three classes, on the average to four, the range spans from 1 to 10.

Arrangement of topics complies with the organizational structure of the library's user. It has not been designed to facilitate document categorization. Some topics we added on the fly as needed, without re-classifying documents inserted in the past. This fact has a negative impact on classifier training.

## 2 Itemsets and Apriori Algorithm

The apriori algorithm (Agrawal et al.) is an efficient algorithm for knowledge mining in form of association rules [2]. We have recognized its convenience for document categorization. The original apriori algorithm is applied to a transactional database of market baskets. In our case, instead of a market basket, we work with the basket of significant terms occurring in a text document and the transactional database is in fact a set of documents (represented by sets of significant terms). Consistently with the usual terminology let us denote terms as items and basket of terms (set of items) as an itemset.

Let $\pi_i$ is an item, $\Pi = \{\pi_1, \pi_2, \ldots, \pi_m\}$ is an itemset and $\Delta$ is our database of documents. The itemset with k items is called k-itemset. Frequency of an itemset is defined as a simultaneous occurrence of items in the data being observed. Within our investigation we often utilize the threshold value employed for the minimum frequency of an itemset. Should frequency of an itemset exceed this threshold value, it is designated as a *frequent itemset* The transaction support in our case corresponds to the frequency of an itemset occurrence in the database $\Delta$. Our goal is to discover frequent itemsets in order to characterize individual topics in the digital library.

Frequent itemsets' searching is an iterative process. At the beginning all frequent 1-itemsets are found, these are used to generate frequent 2-itemsets, then frequent 3-itemsets are found using frequent 2-itemsests, etc.

Let us suppose we have $TD_S$ distinct significant terms in our text database $\Delta$. Firstly we generate candidates of frequent 1-itemsets (shortly candidate 1-itemsets). These are contained in our application directly in DF (Document Frequency) table. Consequently, we compute frequent 1-itemsets. In the next step, we generate 2-itemsets from frequent 1-itemsets. Generation of subsequent candidate and frequent *n*-itemsets continues until the process of frequent itemsets' searching terminates with regard to Apriori property ("all non-empty subsets of a frequent itemset must be frequent"). While implementing this method, we utilize a technique similar to transaction reduction method: a document that does not contain a k-itemset can be left out of our further consideration, since it cannot contain any of (k+1)-itemsets.

Let $Ç_k$ denote a set of candidate k-itemsets and $F_{k-1}$ a set of frequent (k-1)-itemsets. Generation of $Ç_k$ from $F_{k-1}$ is based on the following algorithm:

$Ç_k := \varnothing$;
for $\forall$ document
for $\forall\ \Pi_i \in Ç_{k-1}$ do
for $\forall\ \Pi_j \in Ç_{k-1}$ do
if (first k-2 items in $\Pi_i$ and $\Pi_j$ are identical, but last items differ)
then begin $c := \Pi_i$ join $\Pi_j$ ;
if $\exists$ subset s, $s \subset c$ having k-1 elements, where $s \notin F_{k-1}$
then do not append c to $Ç_k$
else append c to $Ç_k$ ;
end;

## 3 Itemsets Classification Method

### 3.1 Notation Used in the Paper

Within the framework of this paper, we use the following notation:

| | |
|---|---|
| $\Pi_I$ | Frequent itemset |
| T | Topic (representing a categorization class) |
| D | Document |
| $\overline{D}$ | A set of significant terms contained in document D |
| L | The number of topics |
| $N_i$ | The number of frequent itemsets having cardinality *i* |
| $D\Pi_i$ | The set of documents containing the itemset $\Pi_i$ |
| $\|D\Pi_i\|$ | The number of documents containing the itemset $\Pi_i$ |
| $DT_i$ | The set of documents associated with topic $T_i$ |
| $\|DT_i\|$ | The number of documents associated with topic $T_i$ |
| $C_i$ | Set of itemsets characterizing topic $T_i$ |
| $\|C_i\|$ | The number of itemsets characterizing topic $T_i$ |

On the basis of the apriori algorithm above, we will define frequent itemsets of various cardinalities. For 1-itemsets $\Pi_1$, $\Pi_2$, …, $\Pi_{N1}$, for pairs $\Pi_{N1+1}$, $\Pi_{N1+2}$, …, $\Pi_{N1+N2}$, for triplets $\Pi_{N1+N2+1}$, $\Pi_{N1+N2+2}$, …, $\Pi_{N1+N2+\ N3}$ etc.

### 3.2 The Classification Problem

The classification problem can be divided into two parts: *training phase* and *classification*[2] *phase*. The training phase consists of the following:

- Defining a hierarchy (tree) of thematic areas (topics) by a domain expert; L categories are thus defined.
- Manual insertion of a certain number of documents into topics, i.e. classification attributes are defined for each class (training data set). A domain expert performs categorization of

[2] A classifier is a function mapping a vector of terms contained in document D onto a set of topics (classes): $f(\overline{D}) = \{topics\}$

all "training" documents available. Each topic should be assigned a statistically significant number of documents.

- Automatic generation of representative itemsets of various cardinality for each topic.

While performing classification, we utilize representative itemsets to classify documents into corresponding topics.

The classification algorithm can be evaluated in terms of accuracy (*precision* and *recall* parameters) and speed. Accuracy can be measured by means of a test-set, the members of which have a priori known classification.

*Precision*: P = p/q

*Recall*: Q = p/r

Where p = the number of classes determined correctly by the classifier (automatically); q = total number of classes determined automatically; r = the number of classes determined by a domain expert (manually, i.e. correctly).

### 3.3 Phases of Itemsets Method

Training Phase

For each itemset $\Pi_j$ we can find a characteristic set of documents containing $\Pi_j$. Let's designate this set of documents as $D\Pi_j$. It is obvious that cardinality of $D\Pi_j$ will be higher than a certain threshold value, since $\Pi_j$ was selected as a frequent itemset.

Itemset $\Pi_1$ corresponds to the set $D\Pi_1$, $\Pi_2$ corresponds to $D\Pi_2$, etc. If we will work with singles, pairs, triplets, …, we will create N1 + N2 + N3 + … sets of documents.

By analogy, for each topic $T_i$ there is a characteristic set of documents falling into this topic. Let's designate this set as $DT_i$. Topic $T_1$ corresponds to the set $DT_1$, topic $T_2$ to $DT_2$, etc. Altogether we will make L sets.

Our goal is to specify a certain number of itemsets for each topic, where each itemset is associated with a subset of the set of topics. Namely, itemset $\Pi_j$ is associated with topic $T_i$ corresponding to the values of $w_{\Pi j}$ exceeding some threshold value. The weight $w_{\Pi j}$ is calculated, for example, as follows[3]:

$$w_{\Pi j} = \frac{\left|D\Pi_j \cap DT_i\right|}{\left|DT_i\right| \times \left[1 + \left|D\Pi_j\right| - \left|D\Pi_j \cap DT_i\right|\right]} \qquad i = 1, 2, ..., L$$

Denominator is used for normalizing with the number of documents associated with topic $T_i$. It takes into account whether an itemset occurs in other topics as well. Significance of terms occurring frequently in documents other than $DT_i$ is thus suppressed.

Upon associating itemsets with individual topics based on the formula above, we will acquire sets of itemsets $C_i$ representing a particular topic[4] $T_j$. On the whole, there will be L sets of itemsets.

Classification Phase

Within the process of document classification, we must take into account cardinality of itemsets in order to distinguish between correspondence in pairs and correspondence in

[3] This is, of course, an ad-hoc approach. We have tried various formulas leading to various results. It is likely that we will come up with a different formula for the final version of this method.

[4] Each topic is currently represented by a set of itemsets of fixed size.

quadruplets, for instance. That is why we define a weight factor corresponding to the cardinality of an itemset. For pairs we will use $wf_2$, for triplets $wf_3$, for quadruplets $wf_4$, etc.

Now we can proceed with classifying a document into a topic (or several topics). Let's suppose, set $C_j$ contains elements $\Pi_1$, $\Pi_2$, …, $\Pi_{|Cj|}$. We will compute the weight corresponding to the accuracy of associating document D with topic $T_j$:

$$W_{T_j}^{D} = \sum_{i=1}^{|C_j|} wf_{|\Pi_i|} \times w_{\Pi_i} \quad where\ (\Pi_i \in C_j) \wedge (\Pi_i \subseteq \overline{D})\ for\ all\ j = 1, 2, \ldots, L$$

In other words, the classification weight is determined by the sum of products of weights $w_{\Pi i}$ with weight factors $wf_{|\Pi i|}$ for all itemsets of a given topic, which (the itemsets) are contained in the document being classified. Usage of $w_{\Pi i}$ results in emphasizing those itemsets that provide the best description of topic $T_j$.

The document D will be associated with topic $T_j$ corresponding to the highest weight $W_{T_j}^{D}$. Naturally, we can desire to associate the document with several topics. If this is the case, we will classify the document D to all topics $T_j$ where $W_{T_j}^{D}$ exceeds certain threshold value θ. Increasing the value of θ results in decreasing precision of classification, although the recall parameter increases. If we vary the value of θ, precision and recall move in opposite directions.

## 4 Evaluation and Further Research

Results achieved so far:

| Parameter \ Value | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Parameter altered | θ | | | | $wf_1$ and $wf_2$ | | | | α | | | | |
| Min. frequency 1-itemsets | 2,1 % | 2,1 % | 2,1 % | 2,1 % | 2,1 % | 2,1 % | 2,1 % | 2,1 % | 0,6 % | 0,6 % | 0,6 % | 0,6 % | 0,6 % |
| Min. frequency 2-itemsets | 2,1 % | 2,1 % | 2,1 % | 2,1 % | 2,1 % | 2,1 % | 2,1 % | 2,1 % | N/A | N/A | N/A | N/A | N/A |
| Theta θ (%) | **75** | **60** | **40** | **30** | 75 | 75 | 75 | 75 | 75 | 75 | 75 | 75 | 75 |
| Characteristic 1-itemsets | 680 | 680 | 680 | 680 | 680 | 680 | 680 | 680 | 1,943 | 1,943 | 1,943 | 1,943 | 1,943 |
| Characteristic 2-itemsets | 5,123 | 5,123 | 5,123 | 5,123 | 5,123 | 5,123 | 5,123 | 5,123 | N/A | N/A | N/A | N/A | N/A |
| $wf_1$ | 10 | 10 | 10 | 10 | **9** | **7** | **5** | **4** | 10 | 10 | 10 | 10 | 10 |
| $wf_2$ | 1 | 1 | 1 | 1 | **3** | **4** | **5** | **6** | 0 | 0 | 0 | 0 | 0 |
| α - Number of 1-itemsets considered characteristic | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | **500** | **400** | **350** | **300** | **100** |
| β - Number of 2-itemsets considered characteristic | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | N/A | N/A | N/A | N/A | N/A |
| ρ - Manual classification needed [%] | 29 | 24 | 18 | 13 | 31 | 32 | 34 | 34 | 66 | 39 | 27 | 17 | 21 |
| **P – Precision** | 63 | 62 | 57 | 53 | 62 | 61 | 60 | 60 | 30 | 53 | 63 | 72 | 70 |
| **R – Recall** | 33 | 38 | 47 | 55 | 32 | 31 | 30 | 30 | 12 | 29 | 36 | 43 | 39 |
| **P* - Refined precision** | 89 | 81 | 70 | 61 | 90 | 91 | 91 | 92 | 93 | 87 | 87 | 87 | 88 |
| **R* - Refined recall** | 46 | 50 | 57 | 63 | 46 | 46 | 46 | 45 | 36 | 48 | 49 | 52 | 49 |
| **P and R average** | 48,0 | 50,0 | 52,0 | **54,0** | **47,0** | 46,0 | 45,0 | 45,0 | 21,0 | 41,0 | 49,5 | **57,5** | 54,5 |
| **P* and R* average** | **67,5** | 65,5 | 63,5 | 62,0 | 68,0 | 68,5 | 68,5 | **68,5** | 64,5 | 67,5 | 68,0 | **69,5** | 68,5 |

The minimum number of documents in a class = **50** (applicable to all values above).

α, β = The number of best 1-itemsets (2-itemsets) declared characteristic for a given topic.

α + β = Cardinality of $C_i$ class

ρ = Proportion of documents that need manual classification (automatic classifier failed).

P* and R* (refined precision and recall) - Applies to documents that were classified (better or worse) by the classifier – i.e. P* and R* statistics does not include documents that were classified with zero precision and recall.

Precision and recall achieved demonstrate high sensitivity to the first two parameters (min. frequency of 1-/2-itemsets). These factors have the highest impact on memory requirements in the training phase (size of the hashing table).

Values 1-4: $\theta$ being tuned, observing the impact on precision and recall of the classifier. Decreasing value of $\theta$ causes decreasing precision and increasing recall.

Values 5-8: **wf1** and **wf2** being tuned, observing the impact on precision and recall of the classifier. Increasing the impact of characteristic 2-itemsets at the classification phase causes a slight reduction in precision, and a slight increase in refined precision P*.

Values 9-13: $\alpha$ and $\beta$ parameters being tuned: too high a number of characteristic itemsets make classifier confused by introducing too much irrelevant information (see the impact on P and R). On the other hand, insufficient number of characteristic itemsets causes decreasing precision as well as recall.

Low average length of a document (80 terms) would cause problems in case of TF×IDF-based classifier, however, it is beneficial in our method. We are not utilizing repeated occurrence of terms, but rather specific pairs and triplets, which are declared characteristic.

The complexity of the apriori algorithm is much dependent on selecting a threshold value for declaring an itemset frequent or non-frequent. Implementation leads to a reasonable polynomial-bound problem.

We are not presenting a comparison with other available methods, since our method was applied to the Czech corpus only (because of stemming of this language containing complex grammatical rules for word declensions). Other methods (naïve Bayes, etc.) are currently being implemented to provide a rigorous comparison applicable to short documents in various languages.

The classification method proposed in this paper shows viable within the commercial application it will be integrated with. Preliminary results indicate that by implementing the method the task of document classification becomes more efficient and less time-consuming.

## References


1. *Bulletin of the Technical Committee on Data Engineering*, June 1998, Vol. 21, No. 2, IEEE Computer Society
2. Agrawal et al.: *Advances in Knowledge Discovery and Data Mining*, MIT Press 1996, pp. 307-328
3. Mladenic D., Grobelnik M.: *Word Sequences as Features in Text Learning*, Proc. Seventh Electrotechnical and Computer Science Conf. (ERK 98), IEEE Region 8, Slovenia Section IEEE, 1998, pp. 145 – 148
4. Hynek J., Ježek K.: *Document Classification Using Itemsets*, Proc. 34th Spring International Conference – Modeling and Simulation of Systems, MOSIS 2000, pp. 97-102