# Hybrid Approach Combining Machine Learning and a Rule-Based Expert System for Text Categorization

**Julio Villena-Román**[1], **Sonia Collada-Pérez**[2], **Sara Lana-Serrano**[3] and **José C. González-Cristóbal**[3]

[1]Universidad Carlos III de Madrid, Spain
[2]Daedalus – Data, Decisions and Language, S.A., Spain
[3]Universidad Politécnica de Madrid, Spain
jvillena@it.uc3m.es, scollada@daedalus.es, slana@diatel.upm.es, josecarlos.gonzalez@upm.es

## Abstract

This paper discusses a novel hybrid approach for text categorization that combines a machine learning algorithm, which provides a base model trained with a labeled corpus, with a rule-based expert system, which is used to improve the results provided by the previous classifier, by filtering false positives and dealing with false negatives. The main advantage is that the system can be easily fine-tuned by adding specific rules for those noisy or conflicting categories that have not been successfully trained. We also describe an implementation based on k-Nearest Neighbor and a simple rule language to express lists of positive, negative and relevant (multiword) terms appearing in the input text. The system is evaluated in several scenarios, including the popular Reuters-21578 news corpus for comparison to other approaches, and categorization using IPTC metadata, EUROVOC thesaurus and others. Results show that this approach achieves a precision that is comparable to top ranked methods, with the added value that it does not require a demanding human expert workload to train.

## Introduction

In the recent years, the amount of information available on Internet regarding all fields of human knowledge is continuously growing at an exponential rate. One key for success is the ability to present the contents in a clearly organized, searchable and attractive way, providing added value such as links to related contents, information about involved entities or events, opinion on blogs or social networks, etc. In this context, complex information retrieval and categorization systems are required to store, process, filter and organize this massive volume of data, to turn it into useful information, and, eventually, to knowledge.

This paper focuses on methods for text categorization that tackle the problem of automatically organizing the information into meaningful sets. Specifically, a novel method for text categorization is presented: a hybrid approach combining a machine learning algorithm with a rule-based expert system that brings in the advantages of both approaches and overcomes their problems. We will fully discuss on its logical architecture and a possible implementation. Finally we will describe different scenarios on which the system has been successfully applied.



## Background

Automatic text categorization (or classification) is the task of automatically assigning one or several predefined category labels (or classes or topics) to a given text written in a natural language, according to its similarity with respect to a previously labeled corpus used as a reference set. The system can take hard decisions about the document-category pairs (a Boolean belongs/not_belongs decision) or else it can rank the categories for a given document based on some distance metric: the greater the *categorization status value* (CSV) of a category for a document, the better the document belongs to the category (Sebastiani 2002).

There are two traditional approaches for text categorization (Sebastiani 2002). On the one hand, a common approach in the early 80's involved humans in the creation of an expert system with manually defined rules, one per category, using logical expressions using terms in the text combined with AND, OR, NOT Boolean operators:

**if** (logical expression) **then** (category)

Such *knowledge engineering* approach provides rules as accurate as needed. The most famous example of this approach, the CONSTRUE system (Hayes et al. 1990), built by Carnegie Group for the Reuters news agency, was reported to achieve a breakeven value of 0.90. Furthermore, this approach has the additional benefit of being human understandable. However, certain expert knowledge about the domain is required, as well as specific knowledge concerning the details of the rule set as a whole, apart from the intrinsic difficulty to model a text category with a list of logical operations on terms. In any case, the main disadvantage is that the construction of the rule set when dealing

with many hundreds or even thousands of categories is an overwhelming task that puts this approach out of reach in most real-world scenarios.

On the other hand, the *machine learning* approach has become the predominant one since the 90's. In this case, the system is provided with a set of pre-classified (labeled) texts for each category, which is used as the training set, and automatically produces a classifier from them. The advantage is that the domain knowledge is only needed to assign a label to each existing text in the training set, which involves much lower workload than writing the rules.

Many different supervised learning algorithms and techniques have been used for building these classifiers, such as Naïve Bayes (Li and Yamanishi 1999), Linear Regression (Yang and Liu 1999), Nearest Neighbor (Joachims 1998), decision trees (Joachims 1998), artificial neural networks (Yang and Liu 1999), Support Vector Machines (Dumais et al. 1998), Learning Vector Quantization, Latent Semantic Indexing, Boosting (Weiss et at. 1999), genetic algorithms (Hirsch, Hirsch and Saeedi 2007), etc.

Each algorithm follows a different approach (statistical, probabilistic, sample-based, fuzzy logic, neural, etc.), but all of them are based on the fact that the more times a given term occurs in the text, the more relevant it is to the topic. Thus each text of the corpus can be mapped to a high dimensional feature vector $(w_{i1}\, w_{i2}\, \ldots w_{iN})$, where each entry $w_{ij}$ of the vector represents the degree in which the feature is present (or absent) in the text, modeled with a weight (a float value or 0/1 in the binary case).

Algebraically, the training set can be defined as the matrix containing the feature vectors for the M documents in the corpus, along with the CSV-values $(csv_{i1}\, csv_{i2}\, \ldots csv_{iK})$ for each topic to which each document is assigned (0 or 1 in the hard case, a float value in the fuzzy categorization).

$$Tr = \left(\begin{array}{c|c} w_{11}\, w_{12}\, \ldots w_{1N} & csv_{11}\, csv_{12}\, \ldots csv_{1K} \\ \ldots & \ldots \\ w_{M1}\, w_{M2}\, \ldots w_{MN} & csv_{M1}\, csv_{M2}\, \ldots csv_{MK} \end{array}\right) \quad (1)$$

Although the machine learning approach is proved to generate quite accurate classifiers, there are a number of drawbacks when compared to a rule-based system, mainly related to the fact that (in most cases) the model is not human understandable, thus it is hard to diagnose the reason for the false positives/negatives and fine-tune the system. In practical, the only way to improve the classifier is to invest more effort in the construction of the training set and test different alternatives to build the feature vectors.

## Machine-Learning Expert-System (MLES)

In this paper, we propose a novel hybrid text categorization method that combines a Machine Learning algorithm and a rule-based Expert System. The architecture of MLES approach is shown in Figure 1.

The first step is to train a base model from scratch by using available training data (labeled corpus). Any machine learning algorithm that can cope with the requirements of each scenario is suitable. However, in the general case, the classifier should be able to provide a multi-label classification and deal with hundreds or even thousands of classes, probably unbalanced (some classes having a few documents and some with many of them).

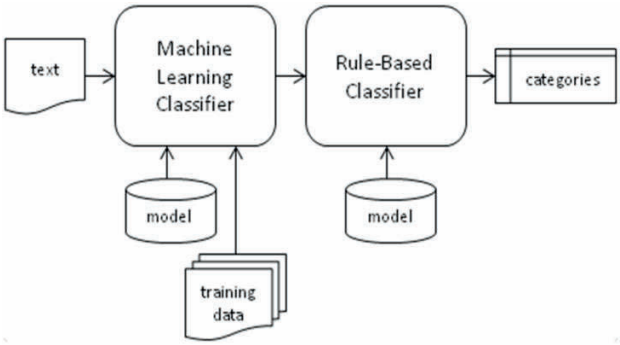


**Figure 1.** Logical architecture

Typically most classifiers are based on the feature vector for the given input text $q$, which is usually built using the Vector Space Model and any of the available weighting schemas (Salton, Wong and Yang 1975). To help the classifier, vector dimensionality may be reduced using any of the Feature Selection techniques that have been proposed in the information retrieval field.

The machine learning algorithm provides a categorization for this vector, by means of a list of classes $(c_i \in C)$, sorted by their CSV with respect to the input text $q$:

$$D_q = \begin{pmatrix} csv_{q,1} \\ \ldots \\ csv_{q,K} \end{pmatrix} \quad (2)$$

Afterwards, this base model may be fine-tuned by an expert system that uses "simple" rules based on logical expressions of natural language terms, as complex as necessary. These rules are used for post-processing the output of the machine learning classifier.

In general, each category can have none, one or several rules associated. Each rule is tested against the input text $q$ to accept (validate), reject (invalidate) or include such category, based on whether the input text satisfies or not the conditions expressed by the rule. Rejecting a category removes false positives returned by the machine learning classifier, thus precision improves. Including a new category solves any false negative, thus recall improves.

Moreover, rules are also used to rerank the list: rules increase (boost) the relevance of a given category, for instance, depending on the number of terms that satisfy the logical expression. This also improves the precision.

The output of this second block and thus the whole system is a list of classes with their relevance:

$$D'_q = D_q * B_q = \begin{pmatrix} csv'_{q,1} \\ \ldots \\ csv'_{q,K} \end{pmatrix} \quad (3)$$

Other works propose hybrid approaches for different categorization tasks, but they mainly combine two or several machine learning classifiers (Kim and Myoung 2003).

The benefits of MLES classifier are twofold, bringing in the advantages of both individual approaches and overcoming their problems. First, the classifier does not require a hard expert effort to train, as it initially only requires a training set with a set of documents belonging to each category. Thus there are considerable savings in terms of expert workload. This training corpus is frequently available in many scenarios, as in archives of news companies.

Second, the classifier is relatively easy to fine-tune, just by adding specific rules for the conflicting categories. The rule language in the expert system must allow writing complex logical expressions that in practical cover any kind of reasoning that a human expert may make about a certain category. Thus, its accuracy is the same as that achieved by human experts.

## Implementation

To evaluate our theoretical model, we chose and built a specific implementation of an actual system. In fact, such system was also designed to be used commercially in different scenarios, so our requirements demanded that the classifier should provide real-time categorization (response time lower than 50 ms in common hardware) and allow real-time category editing (deletion, creation, retraining).

Due to these needs but also because of its simplicity, we chose a k-Nearest Neighbor (kNN) algorithm based on the Euclidean distance for the machine learning classifier. Our implementation is based on Apache Lucene, an open source high-performance information retrieval engine. It uses the instance-based model, i.e., documents in the training corpus represent categories and are generated by aggregating all the texts belonging to that category. We did not opt for the example-based model, in which each document in the training corpus is an actual text belonging to a given category, in order to make the system faster and also more robust against the class imbalance problem.

Moreover, TF*IDF weighting was used for building the feature vectors, and feature selection is done by means of the simple Bag-Of-Words approach, selecting the N top weighted terms (200 terms, in most cases).

Regarding the expert system, in general, MLES supports any rule definition that can force certain constraints on the terms appearing or not in the text. However, for our implementation, a simplified rule language that makes the rule creation easier has been defined. For each i-th category, the rule includes three components:

- List of positive terms $P_i = \{p_{i1},\ p_{i2} \ldots\ p_{ip}\}$: at least one of these *p* terms must occur on the text:
  **if** ($p_{i1}\ OR\ p_{i2}\ OR \ldots OR\ p_{ip}$) **then**
  (category is accepted)
  **else**
  (category is rejected)
- List of negative terms $N_i = \{n_{i1},\ n_{i2} \ldots\ n_{in}\}$: none of these *n* terms must occur on the text:
  **if** ($n_{i1}\ OR\ n_{i2}\ OR \ldots OR\ n_{in}$) **then**
  (category is rejected)
  **else**
  (category is accepted)
- List of relevant terms $R_i = \{r_{i1},\ r_{i2} \ldots\ r_{ir}\}$: in this case, there is no acceptance or rejection, but the terms are used for boosting, as described below.

Based on those premises, the boosting factor for the category is defined as follows. Negative terms are used to reject the category and thus the final relevance is zero.

$$B_{q,i} = \begin{cases} 0, & if \exists t_i \in N \\ 1 + count(t_i \in P) + count(t_i \in R), & \text{otherwise} \end{cases} \quad (4)$$

Terms in rules may be either single words (for instance *fuel*) or multiword units (such as *gross domestic product*). In the latter case, the individual Boolean condition is true where all words are present in the input text:

$$t_i \equiv t_{i1}\ AND\ t_{i2}\ AND \ldots AND\ t_{ik} \quad (5)$$

Moreover, two extra rules are also defined to be used in case that no rule has been defined for a given category:

- ACCEPT: unconditionally accept the category, no matter which terms occur on the text ($B_{q,i} = 1$). This is actually the default rule.
- REJECT: always reject the category ($B_{q,i} = 0$).

# Evaluation

This system has been evaluated in different scenarios. The first one is just for comparison to other algorithms, using the Reuters-21578 well-known news test collection. The second scenario uses the IPTC model, also in the field of news categorization. Other scenarios include the use of EUROVOC thesaurus and categorization of medical text and video transcripts, as described next.

## Reuters-21578 model

The objective of this scenario is to establish a baseline experiment for evaluation and comparison to other methods, and prove that our proposal outperforms other algorithms or, in the worst case, gets similar results, with a much lower human workload.

The Reuters-21578 test collection (Sebastiani 2002) is a set of 21,578 news stories in English that appeared in the Reuters newswire in 1987, which has been widely adopted by the research community as a common benchmark for text categorization tasks throughout the last years. Although the collection covers 115 categories in all, the so-called R115 set, many researchers have focused on the R90 subset, which contains the 90 categories with at least one training example and one test example. Also, in order to make results comparable among systems, some standard splits (into a training and test set) have been widely used.

Table 1 shows the results reported by several groups using different approaches, taken from (Sebastiani 2002) and (Debole and Sebastiani 2004). All of them use the R90 subset and the popular ModApté split into 9,603 training documents and 3,299 test documents. Thus training and testing data available is the same for all methods. Micro-averaged breakeven point metric (where precision and recall are equal) is shown, as it was widely used in the past.

**Table 1. Results of other systems**

| Method | Reported by | BEP |
|---|---|---|
| Bayes | (Joachims 1998) | .720 |
| Bayes | (Li and Yamanishi 1999) | .773 |
| C4.5 | (Joachims 1998) | .794 |
| RIPPER (rules) | (Cohen and Singer 1999) | .820 |
| DL-ESC (rules) | (Li and Yamanishi 1999) | .820 |
| LLSF (regression) | (Yang and Liu 1999) | .849 |
| Widrow-Hoff | (Lam and Ho 1998) | .822 |
| Rocchio (batch) | (Joachims 1998) | .799 |
| Neural networks | (Yang and Liu 1999) | .838 |
| GisW (example) | (Lam and Ho 1998) | .860 |
| kNN | (Yang and Liu 1999) | .856 |
| SVMLight (SVM) | (Joachims 1998) | .864 |
| SVM | (Dumais et al. 1998) | .870 |
| AdaBoost.MH | (Weiss et at. 1999) | .878 |
| Bayesian net | (Dumais et al. 1998) | .800 |
| Genetic algorithms | (Hirsch and Saeedi 2007) | .800 |
| Average | | .824 |

Table 2 shows results from our own system in successive experiments. Run I uses kNN without any rule (i.e., all categories default to the ACCEPT rule), as a baseline experiment. As seen in the table, its performance is similar to the values reported in Table 1. Run II uses a set of simple rules written just for the top 10 categories (those with a higher number of examples in the training corpus) and run III uses similar simple rules for all the 90 categories in the training corpus. Finally, run IV uses specific "complex" rules, described later.

The improvement achieved by the rule-based boosting can be clearly noticed in the table. The final result outperforms all the listed methods but AdaBoost, with the clear advantage of a much lower effort to implement.

**Table 2. Results of our system**

| Run | Description | BEP |
|---|---|---|
| I | kNN only (no rules) | .817 |
| II | Rules for 10 top categories | .846 (+3.5%) |
| III | Rules for all categories | .858 (+5.0%) |
| IV | Complex rules | .877 (+7.3%) |

Table 3 shows the set of rules written for the 10 top categories in run II. These rules were manually extracted from inspection of the training corpus. They are really quite simple to generate, as they only include *relevant* terms that contribute to boost their category. No positive terms are used because most of those categories are about generic concepts such as company acquisitions or trade, which cannot be represented with specific words. Similar rules are written for all categories in run III.

In run IV, rules for categories that can be clearly expressed by specific words are rewritten using *positive* terms that force that those words are present in the text, for example: *wheat*, *corn OR maize*, *aluminum OR aluminium*, *sorghum*, *bop OR balance_of_payments*, etc.

**Table 3. Rule set for the top 10 categories in run II**

| Category | Relevant terms |
|---|---|
| acq | mergers\|merge\|acquisition\|acquisition\| share\|shares\|company\|companies |
| corn | corn\|maize |
| crude | crude\|oil\|barrel\|barrels\|petroleum |
| earn | earnings\|dividend\|dividends\|benefit\| benefits\|loss\|losses\|growth\|income\| incomes\|net\|company\|companies\| deficit\|deficits\|debt\|debts\|reduce\|increase |
| grain | grain\|grains\|crop |
| interest | interest\|interests\|rate\|rates\|prime\|discount |
| money-fx | money_exchange\|exchange\|exchanges\| change\|changes\|money\|value\|monetary\| currency\|currencies\|money_market |
| ship | shipping\|shippings\|ship\|waterway |
| trade | trade\|commerce\|deficit\|import\|imports\| export\|exports\|trade_deficit |
| wheat | wheat |

Table 4 shows the F1-measure for the top 10 categories achieved in the final run. Results are consistent with other experiments that report worse performance for *money* and *interest* categories, which appear to be more difficult to model, and best performance for *earn*.

**Table 4. Results of the 10 top categories in run IV**

| Category | F1 |
|---|---|
| acq | .891 |
| corn | .911 |
| crude | .924 |
| earn | .969 |
| grain | .879 |
| interest | .790 |
| money-fx | .740 |
| ship | .875 |
| trade | .879 |
| wheat | .805 |

## IPTC model

The system has also been trained for the International Press Telecommunication Council[1] (IPTC) hierarchy using news articles in Spanish. IPTC is an international consortium of the world's major news agencies, news publishers and news industry vendors. It develops and maintains technical standards for improved news exchange that are used by virtually every major news organization in the world. Among other activities, IPTC creates and maintains four sets of metadata attributes used to standardize the coding

[1] http://www.iptc.org/

of object metadata, known as *newscodes*: Descriptive, Administrative, Transmission and Exchange Format newscodes. They can be applied to texts, photographs, video and audio files, etc., and refer to different features such as gender, topic, format, scene in a picture, etc.

We have built a model for classifying the Descriptive newscodes, using its three hierarchical levels Subject, Matter and Detail. We used the February 2010 edition, which contains 1,349 categories.

For training the kNN classifier, we used a corpus of 108,838 news articles in Spanish, published in El País[2] during 2006 and 2007, tagged with the IPTC codes.

Rules were iteratively written and fine-tuned by a team of linguists over three weeks, guided by periodic performance evaluations carried out by an external team using actual news articles. The final version of the system, including rules for all 1,349 categories, comprising nearly 8,000 positive, negative and relevant terms, achieves the results shown in Table 5.

**Table 5. Results of our system**

| Parameter | Value |
|---|---|
| # Articles evaluated | 756 |
| Average # categories | 5.16 |
| Articles with all categories ok | 75.4% |
| Articles with all categories wrong | 0% |
| Articles with some categories ok | 100% |
| Articles with some categories wrong | 25% |
| Average precision of categories | 0.948 |

Some categories are easy to model with just a list of positive terms, for instance, articles classified into *15073046 (sport – sport meeting – Super Bowl)* must include *superbowl* or *super bowl* terms.

More general categories can be expressed with a list of relevant terms, such as in *01013000 (arts, culture and entertainment – photography),* which must include *photograph*, *photographs*, *photo*, *photos*, *photographer*, *photographers* and *photographed*.

In many other cases, negative terms must be included to differentiate among categories. For instance, *15039000 (sports – motor racing)* must include *motor racing*, *circuit* and *team*, but must not include *trucki*, *trucks*, *nascar* and *formula 1*. Obviously, category *15039007 (sports – motor racing – NASCAR)* must in turn include *nascar* and exclude *formula 1*, *trucki*, *trucks*, etc.

Currently this model is being used in several top news companies in Spain as their core content categorization system[3], in both supervised (editors choose the categories from the list proposed by the system) and unsupervised (completely automatic) tasks.

[2] El País (http://www.elpais.com) is the widest selling non-sports paper in Spain.

[3] Demo in http://showroom.daedalus.es/en/language-technologies/newscl/

## Other models

**EUROVOC thesaurus.** Another scenario is the categorization of legal documents appearing in public journals of different public bodies, according to the EUROVOC multilingual thesaurus used in the European Union[4]. The high number of categories (currently 6,797 categories) and the constant update makes the training and maintenance of the model a very challenging task. Rules for Spanish and Catalan have been developed, using the names and alias of the descriptors as positive terms (45,217 in all), and fine-tuning some frequent categories with negative terms (123 in all). Rules for the rest of the 22 official languages could be written in the same way with a reduced effort.

After an informal evaluation, the precision achieved for those languages is about 78%, just considering the first result returned by the system, and 84% with 5 results. These values are good enough in a supervised process.

**Medical text categorization**. A preliminary version of the method proposed in this paper was used to build MIDAS (Medical Diagnosis Assistant) system (Sotelsek-Margalef 2008), an advanced expert system able to suggest medical diagnosis (automating the assignment of ICD-9-CM codes) from the radiological and clinical patient records, based on machine learning from clinical histories of previously diagnosed patients. This system was specifically designed to participate in the 2007 Medical Natural Language Processing Challenge, achieving good precision rates.

**Video categorization**. Another application (Villena-Román 2009) was the topic categorization performed on dual language (English and Dutch) videos of television episodes using speech recognition transcripts and, optionally, metadata records (title and description). Results were promising, considering that it is still an ongoing research.

## Conclusions and Future Work

We have presented a hybrid approach for text categorization that combines a machine learning algorithm, which provides a base model that is relatively easy to train, with a rule-based expert system, which is used to post-process and improve the results provided by the previous classifier by filtering false positives and dealing with false negatives. We have also described a feasible implementation based on kNN and a simple rule language that allows to express lists of positive, negative and relevant (multiword) terms appearing in the text. Moreover we have described and evaluated the application of such system in different scenarios.

The main conclusion that can be drawn from the evaluation using Reuters-21578 is that MLES achieves a precision that is at least comparable to top ranked methods, with the added value that the model is built with a reduced human expert workload. If the output of the base classifier is satisfactory, there is no need to write a single rule. Howev-

[4] Demo in http://showroom.daedalus.es/en/language-technologies/eurovoc/

er, if any category turns out to be noisy and gets a low precision or recall, the system can be fine-tuned by adding specific rules for such categories. This is not feasible when using certain machine learning algorithms, where the only place for improvement is in the preparation of the training corpus or in the preprocessing of the input text.

We are currently researching on the extension of the rule language. Specifically we are studying the convenience of a fourth set of *irrelevant terms* in the rules used to decrease the CSV value of the category, but there are still open questions, for instance, regarding the boosting factor. We are also working on other implementations of the machine learning classifier, using Naïve Bayes or decision trees.

In addition, we are already working on the application of MLES to other scenarios such as foul language filtering, emerging trend detection and opinion mining (sentiment analysis). It also has the potential to be used in other disciplines such as social tagging or digital libraries, comparing human vs. computer generated tags (Heymann 2010). We believe that these are areas where the benefits of this approach may be of clear use.

## Acknowledgments

This work has been partially supported by the Spanish research projects: MA2VICMR: Improving the access, analysis and visibility of the multilingual and multimedia information in web for the Region of Madrid (S2009/TIC-1542), BRAVO: Advanced Multimodal and Multilingual Question Answering (TIN2007-67407-C03-01), MULTIMEDICA: Multilingual Information Extraction in Health domain and application to scientific and informative documents (TIN2010-20644-C03-01) and BUSCAMEDIA: Towards a semantic adaptation of multi-network-multiterminal digital media (CEN-20091026).

## References

Cohen, W. W. and Singer, Y. 1999. Context sensitive learning methods for text categorization. *ACM Transactions On Information Systems*. 17, 2, 141–173.

Debole, F., and Sebastiani, F. 2004. An Analysis of the Relative Hardness of Reuters-21578 Subsets. *Journal of the American Society for Information Science and Technology*, vol 56, pp 971–974.

Dumais, S. T., Platt, J., Heckerman, D., and Sahami, M. 1998. Inductive learning algorithms and representations for text categorization. In *Proceedings of CIKM-98, 7th ACM International Conference on Information and Knowledge Management* (Bethesda, MD, 1998), 148–155.

Hayes, P. J., Andersen, P. M., Nirenburg, I. B., and Schmandt, L. M. 1990. Tcs: a shell for content-based text categorization. In *Proceedings of CAIA-90, 6th IEEE Conference on Artificial Intelligence Applications* (Santa Barbara, CA, 1990), 320–326.

Heymann, P., Paepcke, A., Garcia-Molina, H. 2010. Tagging human knowledge. In *3rd ACM International Conference on Web Search and Data Mining* (WSDM), 51–60.

Hirsch, L., Hirsch, R., and Saeedi, M. 2007. Evolving Lucene search queries for text classification. In *Proceedings of the 9th annual conference on Genetic and Evolutionary Computation* (GECCO '07), pp 1604–1611.

Joachims, T. 1998. Text categorization with support vector machines: learning with many relevant features. In *Proceedings of ECML-98, 10th European Conference on Machine Learning* (Chemnitz, Germany, 1998), 137–142.

Kim, I.C., and Myoung, S. 2003. Text Categorization Using Hybrid Multiple Model Schemes. *Advances in Intelligent Data Analysis V*. Lecture Notes in Computer Science, 2003, Volume 2811/2003, 88–99.

Lam, W. and Ho, C. Y. 1998. Using a generalized instance set for automatic text categorization. In *Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval* (Melbourne, Australia, 1998), 81–89.

Li, H., and Yamanishi, K. 1999. Text classification using ESC-based stochastic decision lists. In *Proceedings of CIKM-99, 8th ACM International Conference on Information and Knowledge Management* (Kansas City, MO, 1999), 122–130.

Salton, G., Wong, A., and Yang, C.S. 1975. A vector space model for automatic indexing. *Communications of the ACM*, volume 18 num 11, pp 613–620.

Sebastiani, F. 2002. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1), pp 1–47.

Sotelsek-Margalef, A., and Villena-Román, Julio. MIDAS: An Information-Extraction Approach to Medical Text Classification. *XXIV Congreso de la Sociedad Española para el Procesamiento del Lenguaje Natural*, SEPLN08, Leganés, Spain, Septiembre 2008.

Villena-Román, J., and Lana-Serrano, S. MIRACLE at VideoCLEF 2008: Topic Identification and Keyframe Extraction in Dual Language Videos. *Evaluating Systems for Multilingual and Multimodal Information Access*. 9th Workshop of the Cross-Language Evaluation Forum, CLEF 2008, Aarhus, Denmark, 2008, Revised Selected Papers. Carol Peters et al (Eds.). Lecture Notes in Computer Science, Vol. 5706, 2009.

Weiss, S. M., Apté, C., Damerau, F. J., Johnson, D. E., Oles, F. J., Goetz, T., AND Hampp, T. 1999. Maximizing text-mining performance. *IEEE Intelligent Systems*. 14, 4, 63–69.

Yang, Y. and Liu, X. 1999. A re-examination of text categorization methods. In *Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval* (Berkeley, CA, 1999), 42–49.