# A robust model for intelligent text classification

Roberto Basili and Alessandro Moschitti
University of Rome Tor Vergata
Department of Computer Science, Systems and Production
00133 Roma (Italy)
{basili, moschitti}@info.uniroma2.it

## Abstract

*Methods for taking into account linguistic content into text retrieval are receiving a growing attention [16],[14]. Text categorization is an interesting area for evaluating and quantifying the impact of linguistic information. Works in text retrieval through Internet suggest that embedding linguistic information at a suitable level within traditional quantitative approaches (e.g. sense distinctions for query expansion as in [14]) is the crucial issue able to bring the experimental stage to operational results.*
*This kind of representational problem is also studied in this paper where traditional methods for statistical text categorization are augmented via a systematic use of linguistic information. Again, as in [14], the addition of NLP capabilities also suggested a different application of existing methods in revised forms. This paper presents an extension of the Rocchio formula [11] as a feature weighting and selection model used as a basis for multilingual Information Extraction. It allows an effective exploitation of the available linguistic information that better emphasizes this latter with significant both data compression and accuracy. The results is an original statistical classifier fed with linguistic (i.e. more complex) features and characterized by the novel feature selection and weighting model. It outperforms existing systems by keeping most of their interesting properties (i.e. easy implementation, low complexity and high scalability). Extensive tests of the model suggest its application as a viable and robust tool for large scale text classification and filtering, as well as a basic module for more complex scenarios.*

## 1 Introduction

Methods for taking into account linguistic content into text retrieval are receiving a growing attention [16],[14]. Text categorization is an interesting area for studying the impact of NLP information in retrieval processes. Works in text retrieval through Internet suggest that embedding linguistic information at a suitable level within traditional quantitative approaches (e.g. sense distinctions for query expansion as in [14]) is the crucial issue able to bring the experimental stage to operational results. This kind of representational problem is also studied in this paper where traditional methods for statistical text categorization are augmented via a systematic use of linguistic information.

The study of text classification ($TC$) is very useful to validate and measure the quality of the lexical methods (e.g. inductive methods over corpora or treebanks). Although text classification cannot objectively measure the relevance of linguistic information for every task, its benefits in $TC$ suggest also a positive impact in other $IR$ tasks. In $TC$ a systematic experimental framework is possible: tasks and performance factors, influenced by the availability of induced lexical information, can be assessed and measured over well-assessed benchmarking data sets.

In [3] an original extension of the well-know Rocchio model for feature weighting was proposed. The aim was to better assess the contribution of richer forms of feature representation on benchmarking data. Weighting was seen as a suitable method for measuring the effects of more informative features on the performance of the target classifier. Large scale experiments confirmed the need of tuning the Rocchio's formula parameters to training data. Sensitivity of the formula to different values of the parameters is also discussed in [7], where warnings on the estimation methodology are also raised.

The technique proposed in [3] is based on empirical parameter estimation aiming to optimize performances over an establish document set of documents. In [3] estimating over the test set itself was used to avoid noise in the model setting. The introduced bias (and its high performance) was thus adopted as an experimental framework to systematically measure the contribution of NLP. The result was that such a feature selection method was effective in emphasizing linguistic features like POS tagged lemmas, complex proper nouns and noun phrases, showing a sig-

nificant improvement with respect to poorer features (i.e. simple stems). The adopted estimation procedure was not generally assessed, as different test sets may lead to different parameter settings. In order to define a valid generalized Rocchio model, we have to show that parameters do not depend on document sets chosen for estimation but can be tuned via generally valid procedures.

In this paper, a parameter estimation procedure for the the extended Rocchio classifier is suggested and experimented. If an improvement similar to those suggested in [3] can be obtained this would assess the methodology as a novel approach to profile based classification. This would depend both on the availability of linguistic (i.e. more complex) information and on the better weighting and selection guaranteed by the proposed generalized formula. The resulting hybrid model would thus be assessed as a viable intelligent approach to $TC$ combining symbolic modeling, used in language processing and disambiguation, with a rather simple quantitative technique largely employed in operational systems.

In Section 2, the basic concepts about the problem issued in this paper will be introduced. The novel feature selection model with its weighting capabilities is presented in Section 3, where the suggested estimation procedure is also defined. In Section 4 experiments are reported aiming to show the effectiveness of the proposed estimation technique as well as to quantify the contribution of linguistic information.

## 2 Language-driven Text Classification

The classification problem is the derivation of a decision function $cat$ that maps documents ($d \in D$) into one or more classes, i.e. $cat : D \rightarrow 2^C$, where a set of classes, $C = \{C_1, ...., C_n\}$, represent topics and subtopics (e.g. "*Politics*"/"*Foreign Politics*") and an extensive collection of examples classified into them, often called *training set*, is available to derive $cat$.

*Profile-based* (or linear) classifiers are characterized by a function $cat$ based on a similarity measure between the representation of the incoming document $d$ and each class $C_i$. Both representations are vectors and similarity is traditionally estimated as the cosine angle between the two vectors. The description $\vec{C_i}$ of each target class ($C_i$) is usually called *profile*, that is the vector summarizing the content of all the training documents pre-categorized under $C_i$. The vector components are called *features* and refer to independent dimensions in the space in which similarity is estimated. The $i$-th components of a vector representing a given document $d$ is a numerical weight associated to the $i$-th feature $w$ of the dictionary that occurs in $d$. Similarly, profiles are derived from the grouping of positive instances $d$ in class $C_i$, i.e. $d \in C_i$.

Traditional techniques (e.g. [15]) make use of single words $w$ as basic features. The next section will describe the kind of linguistic information that extends class profiles and the processes used to obtain them.

### 2.1 Linguistic features in text categorization

Linguistic content in $TC$ can be represented by suitable $features$ able to express the needed evidence to the $cat$ function, i.e. selective information about training and test documents. Basic language processing capabilities traditionally allow to extend the knowledge about words occurring in documents, like for example their canonical forms (i.e. the morphological derivation from a lemma) and their syntactic roles (i.e. part-of-speech (POS) in the input context). Previous works on NLP-driven text classification (e.g. [3]) also suggest that availability of significant (or domain specific) multiwords improves performances. The recognition of Proper Nouns and terminological expressions provides effective information able focus on more selective feature sets.

The next section describe the nature of the linguistic information available from training data set and the processes used to derive them.

#### 2.1.1 The extraction of linguistic features

The $TC$ model that is proposed in this paper has been used within TREVI (Text Retrieval and Enrichment for Vital Information[1]), a system for Intelligent Text Retrieval and Enrichment. TREVI components are servers cooperating to the processing, extraction, classification, enrichment and delivery of news. Basically two TREVI components contribute to the $TC$ task:

- the $Parser$, i.e. a full linguistic preprocessor that take a normalized version of the news and produces a set of grammatical and semantic information for each text.
- a $Subject\ Identifier$, that according to the $Parser$ output and to the derived class profiles assigns one or more topics to each news. This is the proper $TC$ (sub)system.

The $Parser$ in TREVI is a complex (sub)system combining tokenization, lemmatization (via an independent lexical server), Part-of-Speech tagging [5] and robust parsing [4][2]. The information produced by the parser and used by the the $Subject\ Identifier$ component is the following:

[1] TREVI is a distributed object-oriented system, designed and developed within a European consortium under the TREVI ESPRIT project EP23311.

[2] Details on the linguistic methods and algorithms for each phase can be found in [4].

- Lemmas or multiwords expressions. Simple words (e.g. $bank$, $match$) as well as complex terminological expressions (e.g. noun phrases like "*bond issue*" or functional expressions as *in order to*) are detected and properly used during the later phases. Details on the extraction of relevant complex nominals acting as terminological expressions for the target categories is described in Section 2.1.2;

- Proper Nouns ($PN$s). In line with systems for Information Extraction, Named-Entities are recognized by extensive catalogs as well as by the application of NE grammars. A typed set of proper nouns is derived from each news and processed independently from the other lemmas.

- Syntactic Categories of lemmas. Units of text (i.e. simple or complex terms) are tagged by a single Part-of-Speech (POS), (e.g. `N` for nouns, `V` for verbs). Document descriptions include lemmas with their own POS, so that verbal and nominal occurrences are independent (e.g. $rate$/`V` $\neq$ $rate$/`N`)

- Major grammatical relations (i.e. `Subj/Obj` relations among words) are detected. News are thus annotated with basic syntactic structures emphasizing the roles of significant constituents (verbs and their modifiers).

The classification model that we propose is a profile-based classifier using as features the document's lemmas associated with their part-of-speech (POS) labels and the terminological expressions. Only nouns, verbs and adjectives are considered candidates features, and the resulting indexes are couples `<lemma, POStag>`. Proper Nouns (PNs) are also part of the profile[3]. Moreover, no stop list is used in TREVI, as POS tagging supplies the corresponding, and linguistically principled, filtering ability.

### 2.1.2 Corpus-driven terminology extraction

The noun phrase detection is supported by an inductive method for (off-line) terminology extraction early introduced in [2]. It is based on an integration of symbolic and statistical modeling. First, relevant atomic terms $ht$ (i.e. singleton words) are identified by traditional techniques, e.g. the $idf$ score early suggested in [15]. Linguistically principled grammars[4] are then applied to identify linguistic structures (headed by $ht$) as admissible candidates for terminological expressions. Finally, extracted candidates are validated and selected by the of use statistical filters. Statistical properties imposed on the occurrences of multiword sequences aim to restrict the semantic relations expressed by terms.

In terminology terms are surface canonical forms of structured expressions referring to entities with complex properties in a domain. They are nouns or noun phrases generally denoting specific concepts in a given corpus, i.e. in a given domain.

Usually term candidates are couples $(x, \vec{y})$, where $\vec{y}$ represents the sequence of (left and/or right) modifiers, e.g. *(disk, (-1,hard)), (system, ((-2,cable),(-1,television))* for *hard disk* and *cable television system*, respectively. Mutual information (MI), [10], has been often used to capture linguistic relations between words (e.g. [6, 8]):

$$I(x, y) = \log_2 \frac{P(x,y)}{P(x)P(y)}.$$

The stronger is the captured relation between $x$ and $y$ the larger is the joint with respect to marginal probabilities[5]. The basic problem is that MI (and its estimation) is concerned with only two events, and is better suited with bigrams, e.g. *hard disk*. Longer expressions usually require an iterative estimation as in [2, 9]. In [1] a different approach is proposed based on an extension of MI to collections of events (i.e. vector of words):

$$I(x, \vec{y}) = \log_2 \frac{P(x,\vec{y})}{P(x)P(\vec{y})}$$

where the conceptual link is considered between word $x$ and the vector $\vec{y} = (y_1, y_2, ..., y_n)$. The MI estimation $I(x, \vec{y})$ is obtained first by estimating each $i$-th component, $\hat{I}(x, y_i)$, then by graphical comparison among the obtained $\hat{I}(x, y_i)$. The obtained points define an histogram corresponding to a complex noun phrase. The study of the envelope by a shape factor allows to analyse the MIs of "multiple" modifiers. If a semantic relation holds between the modifiers $\vec{y}$ and the head $x$, than the obtained plot should be flat, i.e. no significant difference between the $I(x, y_i)$ values should be observed. In this way each candidate term $(x, \vec{y})$ is analysed looking "in parallel" to all its different MIs (i.e. $I(x, y_i) \quad \forall i$). Thresholding on the differences provides a straightforward and efficient decision criteria applied without iterating.

We processed the full training set available for a class $C_i$ an derived specific terminological datasets, $Term_i$. During preprocessing (i.e parsing), items in $\cup_i Term_i$ are thus

[3] Future work will include in the profi le also the available syntagmatic information, as its treatment requires a more complex description language and statistical modeling. No grammatic relation is thus considered in the feature set, although terminological structures brings information by hiding inner modifi ers and relations.

[4] A linguistic preprocessing supports tokenization, Part-of-Speech tagging and lemmatization for the grammatical recognition.

[5] A variety of estimations and extension of MI have been proposed, [6], like the following:

$$\hat{I}(x, y) = \log_2 N \frac{f_i(x, y)}{f(x)f(y)} \quad (1)$$

where $f_i(x, y)$ is the frequency of cooccurence of words $x$ and $y$ at distance $i$.

matched and represented as document features. Such complex noun phrases have been employed within the $TC$ experiments described in the Section 4.

## 3 Extending the Rocchio's formula for optimal feature selection and weighting

The poor improvements observed in NLP-driven IR tasks (e.g. [16]) usually depends on the noise introduced by the linguistic recognition errors or ambiguities (e.g. sense ambiguity in query expansion) which provides drawbacks comparable to the significant advantages. When more complex features (e.g. words and their POS tag or terminological units) are captured, it can be even more difficult to select the relevant ones among the set of all features. Data sparseness effects (e.g. the lower frequency of $n$-grams wrt simple words) interact with wrong recognitions (e.g. errors in POS assignment) and the overall information has a lower selectivity for the function $cat$.

The traditional solution is usually the *feature selection*, discussed for example in [18]. By applying statistical methods, (information gain, $\chi^2$, mutual information ...), the not relevant features are removed. Major drawbacks are that features irrelevant for a class may be removed even if they are important for another one. $Important$ but rare or specific $features$ may be cut in this way, as also noted in [13]. The crucial issue here is how to give the right weight to a given feature in different classes. This is even more important when NLP (and, especially, terminology recognition) is applied: some technical terms can be perfectly valid features for a class and, at the same time, totally irrelevant or misleading for others.

The Rocchio's formula has been traditionally used in order to build profiles associated to categories in Profile-Based Text Classifier. It is defined as follows. Given:

- the set of training documents $R_i$ classified under the topics $C_i$ (positive examples),
- the set $\bar{R}_i$ of the documents not belonging to $C_i$ (negative examples) and
- $\omega_f^h$, the weights[6] of feature $f$ in document $h$,

the weight $\Omega_f^h$ of a given feature $f$ in the profile of the class $C_i =< \Omega_f^1, \Omega_f^2, ... >$ is:

$$\Omega_f^i = \max\left\{0, \frac{\beta}{|R_i|}\sum_{h\in R_i}\omega_f^h - \frac{\gamma}{|\bar{R}_i|}\sum_{h\in \bar{R}_i}\omega_f^h\right\} \quad (2)$$

In Eq. 2 the parameters $\beta$ and $\gamma$ control the relative impact of positive and negative examples and determine the weight of $f$ in the $i$-th profile. In [11], Eq. (2) has been used with values $\beta$ = 16 and $\gamma$ = 4 as the task was categorization of low quality images.

The relevance of a feature deeply depends on the corpus characteristic and, in particular, on the differences among the training material for the different classes, e.g. size, the structure of topics or the style of documents. They sensibly change according to text collections and classes. The Equation 2 takes this into account setting to 0 features with a negative difference between positive and negative relevance. This aspect is crucial since the 0-valued features are irrelevant in the similarity estimation (i.e. they give a null contribution to the scalar product). This form of selection is rather smooth and allows to retain features that are selective only for some of the target classes. As a result, features are optimally used as they influence the similarity estimation for all and only the classes for which they are selective. The $\gamma$ and $\beta$ setting that optimizes the classification performance allows to drastically reduce noise without direct feature elimination.

At the same time Eq. 2 provides scores, $\Omega_f^i$, that can be directly used as weights in the associated feature space. Each category has in this way its own set of relevant and irrelevant features. It has been thus proposed in [3] that the optimal values of these two parameters can be obtained by estimating them independently for each class $i$. This results in a vector of ($\gamma_i$,$\beta_i$) couples each one optimizing the performance of the classifier over the $i$-th class. From now on we will refer to this model as the $Rocchio_{\gamma_i}$ classifier. Notice that the combined estimation of the two parameters is not required. For each class, one parameter ($\beta_i$=1) is fixed and $\gamma_i$ is tuned until the optimal performance is reached. The weighting, ranking and selection scheme used for $Rocchio_{\gamma_i}$ classifier is thus the following:

$$\Omega_f^i = \max\left\{0, \frac{1}{|R_i|}\sum_{h\in R_i}\omega_f^h - \frac{\gamma_i}{|\bar{R}_i|}\sum_{h\in \bar{R}_i}\omega_f^h\right\} \quad (3)$$

Equation 3 has been applied given the parameters $\gamma_i$ that for each class $C_i$ lead to the maximum breakeven point[7] of $C_i$.

### 3.1 Estimating parameters in a generalized Rocchio model

The idea of parameter adjustment in the Rocchio formula is not completely new. In [7] has been pointed out that these parameters greatly depend on the training corpus and different settings of their values produce a significant variation in performances. However their estimation was not clarified. The major problem was that the simple parameter estimation procedure that provides the lowest $training$ set error

[6] Several methods are used to assign weights of a feature, as widely discussed in [15].

[7] It is the threshold values for which precision and recall coincide (see [17] for more details).

produced a small improvement in the error rate over the reference test-set. The reason was that parameters for optimizing classification of training documents are very different from those optimizing the test-set classification. This did lead to the erroneous conclusion that the parameters are a property of the document set used for their derivation and so their use cannot increase general classification performances.

We are in agreement with the obtained results, but, as usually suggested, parameter estimation should never be carried out just on the set also used for training. The consequence can be a parameterization which depends heavily on the evidence extracted from training texts, that is too biased by this last information. Notice that an approach that takes a set of training documents for profile building and a second different subset, called the *estimation* set, for parameter estimation is more reasonable. First, the estimation is still carried out over data independent on the test set. Moreover, the obvious bias due to training material is avoided. If the estimated parameters converge to settings that have comparable (i.e optimal) performance also on the target test set we can conclude that:

- $\gamma_i$ values do not dependent on document sets but are tightly related to the categories $C_i$, and
- this procedure is general enough to be largely applied in operational scenarios of real AI applications.

More technically, the following parameter estimation procedure has been used. A benchmarking collection is usually made by a set of controlled, i.e. already categorized, documents. This set is then splitted into a first subset of training documents, called *learning* set $LS$, and a second subset of documents used to evaluate performance, called *test* set. This split can be fixed (as in the Reuters 3 collection [17]), or generated randomly from the collection. In statistical text categorization the learning set is traditionally used to extract features and build profiles.
As somewhere applied to statistical NLP, the parameter estimation for the Eq. 3 can be carried according to an held-out estimation procedure.

1. First, a subset of $LS$, called estimation set $ES$, is defined.
2. The set $LS - ES$ is then used for profile building
3. Estimation of the $\gamma_i$ parameters is finally carried out over $ES$.

Performance of the resulting model can be thus measured over the $TS$ documents. Notice that this procedure can be applied iteratively if steps 2-3 are carried out according to different, randomly generated splits $ES_k$ and $LS - ES_k$. Several vectors $\vec{\gamma}_i$ are thus derived at steps $k$, denoted by $\vec{\gamma}_i^{(k)}$. A resulting $\vec{\Gamma}_i$ can be thus obtained via a point wise estimator $\Theta$ applied to the $\vec{\gamma}_i^{(k)}$ distribution, i.e.

$$\vec{\Gamma}_i = \Theta(\vec{\gamma}_i^{(1)}, ..., \vec{\gamma}_i^{(K)}) \quad (4)$$

Performance of the model parameterized by $\Gamma_i$ can be then measured over the $TS$ documents.

The above procedure is easily applicable whenever the number of documents in the training set $LS$ is large enough for $ES$ (or $ES_k$) to be representative of all the classes. If the number of training documents available in $ES$ for a class $C_i$ is too low, the parameter estimation procedure that optimize BEP is not stable, possibly producing biased results. Unfortunately, a number of benchmarking collections are characterized by a poor balancing between the number of available training material for the target categories. This prevents the choice of smaller $ES$ sets, as they would not provide enough information for reliable parameter estimation: this can penalize the accuracy of the profile building phase. However, real operational scenarios (e.g. news agencies repositories, like the Reuters one used within the TREVI project) are less affected by these problems as larger data set can be made available.

It should be noticed that the use of just one parameter (i.e. $\gamma$) allows the estimation procedure to be easily implemented. Other models, as [7], use to select $\beta$ and $\gamma$ among a small set of values, empirically defined. The procedure presented above keeps $\beta_i$ fixed and allows to tune the negative contribution of other categories expressed by $\gamma_i$. In this way, the $\gamma_i$ estimation implements a pruning of features that are too frequent in other categories (singletons, or $n$-grams, assigned with a 0 weight). This naturally shrinks the range of parameter values (i.e. $\gamma_i$) to be tried.

If the suggested procedure provides an increase in performances with respect to the previous Rocchio-based models, several implications can be drawn:

- First, a systematic feature selection is available so that it can be used to emphasize the linguistic features in $TC$.
- The overall performances are in line with traditional benchmarking in the $TC$ area and can be thus used as a comparative result with respect to other models
- Finally, the relative low complexity of the overall model can be generalized to real (i.e operational) tasks in Information Filtering and Knowledge Management areas.

### 3.2 Related works

A probabilistic analysis of the Rocchio classifier algorithm has been carried out in [12], that discusses a version of the Rocchio formula using $TF \cdot IDF$ (product between

term frequency and inverse document frequency). A theoretical explanation of $TF \cdot IDF$ heuristic weighting (within a vector space model for text classification) is given. In this work the equivalence between the probability of a document $d$ in a category $C_i$ (i.e. $P(C_i|d)$) and the scalar product [8] $\vec{C_i} \cdot \vec{d}$ is discussed. This equivalence is shown to hold when $\gamma_i = 0$ and $\beta_i = \frac{|C_i|}{|D|}$, where $|D|$ is the number of corpus documents. The above theoretical interpretation (called $PrTFIDF$) is then used to justify a parameter setting in several experiments. Five categories of Reuters corpus have been used to measure performance and suggest an improvement with respect to a classic Rocchio classifier. The conclusion was that such a probabilistic model is preferable to the $TF \cdot IDF$ empirical weighting.

It should be noticed that an assumption at the basis of the above characterization is that the probability $P(d|w, C_i) = P(d|w)$ (where the word $w$ is a descriptor of $d$). This means that $P(C_i|d)$ is approximated by the expectation of $\sum_w P(C_i|w)P(w|d)$.

The assumption is critical. It assumes that sets of words are as informative as word sequences. This seems to suggest that $n$-grams are not useful in text classification. Previous systematic results pointed out that this is not true (e.g. [3]). In the rest of the paper we will experiment the proposed model that does not rely on the above hypothesis, not applicable to most operational scenarios of AI tools.

# 4 Performance Evaluation

The aim of the tests is to experimentally assess viability and effectiveness of the proposed generalized model (Eq. 3), and its estimation procedure. This enables a systematic evaluation of the overall performance supporting also contrastive analysis with previous statistical classifiers.

The next section will present the main aspects of the adopted benchmarking collection while experiments will be reported in Section 4.2.

## 4.1 The experimental set-up

As a reference collection the Reuters corpus, version 3, prepared by Apté [17] has been used. It will be hereafter referred as Reuters 3. The collection includes 11,099 documents for 93 classes, with a fixed splitting between test $TS$ and learning data $LS$ (3,309 vs. 7,789).

Performance scores are always expressed by means of *breakeven point* ($BEP$). When global performance values are reported, microaveraging among 93 classes is applied. The simple (i.e. non linguistic) feature set (named *Token Features*) includes unstemmed words that do not appear in the $SMART$ stop list. The linguistic feature set (*Linguistic Features*) is made of: POS-tagged lemmas, Proper Nouns and terminological expressions. These last are included in the linguistic feature set, as they have been acquired (according to the model described in Section 2.1.2). They are derived from training material available independently for each class. For example, in the dictionary of the class $acq$ (i.e. $Mergers\ and\ Acquisition$) among the 9,650 different features about 1,688 are made of terminological expressions or proper nouns ( 17%). The weight $\omega_f^h$ of a feature $f$ in a document $h$ is the usual product between the logarithm of the frequency of $f$ in $h$ and the associated inverse document frequency.

[8] Scalar product is used as a similarity measure between the document representation $\vec{d}$ and the profile $\vec{C_i}$

## 4.2 Evaluating the generalized Rocchio model

The first experiment is run to determine the overall performance of the model over the entire set of 93 classes. In this test the estimation procedure has not been run iteratively and only one split is applied where the set $ES$ is about 50% of the Reuters 3 training set $LS$. Moreover, in order to evaluate the impact of linguistic information, linguistic and standard features have been used. Other tests over the two feature sets have been also run by using, as a weighting model, the early Rocchio formula (i.e. *Std Rocchio* as in [11]) and the model characterized by $\gamma = \beta = 1$, [13].

The outcome of the experiments has been compared in Table 1. Last line reports the results of test performed by other authors, according to the same models but over standard feature sets (i.e. stemmed indexes filterd by stoplists).

**Table 1. Micro averaged Breakeven points of three Rocchio-based models on Reuters 3 (all 93 category of Apté split)**

| Feature Sets | $Rocchio_{\gamma_i}$ | ($\gamma = \beta = 1$) | *Std Rocchio* |
|---|---|---|---|
| Linguistic | **83.60%** | 80.75% | 78.90 % |
| Token | **82.15%** | 78.52% | 73.85 % |
| Literature | - | 79.9 % | 75% - 78% |

The generalized Rocchio classifier $Rocchio_{\gamma_i}$ outperforms as its overall performance is about 3% higher than the best results obtained by other models over linguistic data. The contribution of linguistic information and the selectivity provided by the estimation process seem to better capture training information. Notice that linguistic features (line 1) provide always the best results. This suggests that their information is relevant to the task.

Moreover, empirical settings of parameters (e.g. column 2 and 3) are still about 3% below the generalized Rocchio model, even when linguistic features are adopted. It seems that a suitable parameter setting for the $\gamma_i$ provides a systematic way to exploit the source linguistic information. It has to be observed that in NLP experiment we obtained a source set of 9,650 features for the Reuters 3 $acq$ category. After $\gamma_{acq}$ setting, only 4,957 features are assigned with a weight greater than 0. A data compression of about $\sim$ 51,3% is thus the overall effect of the feature weighting and selection.

However, all the tests suggests that linguistic features improve the behavior of these models with respect the experiments (line 3) previously reported. The use of NLP methods allows to include as features *n-gram*s not bound to a specific $n$. Terminological expressions may span over more than 2 or 3 constituents: complex proper nouns like *Federal Home Loan Bank* are usually captured. More interestingly, chains of noun phrases modifying other nouns or even proper nouns, as in *federal securities laws*, *temporary restraining order*, *Federal Home Loan Bank board* are recognized and normalized accordingly.

It must be said that the an *optimal* setting of the $\gamma_i$ parameters (i.e. the best performing) can be obtained by estimating them over the $TS$ (i.e. selecting those $\gamma_i$ that optimize BEP over the test set). When run in this way the $Rocchio_{\gamma_i}$ (as reported in [3]) has a BEP of 85.13%.

In order to explain the difference between these two outcomes, we designed a second experiment. in which parameters are estimated over the same $ES$ of the first one, but by carrying on the profile learning, the parameter setting and then measuring the performance only for a subset of the Reuters 3 categories. We included in this test only the 14 top-sized categories. For contrastive analysis, we also run the same experiment using the best setting of table 1, i.e. $\gamma = \beta = 1$. Both feature sets are then adopted.

Table 2 reports the results for the linguistic feature set, while Table 3 shows the outcome obtained by standard features (i.e. Token features). The bottom lines in the two tables report microaveraged results.

The best performance (86.79%) , obtained by applying the generalized Rocchio model over linguistic features, is higher than the *optimal* BEP previously obtained. This suggests that the estimation procedure, when fed with reliable material, is very effective. The categorization task over this subset of categories appears "easier", as the other model is also performing better. However, the influence of the $earn$ class (2035 documents in the test set) is a strong bias for the $\gamma = \beta = 1$ setting. Such a setting has been automatically estimated for the $Rocchio_{\gamma_i}$ model, while it is the outcome of repeated tests and is picked as the best setting (over the test set). In the $\gamma = \beta = 1$ model this setting is applied equivalently to all classes. As its is optimal for $earn$ loss of performances is not emerging. If $earn$ is removed from the 14 categories and $\gamma = \beta = 1$ is kept, the microaverage in column 2 is 77.33%, while 81.19% is the corresponding value in column 3 (about +4%). An improvement of about 5% (71.8% vs. 76.55%) is obtained by removing also the second largest class ($acq$). The systematic estimation proposed in Section 3.1 is thus more robust with respect to harder classification (sub)tasks, where differences in the weighting model ($\gamma_i$ in Eq 3) have to be captured.

**Table 2. Category performance with** *linguistic features*

| Category Name | $\gamma = \beta = 1$ | $Rocchio_{\gamma_i}$ |
|---|---|---|
| acq | 87.29 | 89.64 (+2.34) |
| corn | 66.66 | 84.72 (+18.05) |
| crude | 78.77 | 78.77 (0) |
| dlr | 62.71 | 64.40 (+1.69) |
| earn | 96.19 | 96.19 (+0) |
| grain | 78.57 | 85.71 (+7.14) |
| interest | 71.89 | 78.43 (+6.53) |
| money-fx | 61.71 | 63.06 (+1.35) |
| money-supply | 66.66 | 68.62 (+1.96) |
| oilseed | 50.00 | 67.74 (+17.74) |
| ship | 85.39 | 86.51 (+01.12) |
| sugar | 80.00 | 80.00 (0) |
| trade | 75.16 | 75.16 (0) |
| wheat | 72.41 | 85.05 (+12.64) |
| MicroAverage | 84.38 | **86.79** (+2.41) |

The nature of the benchmarking collection unfortunately prevents a full assessment of the best performances reachable with the generalized Rocchio model, at least for comparative purposes.

One noticeable observation is that when linguistic features are not used the gap between the BEP of previous models (81.50%) and the $Rocchio_{\gamma_i}$ (85.82%) (column 2 vs. column 3 in Table 3) increases, as the effects of the parameter estimation is stronger. A lower gap is shown in Table 2: it seems that part of the selective information is better captured by the linguistic feature set.

The comparison of column 2 in Table 3 and column 3 in Table 2 suggests that the combined use of linguistic information and the generalized Rocchio formula (Eq. 3) provides an overall increase of more than 5%.

## 5 Conclusion

In this paper, a robust model for NLP-driven text categorization and its training procedure have been described. Systematic experiments shows the superiority of the method with respect to previously reported results.

**Table 3. Category performance with** *token features*

| Category Name | $\gamma = \beta = 1$ | $Rocchio_{\gamma_i}$ |
|---|---|---|
| acq | 85.02 | 85.83 (+0.81) |
| corn | 73.26 | 86.30 (+13.04) |
| crude | 80.76 | 85.63 (+4.87) |
| dlr | 15.94 | 63.33 (+47.39) |
| earn | 95.49 | 96.37 (+0.88) |
| grain | 76.61 | 84.23 (+7.62) |
| interest | 60.09 | 70.12 (+10.03) |
| money-fx | 57.62 | 67.69 (+10.07) |
| money-supply | 37.87 | 63.46 (+25.59) |
| oilseed | 56.41 | 71.42 (+15.01) |
| ship | 79.09 | 87.91 (+8.82) |
| sugar | 87.50 | 94.00 (+6.50) |
| trade | 68.33 | 60.92 (-7.41) |
| wheat | 73.72 | 85.22 (+11.50) |
| Micro Average | 81.50% | **85.82%** (+4.32) |

The proposed estimation procedure seems able to better emphasize the contribution of NLP-driven preprocessing to the text classification task. From one side, it assesses the role of linguistic features in the $TC$ area of IR. Efficient extraction/matching of linguistically motivated complex features (including multi-word patterns) as well as proper noun detection are able to produce selective information. On the other side, the proposed weighting method (Eq. 3) and the corresponding estimation procedure define a systematic feature selection technique robust and effective with respect to noise and ambiguity in the data.

An improvement similar to those suggested in [3] has been obtained. This depend both on the availability of linguistic (i.e. more complex) information and on the better weighting and selection guaranteed by the proposed generalized formula. The resulting $Rocchio_{\gamma_i}$ model applied to linguistic material supports a computationally efficient classification (typical of purely statistical models) and produces performances close to the best (but computationally more expensive) classifiers (e.g. KNN and SVM). In the current phase this model is adopted as a filtering subsystem in a large project (NAMIC, LE n. 12391) for hypertextual authoring of news streams. First news classification is applied and then a domain-specific information extraction is carried out. Different ontologies can thus be applied according to the categorization results (e.g. finance and sport are the two target domains). The overall system is a complex Information Retrieval and Extraction platform, combining benefits of quantitative (i.e. statistical) and symbolic (i.e. knowledge-based IE) models. The evaluation of the classification component on real operational scenarios (i.e. news agencies on-line services) will be part of the medium term research on the model proposed in this paper.

## References


[1] R. Basili, A. Bonelli, and M. T. Pazienza. Estrazione e rappresentazione di informazioni terminologiche eterogenee. In *AI*IA '98 - VI Convegno*, 1998.

[2] R. Basili, G. De Rossi, and M. Pazienza. Inducing terminology for lexical acquisition. In *Preoceeding of EMNLP 97 Conference, Providence, USA*, 1997.

[3] R. Basili, A. Moschitti, and M. Pazienza. NLP-driven IR: Evaluating performances over text classifi cation task. In *Proceedings of IJCAI 2001 Conference, Seattle, USA*, 2001.

[4] R. Basili, M. T. Pazienza, and F. M. Zanzotto. Effi cient parsing for information extraction. In *Proc. of the ECAI98*, Brighton, UK, 1998.

[5] E. Brill. A simple rule-based part of speech tagger. In *Proc. of the Third Applied Natural Language Processing, Povo, Trento, Italy*, 1992.

[6] K. W. Church and P. Hanks. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1), 1990.

[7] W. W. Cohen and Y. Singer. Context-sensitive learning methods for text categorization. In *Proceedings of SIGIR 96' Conference*, pages 12–20, 1996.

[8] I. Dagan, S. Marcus, and S. Markovitch. Contextual word similarity and estimation from sparse data. In *COLING-94*, 1994.

[9] B. Daille. Study and implementation of combined techniques for automatic extraction of terminology. In *The Balancing Act: Combining Symbolic and Statistical Approaches to Language, WorkShop of the ACL*, 1994.

[10] R. Fano. *Transmission of information*. MIT Press, Cambridge, 1961.

[11] D. J. Ittner, D. D. Lewis, and D. D. Ahn. Text categorization of low quality images. In *Proceedings of SDAIR-95*, pages 301–315, Las Vegas, US, 1995.

[12] T. Joachims. A probabilistic analysis of the rocchio algorithm with tfi df for text categorization. In *Proceedings of ICML97 Conference*. Morgan Kaufmann, 1997.

[13] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *In Proceedings of ECML-98*, pages 137–142, 1998.

[14] D. I. Moldovan and R. Mihalcea. Using wordnet and lexical operators to improve internet searches. *IEEE Internet Computing*, January-February, 2000.

[15] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[16] E. Voorhees. Query Expansion Using Lexical-Semantic Relations. In *Proceedings of the Seventeenth Annual International ACM-SIGIR Conference*, Dublin, Ireland, July 1994.

[17] Y. Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval Journal*, 1999.

[18] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of ICML-97*, pages 412–420, Nashville, US, 1997.