NLP-driven IR: Evaluating Performances over a Text Classification task

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

Abstract

Although several attempts have been made to introduce Natural Language Processing (NLP) techniques in Information Retrieval, most ones failed to prove their effectiveness in increasing performances. In this paper Text Classification (TC) has been taken as the IR task and the effect of linguistic capabilities of the underlying system have been studied. A novel model for TC, extending a well know statistical model (i.e. Rocchio's formula [Ittner *et al.*, 1995]) and applied to linguistic features has been defined and experimented. The proposed model represents an effective feature selection methodology. All the experiments result in a significant improvement with respect to other purely statistical methods (e.g. [Yang, 1999]), thus stressing the relevance of the available linguistic information. Moreover, the derived classifier reachs the performance (about 85%) of the best known models (i.e. Support Vector Machines (SVM) and K-Nearest Neighbour (KNN)) characterized by an higher computational complexity for training and processing.

1 Introduction

Although in literature poor evidence assessing the relevance of Natural Language Processing (NLP) in improving Information Retrieval (IR) has been derived, a shared belief exists that linguistic processing can capture critical semantic aspects of document content that simple word matching cannot do. It has been also stressed (e.g. [Grefenstette, 1997]), that vector space models are inadequate to deal with retrieval from Web via commonly available simple and short queries. Language processing enables to enrich the document representation with semantic structures although the nature and methods for doing this are still under debate. Which specific information (structures and dependencies) can be suitably derived and which combined use of linguistic processes and IR models are to be applied represent still open questions.

In order to derive more insight on the above issues, a systematic experimental framework has to be defined, where tasks and performance factors can be assessed and measured. Among other IR tasks, Text Classification (TC) is a promising process for our objectives. It plays a major role in retrieval/filtering processes. Moreover, given the rich experimental evidence on well-assessed benchmarking collections, TC better supports a comparative evaluation of the impact of linguistic information with respect to approaches based on word matching.

The classification problem is traditionally described as follows: given a set of classes ($C = \{C_1, ..., C_n\}$, i.e. topics/subtopics labels, e.g. "*Politics*"/"*Foreign Politics*") and an extensive collection of examples classified into these classes, often called *training set*, the classification problem is the derivation of a decision function f that maps documents ($d \in D$) into one or more classes, i.e. $f : D \to 2^C$. As the specific topics (classes) are fixed, the extraction of $relevant$ content from document (for retrieval purposes) can be more systematic (given their focused semantics) and less complex than in other IR scenarios. Therefore TC represents a suitable environment for testing the capabilities of NLP to capture such semantic aspects.

The role of linguistic content in TC relates to the definition of $features$ able to provide specific and selective information about training and test documents and (consequently) about the target classes. Basic language processing capabilities allow to extend the knowledge on the words occurring in documents, e.g. their canonical form (i.e. the morphological derivation from a lemma) and their syntactic role (i.e. their part-of-speech (POS) in the input context). Previous works on NLP-driven text classification (e.g. [Basili *et al.*, 2000b]) suggest that such information improves performances. In particular, lemmatization and recognition (i.e. removal of Proper Nouns from the set of selective feature) provide a linguistically principled way to compress the features set (usually obtained by traditional crude methods like stop lists or statistical thresholds, e.g. χ^2). Statistical unsupervised terminological extraction has been also applied to TC training. It allows detecting more complex and relevant features, i.e. complex nominal groups typical of the different topics. The results are improved TC performances, although the contribution given by such modules has not yet been accurately measured.

The main reason for poor improvements (if any) when NLP is applied to IR is the noise introduced by the linguistic recognition errors which provides drawbacks comparable to the significant advantages. In the specific case of TC, 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 about a class looses its potential selectivity.

The traditional solution is usually the *feature selection*, discussed for example in [Yang and Pedersen, 1997]. By applying statistical methods, (information gain, χ^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 [Joachims, 1998]. 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.

In this paper an original TC model for selection and weighting of linguistically motivated features, as an extension of the the Rocchio classifier ([Ittner *et al.*, 1995]), has been designed and implemented. It has been experimented on feature sets extracted by NLP techniques: terminological expressions, part-of-speech tagged lemmas and proper nouns.

In Section 2 the novel feature selection model with its weighting capabilities is presented. Section 3 describes the NLP functionalities adopted for extracting the feature sets from the target documents during training and testing. In Section 4 the experiments aiming to measure the impact of the feature selection on the classification performances as well as of the contribution of linguistic information are described.

2 Text Classification

Two main approaches to the construction of a non-parametric classifier have been proposed and experimented in literature [Lewis *et al.*, 1996].

Profile-based (or linear) classifiers are characterized by a function f that is 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. Traditional techniques (e.g. [Salton and Buckley, 1988; Salton, 1991]) make use of single words w as basic features. 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. $d \in C_i$.

Example-based are other types of classifiers, in which the incoming document d is used as a query against the training data (i.e. the set of training documents). Similarity between d and class is evaluated as cumulative estimation between the input document and a portion of the training documents belonging to that class. The categories under which the training documents with the highest similarity are categorized, are considered as promising classification candidates for d. This approach is also referred as *document-centered* categorization. For both the above models a document is considered valid for a given class *iff* the similarity estimation overcomes established thresholds. The latter are parameters that adjust the trade-off between precision and recall.

2.1 The Problem of Feature Selection

Feature Selection techniques have been early introduced in order to limit the dimensionality of the feature space of text categorization problems. The native feature space consists of the unique terms (words or phrases) that occur in documents, which can be hundreds of thousands of terms even for a small text collection. This size prevents the applicability of many learning algorithms. Few neural models, for example, can handle such a large number of features usually mapped into input nodes.

Automatic feature selection methods foresee the removal of noninformative terms according to corpus statistics, and the construction of new (i.e. reduced or remapped) feature set. Common statistics parameters are: the *information gain* (e.g. [Yang and Pedersen, 1997]) aggressively reduces the document vocabulary, according to a naive Bayes model; a decision tree approach to select the most promising features wrt to a binary classification task; mutual information and a χ^2 statistic have been used to select features for input to neural networks; document clustering techniques estimating probabilistic ”term strength”; inductive learning algorithms that derive features in disjunctive normal form.

As pointed out in [Yang and Pedersen, 1997] document frequency (DF), χ^2 and information gain provide the best selectors able to reduce the feature set cardinality and produce an increment of text classifier performances. The following equations describes four selectors among those experimented in [Yang and Pedersen, 1997]. They are based on both mutual information and χ^2 statistics:

$$I_{max}(f) = \max_i\big\{I(f, C_i)\big\},\ I_{avg}(f) = \sum_i P_r(C_i) \cdot I(f, C_i)$$

$$\chi^2_{max}(f) = \max_i\big\{\chi^2(f, C_i)\big\},\ \chi^2_{avg}(f) = \sum_i P_r(C_i) \cdot \chi^2(f, C_i)$$

where

- $P_r(C_i)$ is the probability of a generic document belonging to a class C_i, as observed in the training corpus
- f is a generic feature
- $I(f, C_i)$ is the mutual information between f and C_i,
- $\chi^2(f, C_i)$ is the χ^2 value between f and C_i

After the ranking is derived, selection is carried out by removing the features characterized by the lowest scores (thresholding). Each of the above models produces a ranking of the different features f that is the same for all the classes: each of the above formulas suggests only one weight depending on all the classes. For example, the selector of a feature by

I_{avg} applies the average function to the set of $I(f, C_i)$ scores: every dependence on the i-th class disappear resulting in one single ranking. The same is true for χ^2_{max} and χ^2_{avg}.

Notice that this ranking, uniform throughout categories, may select features which are non globally informative but are enough relevant only for a given (or few) class(es) (e.g. the max or avg). The selection cannot take into account differences in the relevance among classes. Classes that are more generic, e.g. whose values of $I(f, C_i)$ (or χ^2) tend to be low, may result in a very poor profile, i.e. fewer number of selected features. This is in line with the observation in [Joachims, 1998] where the removal of features is suggested as a loss of important information, as the number of truly irrelevant features is negligible. Moreover, functions like avg are even more penalizing as they flatten the relevance of a single feature for each class to an ”ideal” average value. Notice that this weakness is also reflected by the poorer results reported in [Yang and Pedersen, 1997].

In order to account for differences in the distribution of relevance throughout classes, we should depart from the idea of a uniform ranking. Features should be selected with respect to a single category. This can lead to retain features only when they are truly informative for some classes. Moreover a suitable class-based ranking is obtained, so that the feature scores (e.g. the mutual information $I(f, C_i)$) can be straightforwardly assumed as weights for the features in class C_i.

In next section an extension of the Rocchio formula aiming to obtain such desirable feature weights is presented.

2.2 Rocchio classifiers

The Rocchio classifier is a profile based classifier, presented in [Ittner *et al.*, 1995], which uses the Rocchio's formula for building class profiles. Given the set of training documents R_i classified under the topics C_i, the set $\bar{R}_i$ of the documents not belonging to C_i, and given a document h and a feature f, the weight Ω^h_f of f in the profile of C_i is:

$$\Omega^i_f = \max\left\{0, \frac{\beta}{|R_i|}\sum_{d_h \in R_i}\omega^h_f - \frac{\gamma}{|\bar{R}_i|}\sum_{d_h \in \bar{R}_i}\omega^h_f\right\} \quad (1)$$

where ω^h_f represent the weights of features in documents[1]. In Eq. 1 the parameters β and γ control the relative impact of positive and negative examples and determine the weight of f in the i-th profile. In [Ittner *et al.*, 1995], (1) has been firstly used with values β = 16 and γ = 4: the task was categorisation of low quality images. The success of these values possibly led to a wrong reuse of them in other fields [Cohen and Singer, 1996].

These parameters indeed greatly depend on the training corpus and different settings of their values produce a significant variation in performances. Poor performances have been obtained indeed in [Yang, 1999], and a wrong γ and β setting (maybe the orginal Ittner one) is a possible explanation. In [Joachims, 1998], the trial with a small set of values for β ($\{0, 0.1, 0.25, 0.5, 1.0\}$) is carried out and increased performance wrt those previously assessed by other authors are obtained. However, the corresponding γ values are not mentioned. The impact of the adjustment of γ and β is significant if optimal values are systematically estimated from the training corpus. Experimental evidence will be further shown in Section 4.1.

Tuning Rocchio's formula parameters

As previously discussed, the Rocchio classifier strongly relies on the γ and β setting. However, 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, i.e. size, the structure of topics, the style of documents, ... This varies very much across text collections and across the different classes within the same collection.

Notice that, in Equation 1, features with negative difference between positive and negative relevance are set to 0. This implies a discontinuous behavior of the Ω^i_f values around the 0. 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. In this way, the minimal set of truly irrelevant features (giving 0 values for all the classes) can be better captured and removed, in line with [Joachims, 1998].

Moreover, the γ and β setting that is fitted with respect to the classification performance has two main objectives:

- First, noise is drastically reduced by the Rocchio formula smoothing and without direct feature deletion.
- Second, the resulting ranking provides Rocchio-based scores that can be directly used as weights in the associated feature space. The higher is the positive evidence (wrt to the negative one) the higher is the relevance and this may vary for each target class.

Notice now that each category has its own set of relevant and irrelevant features and Eq. 1 depends for each class i on γ and β. Now if we assume the optimal values of these two parameters can be obtained by estimating their impact on the classification performance, nothing prevents us from driving this estimation independently for each class i. This will result in a vector of (γ_i,β_i) couples each one optimizing the performance of the classifier over the i-th class. Hereafter we will refer to this model as the $Rocchio_{\gamma_i}$ classifier.

Notice that the proposed approach could converge to the traditional Rocchio weighting *if and only if* a single optimal value for γ and β is obtained, i.e. $\forall i \gamma_i = \gamma$ and $\beta_i = \beta$. This has not been the case as in Section 4.2 will be shown.

Finally, it has to be noticed that combined estimation of the two parameters is not required. For each class, we fixed one parameter (β_i indeed) and let γ_i vary until the optimal performance is reached. The weighting, ranking and selection scheme used in the for $Rocchio_{\gamma_i}$ classifier is thus the following:

$$\Omega^i_f = \max\left\{0, \frac{1}{|R_i|}\sum_{d_h \in R_i}\omega^h_f - \frac{\gamma_i}{|\bar{R}_i|}\sum_{d_h \in \bar{R}_i}\omega^h_f\right\} \quad (2)$$

[1] Several methods are used to assign weights of a feature, as widely discussed in [Salton and Buckley, 1988]

In our experiments, β has been set to 1, Equation 2 has been applied given the parameters γ_i that for each class C_i lead to the maximum breakeven point[2] of C_i.

3 The role of NLP in feature extraction

One of the aim of this work was to emphasize the role of linguistic information in the description (i.e. feature extraction) of different classes in a TC task. It is to be noticed that these latter are often characterized by sets of *typical* concepts usually expressed by multi-words expressions, i.e. linguistic structures synthesizing widely accepted definitions (e.g. "*bond issues*" in topics like "*Finance* or *Stock Exchange*"). These sets provide useful information to capture semantic aspects of a *topics*. The multi-word expressions are at least in two general classes useful for TC:

- Proper Nouns (PN), which usually do not bring much selective information in TC. Most named entities are locations, persons or artifacts and are rarely related to the semantics of a class. An evidence of this is discussed in [Basili *et al.*, 2000b] where PN removal is shown to improve performances.
- Terminological expressions, i.e. lemmatized phrase structures or single terms. Their detection results in a more precise set of features to be included in the target vector space.

The identification of linguistically motivated terminological structures usually requires external resources (thesaura or glossaries): as extensive repositories are costly to be developed and simply missing in most domains, an enumerative approach cannot be fully applied. Automatic methods for the derivation of terminological information from texts can thus play a key role in content sensitive text classification.

As terms embody domain specific knowledge we expect that their derivation from a specialized corpus can support the matching of features useful for text classification. Once terms specific to a given topics C_i are available (and they can be estimated from the training material for C_i), their matching in future texts d should strongly suggest classification of d in C_i.

Several methods for corpus-driven terminology extraction have been proposed (e.g. [Daille, 1994; Arppe, 1995; Basili *et al.*, 1997]). In this work, the terminology extractor described in [Basili *et al.*, 1997] has been adopted in the training phase. Each class (considered as a separate corpus) gives rise to a set of terms, T_i. When available, lemmatized phrase structures or sinle lemmas in T_i can be matched in future test documents. They are thus included in the final set of features of the target classifier.

Other features provided by linguistic processing capabilities are lemmas and their associated POS information able to capture word syntactic roles (e.g. *adjective*, *verb*, *noun*)[3].

A further novel aspect of the classifier proposed in this paper is the application of the Equation 2 as a weighting scheme of the linguistically derived features. This allows to better separate the relevant information from the irrelevant one (possibly introduced by errors in the linguistic processing, e.g. wrong POS assignment). Finally, those irrelevant features, that are not necessarily produced via complex linguistic processing (e.g. single words), are correctly smoothed by Eq. 2 and this also helps in a more precise measurement of the NLP contribution.

4 Experimenting NLP-driven classification

The experiments have been carried out in two phases. First, we experimented Rocchio classifiers over a standard feature set, i.e. simple words. This serves two purposes. First, it provides the evaluation of the Breakeven Point reachable by a Rocchio classifier (via estimation of the suitable, but global, γ parameter). This allows a direct and consistent comparisons with the Rocchio models proposed in literature. Furthermore the first expriment also suggests the parameter setting that provides the best breakeven point of the extended $Rocchio_{\gamma_i}$ model. In a second phase this optimal $Rocchio_{\gamma_i}$ model has been experimented by feeding it with linguistic features. Comparative analysis of the two outcomes provides evidence of the role of this augmented information.

As a reference collection the Reuters, version 3, corpus prepared by Apté [Yang, 1999; Apté *et al.*, 1994] has been used. It will be hereafter referred as Reuters3. It includes 11,099 documents for 93 classes, with a fixed splitting between test and learning data (3,309 vs. 7,789). Every experiment thus allows direct comparisons with others models described in Section 5.

4.1 Deriving a baseline classifier

In these experiments, only words are taken as features and no other NLP facility has been applied in agreement with other methods described in literature. The feature weight in a document is the usual product between the logarithm of the feature frequency (inside the document) and the associated inverse document frequency. The best Rocchio classifier performances has been derived by systematically setting different values of γ and optimizing performance (i.e. the breakeven point, BEP). A sequence of classifiers has thus been obtained. In Figure 1 the plot of BEPs with respect to γ is shown. The two plots refer to two different feature sets: *SimpleFeatures* refers to single words while *LingFeatures* refers to the model using all the available linguistic information (see Sect. 4.2). First of all, performances depend strongly on the parameters. The best performance of the *SimpleFeatures* model is reached with γ = 6 (and β=1): these values significantly differ from the γ=16 and β=4 used elsewhere. Second, significant higher performances characterize the language-driven model for all the γ values: this shows an inherent superiority of the source linguistic information. However, when a single γ for all the classes is used, a suitable adjustment let the *SimpleFeatures* model to approximate the behaviour of the linguistic one. This is no longer true when more selective estimation of the parameter (i.e. γ_i $\forall i$) is applied.

[2] It is the threshold values for which precision and recall coincide (see [Yang, 1999] for more details).

[3] These features have been built using the linguistic engine of the Trevi system, [Basili *et al.*, 2000a], where the interested reader can also find more details on the linguistic processing.

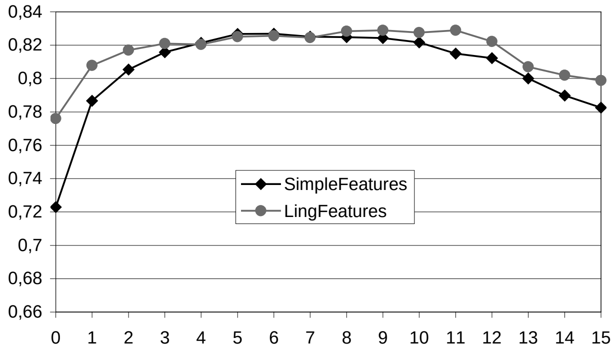

Figure 1: Break-even point performances of the Rocchio classifier according to different γ values

In a second experiment indeed the parameter estimation process has been individually applied to each class i, and the optimal sequence of γ_i values has been obtained. In the Table 1 are shown the performances of three Rocchio classifiers: simple Rocchio (as in [Ittner *et al.*, 1995], with $\gamma = 4/16$), as well as *Rocchio* $\gamma = 6$ and $Rocchio_{\gamma_i}$ characterized by one global parameter and individual parameters, respectively. The three tests have been carried out by using only simple words as features, in line with traditional techniques. Notice that both the optimized *Rocchio* $\gamma = 6$ and $Rocchio_{\gamma_i}$ models, proposed in this paper, outperform all the best results obtained in literature for Rocchio classifiers (e.g. [Joachims, 1998; Cohen and Singer, 1996]).

Table 1: Breakeven points of three Rocchio-based models on Reuters3

$Rocchio_{\gamma_i}$	*Rocchio* $\gamma = 6$	Rocchio
83.82%	82.61%	75% - 79.9%

4.2 Comparing different NLP-based classifiers

Once the best weighting technique has been assessed as the optimal estimation of parameters γ_i in the previous experiments, it is possible to selectively measure the contribution given by NLP. In fact an independent baseline model, with minimal noisy information, (i.e. the $Rocchio_{\gamma_i}$ model in Table 1), is used contrastively to correctly measure the contribution brought by the augmented features. In the following experiment, the novel sets of features described in Section 3 have been added to the standard set. They consist of:

- Proper Nouns (+PN or -PN if recognition in text is followed by removal during TC training)
- Terminological Expressions (+TE)
- Lemmas (-POS)
- Lemmas augmented with their POS tags in context (+POS)

Terminological expressions have been firstly derived from the training material of one class: for example, in the class *acq* (i.e. $Mergers\ and\ Acquisition$) of the Reuters3, among the 9,650 different features about 1,688 are represented by terminological expressions or complex Proper Nouns (17%). The $Rocchio_{\gamma_i}$ model has been selectively applied to three linguistic features set: the first includes only the lemmas associated to the POS tag (+POS), the second lemmas only (-POS), and the third Proper Nouns and Terminological Expressions (+POS+PN+TE).

In Table 2 is reported the BEP of the three feature sets: the comparison is against the baseline, i.e. the best non linguistic result of Tab. 1, although reestimation of the parameters has been carried out (as shown by Fig. 1).

We observe that both POS tag (column 4 *vs* column 3) and terminological expressions (column 3 vs column 1) produce improvements when included as features. Moreover PNs seems not to bring more information than POS tags, as column 2 suggests. The best model is the one using all the linguistic features provided by NLP. This increases performance ($> 1\%$) which is not negligible if considering the very high baseline.

Table 2: Breakeven points of $Rocchio_{\gamma_i}$ on three feature set provides with NLP applied to Reuters version 3

Base-Line	+POS-PN	+PN+TE	+PN+TE+POS
83.82%	83.86%	84.48%	84.94%

5 Discussion

In Table 3 the performances of the most successfull methods proposed in literature are reported. Some of their distinctive aspects are here briefly summarized. Support Vector Machines SVM recently proposed in [Joachims, 1999] is based on the structural risk minimisation principle. It uses quadratic programming technique for finding a surface that ”best” separates the data points (the representation of training documents in the vector space model) in two classes. K-Nearest Neighbour is an example-based classifier, [Yang and Liu, 1999], making use of document to document similarity estimation that selects a class for a document through a k-nearest heuristics. RIPPER [Cohen and Singer, 1996] uses an extended notion of profile, by learning contexts that are positively correlated with the target classes. A machine learning algorithms allows the ”contexts” of a word w to affect how (or whether) presence/absence of w contribute actually to a classification. CLASSI is a system that uses a neural network-based approach to text categorization [H.T. Ng, 1997]. The basic units of the network are only perceptrons. Dtree [Quinlan, 1986] is a system based decision trees. The Dtree model allows to select relevant words (i.e. features), according to an information gain criterion. CHARADE [I. Moulinier and Ganascia, 1996] and SWAP1 [Apté *et al.*, 1994] use machine learning algorithms to inductively extract Disjunctive Normal Form rules from training documents. Sleeping Experts (EXPERTS) [Cohen and Singer, 1996] are learning algorithms that works on-line. They reduce the computation complexity of the training phase for large applications updating incrementally the weights of *n*-gram phrases.

Two major conclusions can be drawned. First of all the parameter estimation proposed in this paper is a significant improvement with respect to other proposed uses of the Rocchio formula. The application of this method over crude fea-

Table 3: BEP of best classifiers on Reuters3 - Revised

SVM	KNN	$Rocchio_{\gamma_i}$+NLP	$Rocchio_{\gamma_i}$
85.99%	85.67%	**84.94%**	**83.82%**
RIPPER	CLASSI	DTREE	SWAP1
80%	80%	79%	79%
CHARADE	EXPERT	Rocchio	Naive Bayes
78%	76%	*82.61%* (75% - 79.9%)	71%-79%

ture sets (i.e. simple words and without any selection) improve significantly with respect to the best obtained Rocchio methods (83.82% vs 79.9%). This weighting scheme is a robust filtering technique for sparse data in the training corpus. It has been suitably applied to derive the baseline figures for contrastive analysis of the role of linguistic features.

The comparative evaluation of simpler feature sets with linguistically motivated information (i.e. POS tagged lemmas and terminological information) suggests the superiority of the latter. This is mainly due the adoption of the optimal selection and weighting method proposed. It provides a systematic way to employ the source linguistic information. It has to be noiced that in the set of 9,650 features (including linguistic ones) derived from the training material of the Reuters3 *acq* category, only 4,957 ($\sim$ 51,3%) assumes a weight greater than 0 after γ_{acq} optimization is carried out. Notice that the use of a single (global) γ value over linguistic features (i.e. +POS+PN+TE), shown in Fig. 1 (*LingFeature* plot), reaches a best BEP of about 0.828: this is improved of more that 2% in BEP when selective γ_i setting is applied (Tab. 2). This form of weighting is thus responsible for an optimal employment of linguistic information that is, by its nature, often affected by data sparseness and noise.

6 Conclusion

In this paper a new model able to exactly measure the contribution given by NLP has been designed and experimented. It brings significant evidence of the role of natural language processing techniques in the specific TC area of IR. The benefits of NLP methods are the efficient extraction of linguistically motivated complex features (including multi-word patterns). A novel weighting method has been also proposed. It provides a systematic feature selection functionality with a systematic estimation of the γ_i parameters in the Rocchio formula. The method is robust and effective wrt noise as analysis over non linguistic feature sets demonstrates. This gave us the possibility of focusing on the measurement of relevance of NLP derived features. The $Rocchio_{\gamma_i}$ applied to linguistic material supports thus a computationally efficient classification (typical of purely statistical models) and produces performances (about 85%) comparable with the best (but computationally more expensive) classifiers (e.g. KNN and SVM).

References

[Apt´e *et al.*, 1994] Chidanand Apt´e, Fred Damerau, and Sholom Weiss. Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems*, 12(3):233–251, 1994.

[Arppe, 1995] A. Arppe. Term extraction from unrestricted text. In *NODALIDA*, 1995.

[Basili *et al.*, 1997] R. Basili, G. De Rossi, Pazienza, and M.T. Inducing terminology for lexical acquisition. In *Proceedings of the Second Conference on Empirical Methods in NLP, Providence, USA*, 1997.

[Basili *et al.*, 2000a] R. Basili, L. Mazzucchelli, and M.T. Pazienza. An adaptive and distributed framework for advanced ir. In *In proceeding of 6th RIAO Conference, Content-Based Multimedia Information Access, Collge de France, Paris, France*, 2000.

[Basili *et al.*, 2000b] R. Basili, A. Moschitti, and M.T. Pazienza. Language sensitive text classification. In *In proceeding of 6th RIAO Conference, Content-Based Multimedia Information Access, Collge de France, Paris, France*, 2000.

[Cohen and Singer, 1996] William W. Cohen and Yoram Singer. Context-sensitive learning methods for text categorization. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 96')*, pages 12–20, 1996.

[Daille, 1994] B. Daille. Study and implementation of combined techniques for automatic extraction of terminology. In *The Balancing Act* WorkShop of the *32nd Annual Meeting of the ACL*, 1994.

[Grefenstette, 1997] Gregory Grefenstette. Short queries linguistic expansion techniques: Palliating one-word queries by providing intermediate structures to text. In M.T. Pazienza, editor, *Information Extraction*. Springer Verlag, Berlin, 1997.

[H.T. Ng, 1997] K.L. Low H.T. Ng, W.B. Goh. Features selection, perceptron learning, and a usability case study for text categorization. In *Proceedings of the 20th ACM SIGIR Conference*, pages 67–73, 1997.

[I. Moulinier and Ganascia, 1996] G. Raskinis I. Moulinier and J. Ganascia. Text categorization: a symbolic approach. In *In Proceedings of the 5th Annual Symposium on Document Analysis and Information Retrieval*, 1996.

[Ittner *et al.*, 1995] David J. Ittner, David D. Lewis, and David D. Ahn. Text categorization of low quality images. In *Proceedings of SDAIR-95, 4th Annual Symposium on Document Analysis and Information Retrieval*, pages 301–315, Las Vegas, US, 1995.

[Joachims, 1999] T. Joachims. Transductive inference for text classification using support vector machines. In *I. Bratko and S. Dzeroski* editors, *Proceedings of ICML-99, 16th International Conference on Machine Learning*, pages 200–209,1999.

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

[Lewis *et al.*, 1996] David D. Lewis, Robert E. Schapiro, James P. Callan, and Ron Papka. Training algorithms for linear text classifiers. In *Proceedings of 19th ACM SIGIR-96*, pages 298–306, Zürich, CH, 1996.

[Quinlan, 1986] J.R. Quinlan. Induction of decision trees. In *Machine Learning*, pages 81–106, 1986.

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

[Salton, 1991] G. Salton. Development in automatic text retrieval. *Science*, 253:974–980, 1991.

[Yang and Liu, 1999] Yiming Yang and Xin Liu. A re-examination of text categorization methods. In *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999.

[Yang and Pedersen, 1997] Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of 14th ICML-97*, pages 412–420, Nashville, US, 1997.

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