# Natural Language Processing and Automated Text Categorization

## A study on the reciprocal beneficial interactions

A dissertation submitted to the department of

Computer Science, Systems and Production

in candidacy for the degree of Doctor of Philosophy in

Computer Science and Control Engineering

by

Alessandro Moschitti
May 8, 2003

University of Rome
*Tor Vergata*

# Abstract

Modern Information Technologies and Web-based services are faced with the problem of selecting, filtering and managing growing amounts of textual information to which access is usually critical. Text Categorization is a subtask of Information Retrieval that allows users to browse more easily the set of texts of their own interests, by navigating in category hierarchies. This paradigm is very effective for retrieval/filtering of information but also in the development of user-driven on-line services.

Given the large amounts of documents involved in the above applications, automated approaches to categorize data efficiently are needed. Standard statistical Machine Learning models, use the bag-of-words representation to train the target classification function. Only the single words, contained in the documents, are used as features to learn the statistical models. Typical natural language structures, e.g., morphology, syntax and semantic are completely neglected in the developing of the classification function. In turn, the semantic information generated by the Text Categorization models is not used yet for the most important natural language applications. Information Extraction, Question/Answering and Text Summarization should take advantage from category information as it helps to select the domain knowledge that language applications usually use in their processing.

In this thesis, a study of the interaction between Natural Language Processing and Text Categorization has been carried out for operational applications. Since these latter require high efficient and accuracy, we have studied and implemented models that own both characteristics. Next, with the aim to enhance the accuracy in statistical Text Categorization, we have examined the role of Natural Language Processing in document representations. The extensive experimentation of the most part of Natural Language Processing techniques for Information Retrieval has shown the ineffectiveness of current linguistic processing for improving statistical Text Categorization. On the contrary, preliminary experiments on some of the most important natural language systems such as Information Extraction, Question/Answering and Text Summarization, have shown promising enhancements by exploiting Text Categorization models.

*To Alan, Elisabetta, Guido, Monica and Rossana*
*my family, old and new*

## Acknowledgements

I would like to thank my supervisor prof. Roberto Basili that advised me since the earlier years of my PhD course. Most part of the content of this thesis is based on several scientific papers that we wrote together. Then I sincerely need to thank prof. Maria Teresa Pazienza that enrolled me in her Artificial Intelligence and Natural Language Processing group.

Many thanks to Dr. Michele Vindigni, with whom I had the earlier discussion about Text Categorization. He designed and implemented, for my earlier studies, the first software for the evaluation of micro-average Precision and Recall of the classifier embedded in TREVI. Particular thanks are devoted to my colleague Dr. Fabio Massimo Zanzotto with whom I had daily dialectical debates on the use of Natural Language Processing for Information Retrieval. His knowledge together with the Roberto's ones often embodied the NLP soul in my experiments on statistical text categorization. Moreover, I want to thank Dr. Massimo di Nanni who helped me in the first release of the profile-based text classifier software and Armando Stellato for his daily help to our group. Thanks to the other PhD students and staff of ”Tor Vergata” University. In particular, Mauro Andreolini, Valeria Cardellini, Emiliano Casalicchio and Marco Emilio Poleggi made these studying years more pleasant.

I wish to thank, prof. Paola Velardi that advised me on the early steps in the academic career, prof. Giuseppe Iazeolla and prof. Vittoria de Nitto as advisors for my doctorate exams.

Last but not least, I have to thank prof. Sanda Harabagiu and prof. Dan Moldovan from whom I have been visiting scholar also during the writing of this thesis. In particular, the Chapter 4 of this thesis could not be realized without the help of prof. Harabagiu.

# Contents

**Appendix**

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Modern Information Technologies and Web-based services are faced with the problem of selecting, filtering and managing growing amounts of textual information to which access is usually critical. Information Retrieval (IR) is seen as a suitable methodology for automated management of information/knowledge as it includes several techniques that support an accurate retrieval of information and the consequent user satisfaction. Among the others, the classification of electronic documents in general categories (e.g., *Sport*, *Politic*, *Religion*,..) is an interesting mean to improve the performances of IR systems: (a) users can more easily browse the set of documents of their own interests and (b) sophisticated IR models can take advantages of the categorized data. As an example, the authoring of the textual documents is carried out using the document *contents*. A preliminary categorization step provides an indication of the main areas of interest. Text Categorization (TC) is, thus, playing a major role in retrieval/filtering but also in the development of user-driven on-line services.

Given the large amount of documents involved in the above processes, automated approaches to categorize data are needed. Machine Learning techniques are, usually applied to automatically design the target classification function using a set of documents (*learning-set*), previously assigned in the target categories. Such learning algorithms need statistical document representations. The most common representation is the so-called *bag-of-words*, i.e. only the simple document words, are used for feeding the learning algorithm. The linguistic structures (e.g., morphology, syntax and semantic) typical of natural language documents are completely neglected. Nevertheless, this approach has shown high accuracies in the automated classification of a set of unseen documents (*test-set*).

As the vital importance of information for some specific sectors ranging from changes in management positions to business intelligence or information about terrorist acts, the accuracy in selecting only the suitable data has become a crucial issue. The consequence is that more and more accurate TC learning models have been designed: on one hand, researchers have attempted to improve the categorization algorithm by using several theoretical learning models (e.g., [Joachims, 1998; Yang, 1999; Tzeras and Artman, 1993; Cohen and Singer, 1999; Salton and Buckley, 1988; Ng *et al.*, 1997; I. Moulinier and Ganascia, 1996; Apté *et al.*, 1994; Quinlan, 1986; Hull, 1994; Schütze *et al.*, 1995; Wiener *et al.*, 1995; Dagan *et al.*, 1997; Lewis *et al.*, 1996; Ittner *et al.*, 1995]); on the other hand, document representations more sophisticated than *bag-of-words* have been experimented.

The designing of more effective TC models has produced an increase of the time complexity for both training and classification phases. On the contrary, an important requirement of the current operational scenarios is efficiency. For instance, web applications require effective data organization and efficient retrieval as for the huge and growing amount of documents. In order to govern the overall complexity, the current trend is the designing of efficient TC approaches [Lewis and Sebastiani, 2001]. A careful analysis of the literature reveals that on-line classifiers are the most (computationally) efficient models [Sebastiani, 2002]. These are based on a vector representation of both documents and categories by means of feature weights derived via different approaches [Hull, 1994; Schütze *et al.*, 1995; Wiener *et al.*, 1995; Dagan *et al.*, 1997; Lewis *et al.*, 1996; Cohen and Singer, 1999]. The decision if a document belongs or not to a category is then made measuring the similarity between the target vector pair (i.e., document and category).

The drawback of the above classifiers is the accuracy lower than other more complex classification algorithms. An approach to improve the accuracy, maintaining the same complexity, is the use of a richer document representation. Linguistic structure [Voorhees, 1993; Strzalkowski and Jones, 1996] could embed more information than the simple words which helps TC systems to learn the differences among different categories. Typical structures that have triggered the interest of IR researchers are complex nominals, *subject-verb-object* relations and the word meaning. This latter, is particularly useful in representing the document content unambiguously. For example the *slide* as transparency for projectors and the *slide* as sloping chute for children are the same words whereas the meaning is completely different. Richer representations, described above, are usually obtained by applying some of the Natural Language Processing (NLP) techniques. Both simple words and complex NLP structures in statistical learning models need to be treated as single units that usually refer to as *features*.

Automated TC, especially when the implementing algorithm is efficient and accurate, has a large applicability in the designing of IR systems. In the same way, IR is usually exploited for designing NLP applications. Information Extraction (IE), *Question/Answering* (Q/A) and Text Summarization (TS) are important NLP applications that use retrieval models. IR helps in locating specific documents within a huge search space (*localization*) while IE or Q/A support the focusing on specific information within a document (*extraction* or *explanation*). Similarly TC is currently used for general NLP applications but the advantages that it can provide for IE, Q/A and TS systems are less obvious. Anyhow, text classifiers provide for each text a set of categories that constitute an important indication of what are the main subjects of the document. The availability of this category information enables the use of domain-specific NLP techniques. For instance, text classifiers can assign categories to small texts also, e.g., paragraphs or passages. This knowledge can be exploited by IE, Q/A and TS systems to respectively extract the relevant facts, choose the correct answers, select the important passages that are related to target domain.

In this thesis a study of the interaction between NLP and TC, in operational scenarios has been carried out. Since real applications require, high efficiency and accuracy, we have studied and implemented a model that owns both characteristics. Next, we have examined the role of NLP in document representation with the aim to further boost the accuracy of the proposed model as well as the other TC approaches. Finally, original models that use TC for improving NLP systems have been presented.

## 1.1 Efficient Models for Automated TC

Text Categorization is the task of assigning documents to predefined categories. It is an active research area in Information Retrieval and machine learning. A wide range of supervised learning algorithms have been applied to this problem. The classification problem can be modeled as follows. *Given* a set of user interests expressed into classes (i.e. topics/subtopics labels), $\mathcal{C} = \{C_1, ...., C_{|\mathcal{C}|}\}$ and a variety of existing documents already categorized in these classes (i.e. *training-set*), *build a decision function,* $\phi$ able to decide the correct classes for texts, i.e. $\phi : D \rightarrow 2^C$. The decision function is thus asked to map newly incoming documents ($d \in D$) in one (or more) class(es), according to their content.

### 1.1.1 Designing a Text Classifier

The design of general text classifiers foresees a set of tasks universally recognized by the research community:

- *Features design*: in this phase the following pre-processing steps are carried out:
  - *Corpus processing*, filtering, and formatting all the documents belonging to the corpus.
  - *Extraction of relevant information.* Usual approaches make use of words as basic units of information. A *stop list* is here applied to eliminate function words (that exhibit similar frequencies over all classes). The linguistic information that characterizes a document (and its class) is here taken into account. Features more complex than simple words can be built as structured patterns (i.e. multiple word expressions), or by adding lexical information (e.g., word senses).
  - *Normalization.* Word stemming, carried out by removing common suffixes from words, is a classical method applied here. Words after stemming are usually called *stems.* When more complex features

are available via linguistic analysis (i.e. words and/or complex nominal), usually normalization refers to the activity of lemmatization (i.e. detection of the base form of rich morphological categories, such as nouns or verbs[1]).

  – *Feature selection*, which is an attempt to remove non-informative terms from documents to improve categorization effectiveness and reduce computational complexity. Typical selection criteria are $\chi^2$, information gain or document frequency.

- *Feature Weighting*: features assume usually different roles in documents, i.e. they are more or less representative. Different weights are associated to features via different, often diverging, models.

- *Similarity estimation* is modeled via operations in spaces of features. This can be carried out between pairs of documents or between more complex combinations of features (e.g., profiles as the combination of features coming from different representative documents). Usually quantitative models (i.e. metrics) are adopted for this.

- *Inference*: similarity among document/profile representations activates the target classification decision. Assignment of an incoming document to a target class is based on a decision function over similarity scores. Different criteria (i.e. purely heuristics or probability-driven rules) are used in this task.

- *Testing*: the accuracy of the classifier is evaluated by using a set of prelabeled documents (i.e. *test-set*) that are not used in the learning phase (*training-set*). The labels produced by the classifier are compared to the correct ones. The result of this phase is usually one or more numerical scores that provide a measure of the distance between the human choice (embodied by the training data) and the underlying categorization system.

### 1.1.2 Profile-based Text Classifier

Among linear classifiers the *profile-based* [Sebastiani, 2002] provide an explicit representation of each category. The salient information about target categories is acquired during the learning phase and collected in independent profiles. This information can be accessed in linear time during the classification process, thus resulting in fast categorization algorithms. The major advantage is their efficient impact in any real scenario like on line document classification, fast company document management and batch classification of millions of documents. Unfortunately their low computational cost is draw backed by their poorer performance than other complex approaches in terms of precision and recall.

*Profile-based* classifiers derive a description of each target class ($C_i$) in terms of a profile, usually a vector of weighted terms. These vectors are extracted from

[1] Notice that this is very important for languages with a rich generative morphology where hundreds of different forms are derived from the same root.

previously categorized documents under $C_i$ used for training the system. Classification proceeds through the evaluation of similarity between the incoming document $d$ and the different profiles (one for each class). As an example, early profile-based classifier made use of the Vector Space Model [Salton and Buckley, 1988] to define similarity. Notice that main advantages of such an approach are its computational efficiency and easy implementation.

The development of a profile-based classifier requires a specialization of some phases:

- *Features Weighting*, i.e. the building of the synthetic profiles can be defined by two steps:
  - the development of a representation $\vec{d}$ for documents $d$. $\vec{d}$ is defined over the features $f$ extracted from $d$. Components $d_i$ are weights of those features.
  - the development of a representation $\vec{C_i}$ for a class $C_i$. It summarizes the representations $\vec{d}$ of all the positive instances of $C_i$ (i.e. $d \in C_i$)
- *Similarity estimation* in profile-based classifiers is always carried out between unknown (i.e. not classified) documents $d$ and the above defined profiles ($\vec{C_i}$). Similarity is usually established within the space determined by the features (i.e. weighted elements of vectors $\vec{d}$ and $\vec{C_i}$). Section 2.3 discusses different techniques.
- *Inference*: A decision function is usually applied over the similarity scores. The most widely used inference methods are: probability, fixed and proportional threshold. These are respectively called in [Yang, 1999]: *Scut* (a threshold for each class exists and is used to decide whether or not a document belong to it), *Rcut* (the best $k$-ranked classes are assigned to each document) and *Pcut* (the *test-set* documents are assigned to the classes proportionally to their size). Given the importance of these inference methods, a more complete definition and discussions will be given in the next chapter.

### 1.1.3 Some Methods of Text Categorization

In the literature several TC models based on different machine learning approaches have been developed. Whatever is the technology, the adopted models suffer by the trade-off between performance in retrieval and complexity in training and processing. This last, is crucial in operational scenarios and it makes the adoption of the best figure model unappealing. In the following, we briefly revisit the well-known approaches as well as more recent ones. Particular carefulness to operational aspects will be devoted.

Support Vector Machines ($SVM$), recently proposed in [Joachims, 1998], use the Structural Risk Minimization principle in order to assign (or not) a document to a class. This technique is applied to a vector space to obtain the "best" separating hyperplane, which divides the points associated to the training documents in two classes (positive and negative examples). A quadratic programming technique finds out the hyperplane's gradient vector with the minimum Euclidean Norma. This guarantees a minimum distance between the nearest documents of different classes and the hyperplane itself. This classifier has been successfully applied on academic benchmarks as it provides the highest performances (about 86% on Reuters). On those corpora it seems characterized by fast training and processing. The problems arise when it is applied to operational scenarios where the number of training documents is hundreds time greater than the number of documents contained in benchmarks. The disadvantages of SVMs are that the training time can be very large if there are large numbers of training examples and execution can be slow for nonlinear SVMs as it has been pointed out in [Drucker *et al.*, 1999]. In fact, as the number of documents grows, the number of support vectors increase in a non-well understood proportional law. This means that thousands of support vectors, for assigning each single documents, could be involved in classification phase. As each support vector requires a scalar product with the input documents the time for an online classification is usually very high.

$KNN$ is an example-based classifier, [Yang, 1994], making use of document to document similarity estimation that selects a class for a document through a $k$-Nearest Neighbor heuristic. In this case the algorithm requires the calculation of the scalar products between an incoming document and those contained in the *training-set*. The optimization, proposed by the EXP-NET algorithm [Yang, 1994] , reduces the computational complexity to $O(n \times log(n))$ time, where $n$ is the maximum among the number of training documents, the number of categories and the number of features.

*Rocchio* [Ittner *et al.*, 1995; Cohen and Singer, 1999] often refers to TC systems based on the Rocchio's formula for profile estimation. Its major drawback is the low accuracy whereas its efficiency is very high since the learning as well as the classification time is $O(Nlog(N))$, where $N$ is the number of features. Extensions of the algorithm have been given on [Schapire *et al.*, 1998] and [Lam and Ho, 1998] but both approaches relevantly increase the Rocchio complexity.

$RIPPER$ [Cohen and Singer, 1999] uses an extended notion of a profile, by learning contexts that are positively correlated with the target classes. A machine learning algorithm allows the *contexts* of a word $w$ to decide how (or whether) presence/absence of $w$ contribute actually to the classification process. As it is based on profiles, it can be very fast in on line classification task, but it has a noticeable learning time. Moreover, given the complexity for deriving phrases, it is not clear if it can be applied to a huge document space (i.e., millions of documents).

$CLASSI$ is a system that uses a neural network-based approach to text categorization [Ng *et al.*, 1997]. The basic units of the network are only perceptrons. Given the amount of data involved in typical operational scenarios the size of the target networks makes the training and classification complexity prohibitive.

*Dtree* [Quinlan, 1986] is a system based on a well-known machine learning method (i.e. decision trees) applied to training data for the automated derivation of a *classification tree*. The *Dtree* model allows to select relevant words (i.e. features), via an information gain criterion, and, then, to predict categories according to the occurrence of word combinations in documents. It efficiently supports on line classification as an attribute tree describes the categories. However the learning time is considerable.

*CHARADE* [I. Moulinier and Ganascia, 1996] and *SWAP*1 [Apté *et al.*, 1994] use machine learning algorithms to inductively extract Disjunctive Normal Form rules from training documents.

*Sleeping Experts* (EXPERTS) [Cohen and Singer, 1999] are learning algorithms that work on-line. They reduce the computation complexity of the training phase for large applications updating incrementally the weights of $n$-gram phrases. The reduced complexity makes it appealing for a real application but as for Rocchio algorithms the performances are far from the *state-of-the-art*.

*Naive Bayes* [Tzeras and Artman, 1993] is a probabilistic classifier that uses joint probabilities of words and categories to estimates the conditional probabilities of categories given a document. The naive approach refers to the assumption of word independence. Such assumption makes the computation of *Naive Bayes* classifier far more efficient than the exponential complexity of a pure Bayes approach (i.e. where predictors are made of word combinations). In this case the only problem is the low performance in terms of retrieval that it shows on every corpus.

The above models have been compared on a well-known document corpus, i.e. Reuters news collection. Unfortunately, as it has been pointed out in [Yang, 1999] five Reuters versions exist and the TC systems perform differently on them. Table 1.1, indeed, reports system accuracies[2] that have been measured either on Reuters 22173 or on Reuters 21578. Both of these versions have been split between training and testing sets in two different ways: Apté and Lewis modalities [Sebastiani, 2002]. It is worth noticing that the same classifier can achieve different performances on different Reuters versions/splits. Thus, Table 1.1 gives only an approximate ordering of models in terms of accuracy. Moreover the same model is subject to several implementations or variations. For example Naive Bayes has been reported by Yang to have differences in performance: 71% [Yang, 1999] vs. 79.56% [Yang and Liu, 1999].

According to the Table 1.1, the best figure on the Reuters corpus is obtained by the example-driven *KNN* classifier (82.3/85%[3]) and by SVM (86%). However, as previously discussed they have a heavier training and classification complexity, which makes their design and use more difficult within real operational domains. Other classifiers having a fast on line classification (e.g., RIPPER, SWAP-1) are based on complex learning and they do not show performances comparable to the best figure classifiers.

[2] It has been done by means of the breakeven point that is the point where recall and precision assume the same value. A complete description of the most common methodology used to measure text classifier accuracy is given in next chapter.

[3] The higher values (85%) refers to an evaluation in which not labeled documents were removed from the corpus. This makes the results not realistic.

Table 1.1: Accuracy of the most famous models on the Reuters corpus

| *SVM* | *KNN* | RIPPER | CLASSI | Naive Bayes |
|---|---|---|---|---|
| 86% | 85/82.3 % | 81/82% | 80.2% | 71/79.56% |
| SWAP1 | CHARADE | EXPERT | *Rocchio* | Dtree |
| 79/80.5% | 73.8/78.3% | 75.2/82.7% | 74.8/78.1% | 79.4% |

On the contrary, Rocchio text classifier is very efficient but it has an accuracy 8% below the best figure. In order to impact the *trade-off* accuracy/complexity in Chapter 2 we present an original model, the Parameterized Rocchio Model (*PRC*) [Basili *et al.*, 2001; Basili and Moschitti, 2002] that allows to maintain the same Rocchio complexity and to highly improve its accuracy. This result, allows us to partially satisfy the first aim of this thesis, i.e. the designing of efficient and accurate model for TC. Further improvement is needed as it will be shown that the proposed model is still less accurate than the best figure text classifiers. In the next section some improvements of document representation are presented as potential directions for increasing the accuracy of TC models.

## 1.2 The role of NLP in IR

The above section has shown several machine learning approaches that aimed to improve TC. Other studies relate to the designing of a more effective document representation, to increase the accuracy. Documents, as previously introduced, are usually described as pairs <*feature, weight*>, consequently, more suitable representation for the learning algorithm can be modeled using either a more effective weighting schemes [Singhal *et al.*, 1995; Robertson and Walker, 1994; Buckley and Salton, 1995; Sable and Church, 2001], or by adopting alternative features instead of the simple words. In IR several attempts to design complex and effective feature for document retrieval and filtering have been carried out. Some of the well-known representations are:

- Lemmas, i.e., the base form of rich morphological categories, like nouns or verbs. In this representation, lemmas replace the words in the target texts, e.g., *acquisition* and *acquired* both transform in *acquire*. This should increase the probability to match the target concept, e.g., *the act of acquiring* against texts that express it in different forms, e.g., *acquisition* and *acquired*. Lemmatization improves the traditional stemming techniques used in IR. In fact, the stems are evaluated by making a rough approximation of the real root of a word. The result is that many words with different meanings have common stems, e.g., *fabricate* and *fabric*, and many stems are not words, e.g., *harness* becomes *har*.

- *Phrases* relate to the sentence subsets in term of subsequences of words. Several phrase types have been defined:

  - *Simple n-grams*, i.e., sequences of words selected by applying statistical techniques. Given a document corpus all consecutive $n$-sequences of (non-function) words are generated, i.e. the $n$-grams[4]. Then statistical selectors based on occurrences and/or document frequencies of $n$-grams are applied to select those most suitable for the target domain. Typical used selectors, e.g., *mutual information* or $\chi^2$, are described in the next chapter as they are also used in standard feature selection.

  - *Nouns Phrase*, e.g., Proper Nouns and Complex Nominals. A simple regular expression such as $N^+$ (i.e., every sequence of one or more nouns) based on word categories (e.g., nouns, verbs and adjectives) can be used to select the complex term *Minister of Finance* and discard the non-feasible term *Minister formally.* The words *Ministers* and *Finance*, in the first phrase, are often referred to as *head* and *modifier* respectively. More modifiers can appear in a complex nominal, e.g., the phrase *Satellite Cable Television System* is composed of the tree nouns *Satellite*, *Cable* and *Television* that modify the head *System.*

  - $<head, modifier_1, .., modifier_n>$ *tuples*. Parsers, e.g., [Charniak, 2000; Collins, 1997; Basili *et al.*, 1998c] are used to detect complex syntactic relations like *subject-verb-object* to select more complex phrases, e.g., *Minister announces plans*, from texts. An interesting property is that these tuples can contain non adjacent words, i.e. tuple components can be words that are subject to long distance dependencies. Such tuples hardly can be detected via pure statistical models. In [Strzalkowski and Jones, 1996] only the *subject-verb* and *verb-object* pairs named the *<head, modifier> pairs* have been used (see Section 1.2).

  The aim of phrases is to improve the precision on concept matching. For example documents in an *Economic* category could contain the phrase *company acquisition* whereas an *Education* category could include term like *language acquisition.* If the word *acquisition* alone is taken as feature, it will not be useful to distinguish between the two target categories. The whole phrases, instead, give a precise indication of which is the content of the documents.

- *Semantic concepts*, each word is substituted with a representation of its meaning. Assigning the meaning of a content word depends on the definition of word senses in semantic dictionaries. There are two ways of defining the meaning of a word. First, the meaning may be explained, like in a dictionary entry. Second, the meaning may be given through other words that share the same sense, like in a thesaurus. WordNet encodes both forms of meaning definitions. Words that share the same sense are

[4] The term $n$-grams in IR is also referred to as the sequences of $n$ characters from text.

said to be *synonyms* and in WordNet, a set of synonym words is called a *synset*. The advantage of using word senses rather than words is a more precise concept matching. For example, the verb *to raise* could refer to: (a) *agricultural texts*, when the sense is *to cultivate by growing* or (b) *economic activities* when the sense is *to raise costs*.

### 1.2.1 NLP for Text Retrieval

The above techniques appear feasible for improving IR systems, nevertheless, the use of NLP in IR has produced controversial results and debates. In TREC-5 and TREC-6 [Strzalkowski and Jones, 1996; Strzalkowski and Carballo, 1997], document retrieval based on stems has been slightly improved using phrases, noun phrases, *head-modifier* pairs and proper names. However, their evaluation was done on *ad-hoc* retrieval mode only, as the less efficient NLP techniques could not be applied to the same *testing-set* of the pure statistical models. This prevented the comparison with the *state-of-the-art* retrieval systems. In [Strzalkowski *et al.*, 1998; Strzalkowski and Carballo, 1997] a high improvement of retrieval systems was obtained using topic expansion technique. The initial query was expanded with some related passages not necessarily contained inside the relevant documents. The NLP techniques used in TREC-6 have been used to further increase the retrieval accuracy. The success of the above preliminary experiments was not repeated in TREC-8 [Strzalkowski *et al.*, 1999] as the huge amount of data made impossible the correct application of all required steps. The conclusion was that the higher computational cost of NLP prevents its application in operative IR scenario. Another important conclusion was:

*NLP representations can increase basic retrieval models (e.g., SMART) that adopt simple stems for their indexing but if advanced statistical retrieval models are used NLP does not produce any improvement. [Strzalkowski* et al.*, 1998].*

In [Smeaton, 1999] a more critical analysis is made. In the past, the relation between NLP and Machine Translation (MT) has always been close. Thus, much of NLP research has been tailored to the MT applications. This may have prevented that NLP techniques were compatible with task such as retrieval, categorization or filtering. [Smeaton, 1999] assesses that when pure retrieval aspects of IR are considered, such as the statistical measures of word overlapping between queries and documents, the NLP that has been developed recently, has little influence on IR. Moreover, NLP is not useful to retrieve documents when they do not contain many, or, any of the query terms. Current IR is not able to handle cases of different words used to represent the same meaning or concepts within documents or within queries. Polysemous words, which can have more than one meaning, are treated as any other word. Thus, [Smeaton, 1999] suggests to drop the idea of using NLP techniques for IR, instead he suggested to exploit the NLP resources like WordNet. In this perspective Smeaton used WordNet to define a semantic similarity function between noun pairs. The purpose was to retrieve documents that contain terms similar to those included inside the query. As many words are polysemous, a Word Sense Disambiguation algorithm was developed to detect the right word senses. As such algorithm produced a performance ranging between 60-70%, the semantic similarity led to positive results only after the senses were manually validated.

Other studies using semantic information for improving IR have been carried out in [Sussua, 1993] and [Voorhees, 1993; 1994]. They report the use of word semantic information for text indexing and query expansion respectively. The poor results obtained in [Voorhees, 1994] show that semantic information taken directly from WordNet without performing any kind of WSD is not helping IR at all. In contrast, in [Voorhees, 1998] promising results on the same task were obtained after that the senses of select words were manually disambiguated.

In summary the analysis of the literature reveals that the more likely reasons for the failure of NLP for IR are the following:

- High computational cost of NLP due prevalently to the use of the parser in detecting syntactic relations, e.g., the <*head, modifier*> pairs. This prevented a systematic comparison with *the-state of-the-art* statistical models

- Small improvements when complex linguistic representation is used. This may be caused either by the NLP errors in detecting the complex structures or by the use of NLP derived features as informative as the *bag-of-words*.

- The lack of an accurate WSD tools, in case of semantic representation: (a) The ambiguity of the words causes the retrieval of a huge number of irrelevant documents if all senses for each query words are introduced, or (b) if a WSD with 60% is employed to disambiguate document and query word senses, the retrieval precision decrease proportionally to the error, i.e., 40%.

### 1.2.2 NLP for Text Categorization

As the literature work has shown the failure of NLP for IR why should we try to use it for TC? Text categorization is a subtask of IR, thus, the above results should be the same for TC also. However, there are different aspects of TC that require a separated study as:

- In TC both set of positive and negative documents describing categories are available. This enables the application of theoretical motivated machine learning techniques. These methods better exploit and select the document representations.

- Categories differ from queries as they are fixed, i.e., a predefined set of training documents completely define the target category. This enables the use of feature selection techniques to select relevant features and filtering out those irrelevant also derived from NLP errors.

- There is no query involved in the TC task: (a) documents can be retrieved based on the training documents, which provide a stable routing profile, and (b) the smallest data unit is the document for which it is available a more reliable statistical word distribution than in queries.

- Effective WSD algorithms can be applied to documents whereas this was not the case for queries (especially for the short queries). Moreover, an evaluation of WSD tools has been recently carried out in SENSEVAL [Kilgarriff and Rosenzweig, 2000]. The results are an accuracy of 70% for verbs, 75 % for adjectives and 80% for nouns. This last result makes viable the adoption of semantic representation at least for the nouns.

- TC literature studies report contrasting results on the use of NLP for TC. Even if in [Lewis, 1992] is shown that using phrases and phrase clusters generates a decrease of classification accuracy on Reuters documents. On the contrary, more recent results from [Basili *et al.*, 2000a; 2001; 2002] show that including some syntactic information, such as recognition of proper nouns and other complex nominals in the document representation can slightly improve the accuracy of some weak TC models such as the Rocchio classifier. Other work using phrase [Furnkranz *et al.*, 1998; Mladenić and Grobelnik, 1998; Raskutti *et al.*, 2001; Bekkerman *et al.*, 2001; Tan *et al.*, 2002] report noticeable improvement over the *bag-of-words*. These results require a careful analysis that will be carried out.

Semantic information for TC was experimented in [Scott and Matwin, 1999]. WordNet senses have been used to replace the simple words without any word sense disambiguation. The results were mixed as improvements were derived only for small corpus. When a more statistical reliable set of documents was used the adopted representation resulted in performance decrease.

In this paper, the impact of richer document representations on TC has been investigated. The results confirm that even for TC that current NLP tools do not improve text categorization. Explanations of why current NLP does not work as expected as well as the explanation of contrasting positive results reported in other work are given. This has been shown experimenting different corpora and different linguistically rich representations over three TC learning models. We choose Rocchio, Rocchio Paramterized [**?**] and SVM since richer representation can be really useful only if: (a) it causes very computational efficient classifiers (e.g. Rocchio) to reach the accuracy of the best figure classifier , or (b) it allows a target classifier to perform better than models trained with the *bag-of-words*, for this purpose starting from an high accurate classifier (e.g., SVM) is reccomended . In both cases, NLP would advance the *state-of-the-art* in accuracy or in efficiency.

We chose two different TC approaches: Rocchio [Rocchio, 1971] and SVM [Vapnik, 1995] classifiers. The former is a very efficient TC, so, it would be very appealing (especially for real scenario applications) to bring its accuracy *close* to the most accurate classifier. The latter is one of the best figure TC, consequently, improving it causes an improvement of the *state-of-the-art*.

## 1.3 Text Categorization for NLP

Current Natural Language Processing does not seem appealing to improve the accuracy of TC models, on the contrary TC is currently used for NLP applications. The simplest use of TC for Natural Language systems is the enrichment of documents with their category labels. The TREVI[5] system is an example as its purpose was to provide as much information as possible for the document required by users, e.g., news source, issues date and general categories. Other NLP systems exploit categorization schemes as a navigation method to locate the user needed data. A more complex use of TC relates to the IE, Q/A and Summarization system enhancements.

### 1.3.1 Information Extraction

IE is an emerging NLP technology, whose purpose is to locate specific pieces of information called *facts* (e.g., events or finer grained data), from unstructured natural language texts. These information is used to fill some predefined database table, i.e. *the templates.* Current methods extract such information by using linguistically motivated patterns. Each pattern is a regular expression for which is provided a mapping to a logical form. For example given the following fragment of the Reuters news:

```
WASHINGTON, June 2 - Two affiliated investment firms told
the Securities and Exchange Commission they have acquired
593,000 shares of Midway Airlines Inc, or 7.7 pct of the total
outstanding common stock.  The firms, Boston-based FMR Corp
and Fidelity International Ltd, a Bermuda-based investment
advisory firm, said they bought the stake "to acquire an equity
interest in the company in pursuit of specified investment
objectives...."
```

A typical template that aims to represent information relative to the acquisition of companies may be described by the Table 1.2. Note that to correctly fill the template a coreference between *Two affiliated investment firms* and *The firms, Boston-based FMR Corp and Fidelity International Ltd* should be detected.

Each different topic, e.g., *bombing events* or *terrorist acts*, requires different customized pattern sets to extract the related *facts.* The construction of pattern base for new topics is a time-consuming and expensive task, thus methods to automatically generating the extraction pattern have been designed.

Categorized documents have been used to enable the unsupervised patterns extraction in AutoSlog-TS [Riloff, 1996] (See Section 3.3.6). First, all possible patterns that extract noun phrases are generated from documents, using 15 different heuristics. Second, the documents are processed again to extract all

[5] TREVI [Basili *et al.*, 1998b] is a distributed object-oriented system, designed and developed by an European consortium under the TREVI ESPRIT project EP23311, for news agencies in two EU languages, English and Spanish.

| Buyer | Company | Date | Reported-by | # Shares | Pct |
|---|---|---|---|---|---|
| Boston-based FMR Corp and Fidelity International Ltd | Midway Airlines Inc | June 2 | Reuters | 593,000 | 7.7 |

Table 1.2: Example of an Information Extraction template applied to a Reuters news from the *Acquisition* category.

the instances that match the patterns, derived during the first step. Finally, the set of patterns are ranked according to the probability that relevant texts contain the target pattern. The relevant texts for a pattern are assumed to be the documents that belong to the target category. This allows the estimation of the relevance probability for a pattern $p$ as the fraction between the number of instances of $p$ in relevant documents and the total number of instances activated by $p$.

The above method allows the IE designers to save time as the ranked list of patterns can be validated quicker than the manual annotation of the extraction rule from texts. However, the resulting Information Extraction system is clearly domain based and required the manual categorization of the learning documents. An alternative to the manual production of learning data for each application is to use general knowledge valid for any domain. Currently there are two mains linguistic resource based on different knowledge representations: WordNet and FrameNet.

FrameNet is a lexico-semantic database, made recently available[6]. The aim of the FrameNet project is to produce descriptions of words based on semantic frames. Semantic frames, as they have been introduced by [Fillmore, 1982], are schematic representations of situations involving various participants, properties and roles, in which a word may be typically used. The Semantic Frames available from FrameNet are in some way similar to the efforts made to describe the argument structures of lexical items in terms of case-roles or thematic-roles. However, in FrameNet, the role names, which are called Frame Elements (FEs) are local to particular frame structures. For example, the FEs of the ARRIVING frame are *THEME*, *SOURCE*, *GOAL* and *DATE*. They are defined in the following way: the *THEME* represents the object which moves; the *SOURCE* is the starting point of the motion; the *PATH* is a description of the motion trajectory which is neither a *SOURCE* nor a *GOAL*; the *GOAL* is the expression which tells where the theme ends up. A frame has also a description that defines the relations holding between its FEs, which is called the *scene* of the frame. For example, the scene of ARRIVING is: the *THEME* moves in the direction of the *GOAL*, starting at the *SOURCE* along a *PATH*. Additionally, FrameNet contains annotations in the British National Corpus (BNC) of examples of words that evoke each of the frames. Such words are called *target words*, and they may be nouns, verbs or adjectives.

[6] FrameNet is available at the Web site: `www.icsi.berkeley.edu/~framenet`.

This kind of knowledge can be successfully used for generating domain knowledge required for any new domain, i.e. Open-Domain Information Extraction. The corpus annotation available from FrameNet enable us to design learning algorithm that (a) categorize sentences in FrameNet frames and (b) allow, once available the target frame, the recognition of extraction rules for any domain [Moschitti *et al.*, 2003]. Chapter 4 describes in more details the adopted Information Extraction algorithm as well as the use of sentence categorization.

### 1.3.2 Question/Answering

IR techniques have proven quite successful at locating within large collections of documents those relevant to a user's query. Often, however, the user wants not whole documents but brief answers to specific questions like `How old is the President?` or `Who was the second person on the moon?` For this new information needs the sole statistical approach of IR is not sufficient. The result is that a new research area that includes IR and NLP techniques has been consolidating, i.e., Question Answering.

Question Answering (Q/A) is a fast growing area of research and commercial interest: from one hand, it is the only IR subtask that has been proved to be enhanced by NLP; on the other hand, the high capacity of retrieving specific information makes it appealing for business activities, e.g., information management. The problem of Q/A is to find answers to open-domain questions by searching a large collection of documents. Unlike Internet search engines, Q/A systems provide short, relevant answers to questions. The recent explosion of information available on the World Wide Web makes Q/A a compelling framework for finding information that closely matches user needs. One of the important feature of Q/A is the fact that both questions and answers are expressed in natural language. In contrast to the IR methods, Q/A approach deal with language ambiguities and incorporate NLP techniques. All the systems being built in these year exhibit a fairly standard structure: create a query from the user's question, perform IR with the query to locate (segments of) documents likely to contain an answer, and then pinpoint the most likely answer passage within the candidate documents. Answering questions is thus the problem of finding the best combination of word-level (IR) and syntactic/semantic-level (NLP) techniques. The former produces as short a set of likely candidate segments as possible and the latter pinpoints the answer(s) as accurately as possible.

Our idea to improve Q/A systems is to introduce an additional step that uses the TC for filtering incorrect questions and improving the answer ranking. There are two ways to use categorized data in Q/A: (a) to filter paragraphs retrieved by the IR engine and (b) to filter the final answers provided by both IR and NLP processes.

Q/A systems incorporate a paragraph retrieval engine, to find paragraphs that contain candidate answers, as reported in [Clark *et al.*, 1999; Pasca and Harabagiu, 2001]. Then, semantic information, e.g., the class of the expected answers, derived from the question processing, is used to retrieve paragraphs and later to extract answers. Typically, the semantic classes of answers are organized in (hierarchical) ontologies and do not relate in any way to semantic classes typically associated with documents. The ontology of answer classes contains concepts like PERSON, LOCATION or PRODUCT, whereas categories associated with documents are more similar to topics than concepts, e.g., acquisitions, trading or earnings. Given that categories indicate a different semantic information than the class of the expected answer, we argue in this thesis that text categories can be used for improving the quality of textual Q/A.

This approach to our knowledge has not been studied in other Q/A researches. The usual method to exploit text categories to find the desired information is by navigating along subject categories assigned hierarchically to groups of documents, in a style made popular by *Yahoo.com* among others. When the defined category is reached, documents are inspected and the information is eventually retrieved. This is a totally different approach with respect to the methods followed by the use of Q/A models.

In Chapter 4, instead, filtering/re-ranking methods that automatically assigning categories to both questions and texts are presented. The filtering systems allow to eliminate many incorrect answers and to improve the ranking of answers produced by Q/A systems [Moschitti, 2003a]. Additionally, we show that, whenever the semantic class of the expected answer was not recognized, the category information improves the answer ranking. The TC filter was applied to the *LCC Falcon Q/A system* [Pasca and Harabagiu, 2001]. It is the current best figure Q/A system according to TREC 2002 evaluation and it was the best accurate system of past TREC editions.

### 1.3.3 Text Summarization

Text Summarization is the process of distilling the most important information from a source to produce an abridged version for a particular user and task [Chinchor *et al.*, 1998; Kan *et al.*, 2001; Hardy *et al.*, 2001]. It is a hard problem of Natural Language Processing as it implies the understanding of the text content. This latter requires semantic analysis, discourse processing, and inferential interpretation (grouping of the content using world knowledge). As current NLP techniques are not enough accurate to accomplish the above tasks, rather than carrying out true abstraction, approximation are obtained by identifying the most important and central topic(s) of the text, and return them to the reader. Although the summary is not necessarily coherent, the reader can form an opinion of the content of the original. Indeed, most automated summarization systems today produce extracts only.

Following this last approach, there are two main ways to produce a summary:

- *Information Retrieval-based summaries.* Statistical methods are used to find sentences, which are probably the most representative. Thus, the sentence are merged to form an extract (rather than an abstract). The idea is that in this way the essence of all text information is retrieved. The meaning of the words or text is not being considered. This has two advantages: (a) the system needs no ”world knowledge” and (b) by learning the target domain statistics, e.g., words frequencies, the method can be applied on any text domain or even language. It is a ”bottom up” approach: the output is being generated by what is in the text, not by what the user wants to know from it.

- *Information Extraction-based summaries.* In this case, templates that contain the most relevant information, and the patterns for the extraction of template information are designed for the needed summary type. The system knows what type of words to look for in what context and it extracts that information to fill in the templates. This method is ”top down”: it find all and only the information that was asked for. Without a predefined slot the target information is not retrieved. The output text is coherent and balanced unlike the extract generated by IR methods, which may be lacking in balance and cohesion as the sentences are quoted verbatim.

Both techniques can be applied to generate two different type of summaries:

- *Indicative Summaries* that suggest the contents of the document without providing specific detail. They can serve to entice the user into retrieving the full form. *Book jackets*, *card catalog entries* and *movie trailers* are examples of indicative summaries.

- *Informative Summaries* that represent (and often replace) the original document. Therefore it must contain all the pertinent information necessary to convey the core information and omit ancillary information.

Summaries based on IR models, usually, extract relevant passages for the target queries. To our knowledge no summarization approach use TC for summarization, even if the contrary has been experimented, e.g. [Kolcz *et al.*, 2001]. We introduce the concept of relevance with respect to a category. The indicative and informative summaries are extracted based on weighting schemes derived from the training data of the target category [Moschitti and Zanzotto, 2002]. In particular the indicative summaries are composed of the most relevant phrases, i.e., terminological expressions or other complex nominals. Chapter 4 shows that, these kind of summaries allow users to better understand the document content relatively to a predefined categorization scheme.

## 1.4 Thesis Outline

This thesis aims to study the interaction between Text Categorization and Natural Language processing. The reciprocal contribution of each other has been analyzed by measuring: (a) the improvement in accuracy that NLP techniques produce in TC and (b) the enhancement that TC models enable in NLP applications. The thesis is organized as follows:

- Chapter 2 describes the typical steps for designing a text classifier. In particular, several weighting schemes and the designing of profile-based classifiers are shown in detail. Additionally, the learning and classification algorithms for the Rocchio and the Support Vector Machine text classifiers are defined. The original contribution of this chapter relate to the definition of a novel inference method, Relative Difference Score and the Parameterized Rocchio text Classifier. This latter has been extensively experimented and compared using different corpora and different TC models.

- Chapter 3 reports the studies on the use of Natural Language Processing to extract linguistic feature for text categorization. Two main types of linguistically motivated features are studied: (a) those that use syntactic information, e.g., POS-tags and phrases and (b) those based on semantic information, i.e. the word senses. In particular, syntactic information has been divided in efficient, i.e. derivable via very fast algorithms and advanced that requires more complex models (that are usually more time consuming) to be detected. Extensive experimentation of such linguistic information on different corpora as well as on different models has been carried out.

- Chapter 4 proposes some novel use of TC for some sub-tasks of the most topical NLP applications, i.e., Information Extraction, Question Answering and Text Summarization. Preliminary experiments suggest that TC can improve the above NLP systems.

Finally, the conclusions can be found in Chapter 5.

# Chapter 2

# Text Categorization and Optimization

This chapter accurately describes the phases introduced in Section 1.1 concerning the designing and implementation of the TC models used in this thesis. Some new schemes for document weighting alternative to the *Inverse Document Frequency* as well as original profile based TC models have been proposed.

The major contribution of this chapter is the study on Rocchio classifier parameterization to achieve its maximal accuracy. The result is a model for the automatic selection of parameters. Its main feature is to bind the search space so that optimal parameters can be selected quickly. The space has been bound by giving a feature selection interpretation of the Rocchio parameters. The benefit of the approach has been assessed via extensive cross evaluation over three corpora in two languages. Comparative analysis shows that the performances achieved are relatively close to the best TC models (e.g., Support Vector Machines). The Parameterized Rocchio Classifier (PRC) [Basili and Moschitti, 2002; Moschitti, 2003b] maintains the high efficiency of the Rocchio model, thus it can be successfully applied in operational text categorization.

Corpora, weighing schemes, profile-based classification models, score adjustment techniques, inference policies, and performance measurements that are used in the experiments of this thesis have been defined respectively in sections 2.1, 2.2, 2.3 and 2.4. The two main TC models used in this research, Rocchio and Support Vector Machines, have been separately described and analyzed in Section 2.5. In Section 2.6 is shown how Rocchio classifier can be parameterized to enhance its accuracy. *Reuters-21578* has been used to compare the Rocchio, PRC and SVM accuracies in Section 2.7. Finally the conclusions are derived in Section 2.8.

## 2.1 Document Preprocessing

As it has been introduced in Section 1.1, in order to carry out the text classifier learning we need a sufficient number of labeled documents. Fortunately, for TC are available a lot of such resources as the categorization of information is widely used in press Companies as well as in scientific fields. In our experience, News Agencies and Medical fields are those more sensitive to the need of categorizing documents as we got from them the larger number of documents and corpora.

### 2.1.1 Corpora

In this thesis 6 different collections have been considered:

- The *Reuters-21578*[1] collection Apté split. It includes 12,902 documents for 90 classes, with a fixed splitting between *test-set* (here after $RTS$) and learning data $LS$ (3,299 vs. 9,603). As stated in Section 1.1.3 different Reuters versions [Yang, 1999; Sebastiani, 2002] have been used for testing TC algorithms. However, this version has been used for the most part of TC literature works. Thus, it can be considered as main referring TC collection. A description of some categories of this corpus is given in Table 2.1.

- The Reuters corpus, prepared by Y. Yang and colleagues[2], has been also used. It referred to as *Reuters3* versions [Yang, 1999]. It includes 11,099 documents for 93 classes, with a fixed splitting between test and learning data (3,309 vs. 7,789). The differences with the previous Reuters version are: (a) The split adopted is slightly different from the Apté ones and (b) Yang removed from it all not labeled documents. This explain as this last version contain 11,099 vs. 12,902 documents of the previous *Reuters-21578* versions. The removal of unlabeled corpus has prevented a direct comparison with other literature results. We noticed that the classifier accuracies (i.e., *Rocchio*, $PRC$ and $SVM$) on *Reuters-21578* are $\sim 1$ percent points below the performance obtained on *Reuters3*.

- Reuters news from TREVI project, collected in a set of about 26,000 documents, and distributed throughout 20 classes. Main topics of this corpus include specific areas like financial (e.g., *Corporate Industrial* or *Market/Share* news) as well as more general classes (e.g., *Sport* or *Elections*). These categories are very different from the *Acquisition*, *Crude* or *Cocoa* categories of the *Reuters-21578*. We will refer to the TREVI collection as the *TREVI-Reuters* corpus. This is the first *draft* release of Reuters Volume 1 recently made available by the Reuters company. In our experiments we have maintained the first level of the categorization schemes, i.e. the 20 main categories.

[1] Once available at `http://www.research.att.com/~lewis` and now available at `http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html`.

[2] Currently available at Carnegie Mellon University's web site through http://moscow.mt.cs.cmu.edu:8081/reuters 21450/apte.

- The ANSA collection, which includes 16,000 news items in Italian from the ANSA news agency. It makes reference to 8 target categories (2,000 documents each). ANSA categories relate to typical newspaper contents (e.g., Politics, Sport and Economics). It is worth noting that this last collection is closer to operational scenarios: some documents are not correctly assigned to the categories and several ones are repeated more than once. These problems affect almost all corpora but ANSA collection is particularly affected from document preparation errors. As an example, it is possible to find some English and German documents mixed with those in italian.

- The Ohsumed collection[3], including 50,216 medical abstracts. The first 20,000 documents, categorized under the 23 *MeSH diseases* categories, have been used in our experiments. The same subset of documents and categories has been used in [Joachims, 1998], thus, it possible to make a direct comparison with the results obtained in [Joachims, 1998]. Other used the Ohsumed collection for TC experiments, e.g., [Yang and Pedersen, 1997], but the employed document set and categories vary. However, literature results can give an indication of the magnitude order of the Ohsumed performance. For instance, from the fact that accuracy does not overcome 70% in all results obtained in different portion of Ohsumed, it possible to argue that this corpus is more *difficult* than Reuters, for which classifiers reaches 86% of accuracy. Table 2.2 gives a description of some categories used in the experiments.

- *HOS* (Health On-Line) news, a collection of short medicine-related abstracts. The *HOS* corpus is made of about 5,000 documents distributed throughout 11 classes. Typical classes are *Clinical Oncology* vs. *Endocrinology.* It is another example of real scenario corpus. HOS was part of TREVI project and provide us the documents to realize a TC system.

- The 20 Newsgroups[4] (20NG) corpus contains 19997 articles for 20 categories taken from the Usenet newsgroups collection. We used only the subject and the body of each message. Some of the newsgroups are very closely related to each other (e.g., *IBM computer system hardware / Macintosh computer system hardware*), while others are highly unrelated (e.g., *misc forsale / social religion and christian*). This corpus is different from Reuters and Ohsumed because it includes a larger vocabulary and words typically have more meanings. Moreover the stylistic writing is very different from the previous corpora as it referred to *e-mail* dialogues rather than technical summaries in Ohsumed or event reports in the News agencies.

The above corpora contain documents separated in several categories. The most usual approach to designing a classifier is, instead, to separate documents

[3] It has been compiled by William Hersh and it is currently available at `ftp://medir.ohsu.edu/pub/ohsumed`.

[4] Available at *http://www.ai.mit.edu/people/jrennie/20Newsgroups/*.

Table 2.1: Description of some Reuters categories

| Category | Description |
|---|---|
| *Acq* | Acquisition of shares and companies |
| *Earn* | Earns derived by acquisitions or sells |
| *Crude* | Crude oil events: market, Opec decision,.. |
| *Grain* | News about grain production |
| *Trade* | Trade between companies |
| *Ship* | Economic events that involve ships |
| *Cocoa* | Market and events related to Cocoa plants |
| *Nat-gas* | Natural Gas market |
| *Veg-oil* | Vegetal oil market |

Table 2.2: Description of some Ohsumed categories

| Category | Description |
|---|---|
| Pathology | Pathological Conditions |
| Cardiovascular | Cardiovascular Diseases |
| Immunologic | Immunologic Diseases |
| Neoplasms | Neoplasms |
| Digestive Sys. | Digestive System Diseases |
| Hemic & Lymph. | Hemic & Lymphatic Diseases |
| Neonatal | Neonatal Disorders & Abnormalities |
| Skin | Skin & Connective Tissue Diseases |
| Nutritional | Nutritional & Metabolic Diseases |
| Endocrine | Endocrine Diseases |
| Disorders | Disorders of Enviromental Origin |
| Animal | Animal Diseases |

in only two different sets: (1) positive documents that are categorized in the target class and (2) negative documents that are not categorized in it. Positive and negative documents are made available for the classifier designing in various forms. We have notice four main data structures:

- *The SMART format*, in which the document for all categories are given in a unique file. Headers allow to separate documents and to extract title, document *id* and the set of categories for target document. In this format there are available some IR SMART corpus as well as the Reuters and Ohsumed versions prepared by Yang.

- *The SGML format*, in which the tag pairs allow a more direct extraction of information, e.g., `<title>` and `<\title>`. This is the format provided for the Reuters Lewis version that includes the Apté split also.

- *The raw format*: this is the most usual structure in real scenario application. Users, like newspaper agencies, are not aware of data structure access and effective algorithm. They simply know that a set of documents belong to a target category. Thus, they usually produce a file containing all documents categorized for the target category. Documents are simply separated by one more empty line and if they belong to different classes (i.e. the multi-labeled documents), this information will be lost. In our research we have used several corpora of this type e.g., HOS or *TREVI-Reuters*.
- *The raw format per file*: each document is stored as a single file and each category is a directory containing all its document files. Even in this case an additional information source that indicates the multi-labeled documents, is needed. The 20 NewsGroups corpus is available in this format.

Whatever is the format of the training documents, the first step is to divide for each category the positive from negative documents, then the tokenization as well as the NLP module can be applied to both document sets.

### 2.1.2 Tokenization, Stoplist and Stemming

In this phase the relevant features are extracted from documents. As usual, all words as well as numbers are considered feasible features. They are, usually, called tokens. There are two possible way to form tokens:

(a) by selecting all character sequences separated by space. This means that alphanumeric strings like for example *Alex69* as well as more generic strings *tokenization_dir* are included in the resulting feature set.

(b) by considering alphabetic or numeric character sequences separated by all other characters. In this case the feature set contains only the usual words and numbers. The size of this feature set is lower than the set of the point (a).

In almost all our experiments we used the set derived in point (a), hereafter named $Tokens$.

After, the set of tokens is extracted it can be improved by removing features that do not bring any information. Function words (e.g., *What*, *Who*, *at*, *he* and *be*) are removed improving at least the efficiency of the target TC models. For this purpose a list of function words is prepared and used in the preprocessing phase as *stoplist*.

Other methods to improve the set of features consider that the same word is not ever used to describe the same concept (see Section 1.2). Thus the recall of the system can be enhanced by using automatic word associations. For this purpose there are language dependent methods like word stemming. Word stemming is based on two stages: suffix stripping and conflation. Suffix stripping can be achieved by using series of rules. For example biology, biologist, biologists reduce to biology. Some errors occur as there are always some exceptions from the rules. The error rate of word stemming has been measured around 5% [Van Rijsbergen, 1979]. Stemming, has been usually applied to the designing of a text classifier nevertheless, there is no study that proves the superiority of the stemmed word over the simple word sets.

Another important phase of TC pre-processing is the feature selection. As it is done for the stoplist a set of non-informative words are detected and removed from the feature set. The main difference with the stoplist technique is that the words to be removed are selected automatically.

### 2.1.3 Feature Selection

Feature Selection techniques have been early introduced in order to limit the dimensionality of the feature space of text categorization problems. The feature set cardinalities described in the previous section can be hundreds of thousands of elements. 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.

Automated feature selection methods envisage the removal of noninformative terms according to corpus statistics, and the construction of new (i.e. reduced or re-mapped) feature space. Common statistical selector parameters used in TC are: the *information gain*, the *mutual information*, the $\chi^2$ statistics and the document frequency ($DF$). As pointed out in [Yang and Pedersen, 1997] $DF$, $\chi^2$ and *information gain* provide the best selectors able to reduce the feature set cardinality and produce an increase in text classifier performances. The following equations describe four selectors among those experimented in [Yang and Pedersen, 1997]. They are based on both mutual information and $\chi^2$ statistics:

$$I_{max}(f) = \max_i\{I(f, C_i)\}$$

$$I_{avg}(f) = \sum_i P_r(C_i) \times I(f, C_i)$$

$$\chi^2_{max}(f) = \max_i\{\chi^2(f, C_i)\}$$

$$\chi^2_{avg}(f) = \sum_i P_r(C_i) \times \chi^2(f, C_i)$$

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 $\chi^2$ value[5] 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. For example, the selector of a feature by $I_{avg}$ applies the average function to the set of $I(f, C_i)$ scores: each dependence on the $i$-th class disappears resulting in one single ranking. The same is true for $\chi^2_{max}$ and $\chi^2_{avg}$.

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 relevance among classes. Classes that are more generic (e.g., whose values of $I(f, C_i)$ or $\chi^2$ 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, i.e. the number of truly irrelevant features is negligible.

Recently the previously referred techniques have been introduced even for selecting the relevant $n$-grams (see [Caropreso *et al.*, 2001]) in order to add informative features. It was confirmed that these extended features bring further information and often they increase performances of simple features. The problem is that $n$-grams impact on the ranking of other features. When selection is applied only a limited number of (i.e. top ranked) features is taken into account, so that important information may be lost. This happens as the applied methodology forces $n$-grams of a class taking the place of $n$-grams of another class in the ranking.

Other forms of feature selection are based on weighting schemes but they are used to weight features for the learning algorithm rather than to remove them.

## 2.2 Weighting Schemes

Weighting schemes are used in IR to determine which are the more relevant terms in documents and queries. This helps the IR system to rank the retrieved document depending on the expected relevance for the users. Traditionally, weights are heuristic combinations of different corpus statistics, *Term Frequency* and *Inverse Document Frequency*. The former quantity indicates the importance of a feature inside the document: if a word is repeated many time it should be important for that document. The latter quantity is used to assign a global importance: the more a term is frequent the less is its capacity of selecting topic information. Many variation have been studied in [Salton, 1989], the results are that different systems can benefit from the use of different weighting schemes.

In TC weighting schemes are less important even if their correct choice allows the classifier accuracy to be improved. With the aim to verify the above claim, next section describes two traditional weighting schemes as well as an original

[5] See [Yang and Pedersen, 1997] for a definition of $\chi^2$ score between features and categories.

one based on the *Inverse Word Frequency* that is very similar to the *Inverse Document Frequency*. Moreover, two weighting schemes to weight features inside categories[6] are presented.

### 2.2.1 Document Weighting

With the purpose of modeling our document weighting schemes, we need to define a few specific parameters. Given a target feature set $F = \{f_1, ..., f_N\}$ extracted from the *training-set*, a feature $f \in F$, a generic document $d$ of the corpus and the target set of classes $\mathcal{C} = \{C_1, C_2, ..., C_{|\mathcal{C}|}\}$, let the following notations express:

- $M$, the number of documents in the *training-set*,
- $M_f$, the number of documents in which the features $f$ appears and
- $o^d_f$, the occurrences of the features $f$ in the document $d$ ($TF$ of features $f$ in document $d$).

The first weighting scheme that we consider is the $IDF \times TF$, i.e., the traditional weighting strategy used in $SMART$ [Salton, 1991]. Given the $IDF(f)$ as $log(\frac{M}{M_f})$, the weight for the feature $f$ in the document $d$ is:

$$w^d_f = \frac{o^d_f \times IDF(f)}{\sqrt{\sum_{r \in F}(o^d_r \times IDF(r))^2}} \tag{2.1}$$

A second weighting scheme (used in [Ittner *et al.*, 1995]) is $log(TF) \times IDF$. It uses the logarithm of $o^d_f$ as follow:

$$l^d_f = \begin{cases} 0 & \text{if } o^d_f = 0 \\ log(o^d_f) + 1 & \text{otherwise} \end{cases} \tag{2.2}$$

Accordingly, the document weights is:

$$w^d_f = \frac{l^d_f \times IDF(f)}{\sqrt{\sum_{r \in F}(l^d_r \times IDF(r))^2}} \tag{2.3}$$

The third weighting scheme is referred to as $TF \times IWF$ and it introduces new corpus-derived[7] parameters:

- $O$, the overall occurrences of features,
- $O_f$, the occurrences of a feature $f$.

[6] They can be considered as macro-documents that contain all features of their documents.

[7] All these paramenters have to be learned from the documents in the *training-set* only.

By using the above statistics a new quantity, the $IWF$ [Basili *et al.*, 1999] (Inverse Word Frequency) can be defined as

$$IWF = log(\frac{O}{O_f})$$

$IWF$ is used similarly to $IDF$ in the following weighting:

$$w_f^d = \frac{o_f^d \times (IWF(f))^2}{\sqrt{\sum_{f \in F}(o_r^d \times (IWF(r))^2)^2}} \tag{2.4}$$

The above scheme has two major differences with respect to the traditional $TF \times IDF$ weighting strategy (i.e. Eq. 2.1). First, the *Inverse Word Frequency* [Basili *et al.*, 1999] is used in place of $IDF$. Its role is similar to $IDF$, as it penalizes very highly frequent (and less meaningful) terms (e.g., *say*, *be*, *have*) also recovering from systematic errors in POS tagging.

Another aspect is the adoption of $IWF$ squaring. In fact, the product $IWF \times o_f^d$ is too biased by the feature frequency $o_f^d$. In order to balance the $IWF$ contribution its square is thus preferred. A similar adjustment technique has been proposed in [Hull, 1994].

### 2.2.2 Profile Weighting

Once, the appropriate document weighting policy has been chosen, we can apply several methods to obtain the weights for the class profile. The simplest ones is just *Summing-up* for each features $f$ the weights it assumes in different documents of a class $C_i$ as follows:

$$W_f^i = \sum_{d \in P_i} w_f^d, \tag{2.5}$$

where $P_i$ is the set of training documents belonging to class $C_i$.

In this representation a profile is considered as a *macro document* made of all features contained in documents of the target class. Notice that the above model does not consider negative examples, i.e., the weights a feature assumes in other classes. On the contrary, another common weighting scheme attempting to better determine a profile weight, by using negative relevance, is the scheme provided by the *Rocchio's formula* [Rocchio, 1971]:

$$W_f^i = \max\left\{0, \frac{\beta}{|P_i|}\sum_{d \in P_i} w_f^d - \frac{\gamma}{|\bar{P}_i|}\sum_{d \in \bar{P}_i} w_f^d\right\} \tag{2.6}$$

where $\bar{P}_i$ is the set of documents not belonging to $C_i$. The feature weight $W_f^i$ in a profile is the difference between the sum of weights that $f$ assumes in the class $i$ and the sum of weights that $f$ assumes in documents of the other categories. Parameters $\beta$ and $\gamma$ control the relative impact of positive and negative examples on the classifier. The standard values used in literature (e.g., [Cohen and Singer, 1999; Ittner *et al.*, 1995]) are $\beta = 16$ and $\gamma = 4$. It is worth noticing that the *Summing-up* weighting scheme is a special case of the Rocchio's formula in which $\gamma$ is set to 0 and no normalization is applied. However, as the profiles created via *Summing-up* procedure (i.e. macro document building) are conceptually different from those designed by Rocchio's formula (i.e. centroid among positive and negative documents), we prefer maintain a diverse notation for referring them.

## 2.3 Similarity in profile-based Text Categorization

After both the document and profile weights have been defined their vector representations is as follows:

$$\vec{d} =< w^d_{f_1}, ..., w^d_{f_N} >$$

$$\vec{C_i} =< W^i_{f_1}, ..., W^i_{f_N} >$$

Given $\vec{C_i}$ and $\vec{d}$ representations a similarity function that computes the distance in the vector space can be defined. This completes the metric on Vector Space Model. In all our experiments we apply the usual cosine measure:

$$s_{id} = cos(\vec{C_i}, \vec{d}) = \sum_{f \in F} W^i_f w^d_f \tag{2.7}$$

When weighting schemes are applied to training corpus some problems arise as scores produced by the test documents may not be comparable among different classes. They can refer to very different distributions because of the different training evidences.

Weighting formula can be characterized by a large variance across class profiles. The undesired consequence is a very odd distribution of scores obtained by Eq. 2.7 through the different categories. Scores can be thus not comparable across classes. Those decision methods that make use of a single threshold for all classes are weak or even inapplicable. The same can be said of methods adopting a single ranking among the scores even when they originate from different classes.

In order to tackle this problem some techniques have been proposed that change the vector space by rescaling the scores and projecting them in subspaces. This phase is often applied without a specific naming. It will be hereafter referred to as *score adjustment.* Score adjustment is needed to project the similarity function in a unifying space better suited for representing all the classes. Two effective adjustment methods have been proposed and will be discussed in the next sections.

### 2.3.1 Similarity based on Logistic Regression

An attempt to carry out score adjustment is the application of Logistic Regression (*LR*). When *LR* is applied to scores $s_{di}$ an actual estimate of $P(C_i|d)$, i.e. the probability that a document $d$ belong to the class $C_i$, is obtained. This idea has been firstly introduced in [Ittner *et al.*, 1995]. In brief, the *LR* score adjustment algorithm works as follows.

- First all the pairs $<s_{di}, belong\_flag>$ for each training document $d$ and for each fixed class $i$ are evaluated: $belong\_flag$ is set to 1 *iff* $d \in C_i$, and to 0 otherwise.
- The derived pairs are then input to the Logistic Regression algorithm. Two parameters $\alpha_i$ and $\beta_i$ are produced. $\alpha_i$ and $\beta_i$ are set such that $P(C_i|s_{di})$ can be estimated via the logistic function [Ittner *et al.*, 1995]:

  $$F(\alpha_i, \beta_i, s_{di}) = \frac{e^{\alpha_i + \beta_i \times s_{di}}}{1 + e^{\alpha_i + \beta_i \times s_{di}}}$$

  This is a good approximation of $P(C_i|d)$, that is, $\alpha_i$ and $\beta_i$ are estimated such that $P(C_i|d) \simeq F(\alpha_i, \beta_i, s_{di})$. The *LR* function thus produces the conditional probability $P(d \in C_i|s_{di})$.
- Finally, after each class $i$ is assigned with coefficients $\alpha_i$ and $\beta_i$, the final classification is taken over images of similarity scores $P(C_i|s_{di}) \simeq F(\alpha_i, \beta_i, s_{di})$.

Any of the inference strategy can be here applied as the $P(C_i|s_{di})$ are distributed throughout all the classes, $C_i$, better than the source values $s_{di}$. It is worth noticing that the logistic function is monotonic ascending. This implies that when we fix a class $C_i$ the ranking of documents according to $P(C_i|d)$ or to $s_{di}$ does not change.

### 2.3.2 Similarity over differences: Relative Difference Scores

The *LR* score adjustment method allows to consistently rank scores originated from different classes and this may greatly improve the system overall performance. This is especially true for text classifiers based on the *Summing-up* weighting scheme (Eq. 2.5) that does not use negative examples. In fact, the score adjustment allows to better compare scores $s_{di}$ of different categories and to retrieve "odd" test documents showing lower similarity scores with profiles of all classes $C_i$ (i.e., given a document $d$, $s_{di} << 1$ for each category $i$).

However *LR* does not help to better rank documents within a single target class. This is an inherent weakness. As an example, let us imagine a situation where a unique threshold is applied to all the test documents and we have two classes and three documents described as in Table 2.3.

A document ($d_2$) is odd as it shows a low similarity with both the two classes. The other two documents, $d_1$ and $d_3$, should be accepted as members of class

Table 2.3: Scores for a simple Classification Inference case

| Document Index | $Class_1$ (score $s_{d,1}$) | $Class_2$ (score $s_{d,2}$) | *Gold Standard* |
|---|---|---|---|
| $d_1$ | 7 | 1 | $Class_1$ |
| $d_2$ | 0.01 | 0.8 | $Class_2$ |
| $d_3$ | 2 | 5 | $Class_2$ |

$C_1$ and $C_2$ respectively. Notice that in this unfortunate case, the *Scut* inference policy should discard classifications whose scores $s_{id}$ are below 5. This would prevent to accept document $d_2$ in class $C_2$, although its scores are such that $s_{12}$ is about eighty times lower than $s_{22}$! What we need is a technique able to produce a ranking among documents influenced by their general behavior, according to their similarity with respect to all classes. If we could re-rank documents according to this cross-categorical information we would have a ranking for the class $C_2$ like, $d_3 \succeq d_2 \succeq d_1$. This has to violate the monotonicity of the *LR* function (as $s_{21} > s_{22}$).

To overcame this problem we have defined a score adjustment technique based on the *differences among similarity scores* capable to project the similarity function image into a different set whose natural order better reflects the current document ranking. Instead of the $s_{di}$ scores, a slightly more complex score $m_{di}$ is used: it expresses the average difference between the score of the correct (e.g., $i$-th) class and the remaining classes. Formally, given a training document $d$ and a class $C_i$, $m_{di}$ is estimated by:

$$m_{di} = \frac{\sum_{j=1}^{|\mathcal{C}|} s_{di} - s_{dj}}{|\mathcal{C}| - 1} \qquad (2.8)$$

Equation 2.8 is the score adjustment methodology that we call *RDS* (see [Basili *et al.*, 2000a; 2000b] for more details). Notice that in the simple case defined in Table 2.3, the following values are obtained: $m_{21} = -6$, $m_{22} = 0.79$ and $m_{23} = 3$ correctly suggesting the expected ranking $d_3 \succeq d_2 \succeq d_1$ for the class 2. *RDS* produces scores that explicitly depend on the negative information expressed by documents not belonging to a target class in the *training-set.* A study of its positive effects on classifier accuracy is reported in Chapter 3.

## 2.4 Inference Policies and Accuracy Evaluation

When scores estimating the similarity between a newly incoming document $d$ and the different profiles are available, the acceptance/rejection of the different categories $C_i$ can be decided. The decision function $\phi$ can be defined now only in terms of similarity scores ($s_{di}$), i.e. as a $k$-ary real-valued function $\phi : \Re^k \rightarrow 2^{\{C_1,\ldots,C_{|\mathcal{C}|}\}}$. As $\phi$ is applied to a set of documents (e.g., the *test-set*) two different groupings of scores $s_{di}$ are possible depending on classes (index

$i = 1, ..., |\mathcal{C}|$) or documents (index $d$ such that $d \in TS$). The two cases are defined as:

- $k = |\mathcal{C}|$: the set of scores that one target document $d$ assumes in all the $|\mathcal{C}|$ categories, i.e. $\{s_{d1}, ..., s_{d|\mathcal{C}|}\}$. This is referred to the *document pivoted classification scheme* [Sebastiani, 2002].
- $k = |TS|$: the set of scores that *all documents* in $TS$ assume wrt the target category $C_i$, i.e. $\{s_{1i}, s_{2i}, ..., s_{|TS|i}\}$. This is referred to the *category pivoted classification scheme* [Sebastiani, 2002].

The accuracy of $\phi$ can be measured as the correct categories for documents in $TS$ are available. Let us refer to such correct choices as the gold standard $GS(TS)$. The differences between the outcome $\phi(d)$ and the categories suggested by the $GS(d)$ is usually measured by one or more numeric values. It is obtained by counting the number of *correct*, *wrong*[8] categories $\phi(d)$ wrt $G(d)$. Next sections will give details both on possible inference policies embodied by $\phi$ as well as on the definition of accuracy indexes.

### 2.4.1 Inference Policies

A decision function $\phi$ has to select categories that have the highest scores in the score groups (e.g., $\{s_{d1}, ..., s_{|\mathcal{C}|}\}$). This is usually carried out by imposing thresholds according to one of the following strategies [Yang, 1999]:

- probability threshold ($Scut$): for each class $C_i$ a threshold $\sigma_i$ is adopted such that $C_i \in \phi(d)$ only if its membership score $s_{di}$ is over $\sigma_i$ (*category pivoted classification scheme*). The threshold $\sigma_i$ is an upper limit to the risk of misclassification and has a probabilistic nature: it measures the average number of potential misclassifications under a given assumption on the distribution.
- fixed threshold ($Rcut$): It is based on the assumption that $k$ is the average number of classes valid for a generic document $d$. This can be observed usually over the *training-set*. Accordingly, $C_i \in \phi(d)$ only if $C_i$ is one of the first $k$ classes in the ranking obtained via the $s_{di}$ positive scores (*document pivoted classification scheme*).
- proportional threshold ($Pcut$): the threshold is the percentage $prob$ of documents that are to be categorized under $C_i$ (*category pivoted classification scheme*). It is usually estimated from the *training-set* $T$, i.e. $prob(C_i|T)$.

### 2.4.2 Accuracy Measurements

Several measures of performance have been proposed in TC each one with inherent advantages and disadvantages as well. The *error rate* is the ratio between

[8] Notice that an empty set of values output by $\phi(d)$ corresponds to *don't know* choices, i.e. no category is provided for $d$.

the number of documents not correctly categorized and the total number of documents. According to the above definition, if the *test-set* includes a small percentage of documents labeled under a given category, a trivial classifier that refuses all documents of that category will obtain a very low error rate (i.e. a good performance), at least with respect to that category. Two other measures, i.e. *precision* and *recall*, are not affected by such limitation. Given a specific category $C_i$, their technical definition can be somehow informally stated in terms of three scores:

- the correct categories found by the decision function, $CFC_i$, i.e. the number of times $C_i \in \phi(d)$ **and** $C_i \in GS(d)$ for all $d \in TS$.
- the total number of correct categories, $TCC_i$, i.e. the number of times $C_i \in GS(d)$ for all $d \in TS$.
- the total number of system choices, $TCF_i$, i.e. the number of times $C_i \in \phi(d)$ for all documents $d \in TS$.

In synthesis $CFC_i$ is the number of correct system decisions over $C_i$, $TCC_i$ is the number of correct assignments of $d \in TS$ to $C_i$ and $TCF_i$ is the total number of system acceptances. Notice that $TCC_i$ should overlap as much as possible with $TCF_i$ to converge towards a perfect discriminating function. The *recall* and *precision* scores can be thus defined respectively as follows:

$$Recall_i = \frac{CFC_i}{TCC_i} \tag{2.9}$$

$$Precision_i = \frac{CFC_i}{TCF_i} \tag{2.10}$$

Both the measurements depend on the thresholds (as discussed in the previous section) but they are in general inversely proportional. When a threshold (e.g., $\sigma_i$) increases, the precision increases while the recall tends to decrease and vice versa. This variability between recall and precision makes it difficult to compare different classifiers just according to different *(precision, recall)* pairs[9]. In order to get a single performance index, the *Breakeven point* (BEP) is widely adopted. The BEP is the point in which recall and precision are equal. It is estimated iteratively by increasing the threshold from 0 to the highest value for which $precision <= recall$. The major problem is that the correct BEP score could not exist (i.e. for no value of the threshold $recall = precision$). In this case, a conclusive estimation is the mean between the *recall* and *precision* (interpolated BEP) at the best estimated threshold value. However, even this may result artificial [Sebastiani, 2002] when *precision* is not enough *near* to *recall*.

The $f_1$-measure improves the BEP definition by imposing the harmonic mean between *precision* and *recall* as follows:

$$f_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{2.11}$$

[9] A classifier could reach an high *recall* while another could achieve an even higher *precision*, the superiority of which is difficult to establish.

$f_1$ outputs a more reliable value especially when *recall* is highly *different* from *precision*. For example, with a *precision* of .9 and a *recall* of .001 (i.e. the pair of the nearest values obtained by threshold adjustment) simple average is .45 while $f_1$=0.002 corresponds to a more realistic performance indication.

In our experiments a *validation-set*[10] is used to tune the thresholds associated to the maximal BEP. Threshold adjustments are first carried out and then the detected thresholds determine the performance measured over the (separate) *test-set*. For some experiments we reports the interpolated BEP as it also used in previous literature TC evaluations , e.g., [Yang, 1999; Joachims, 1998; Lewis and Gale, 1994; Apté *et al.*, 1994; Lam and Ho, 1998].

Finally, as our target classification problem involves more than one category, we used a binary classifier[11] for each category. The global measure derived from the classifier pool is the *microaverage*. According to definitions given in 2.9 and Eq. 2.10, the equations 2.12 and 2.13 define the *microaverage* of *recall* and the *microaverage* of *precision* for $|\mathcal{C}|$ binary classifiers.

$$\mu Recall = \frac{\sum_{i=1}^{|\mathcal{C}|} CFC_i}{\sum_{i=1}^{|\mathcal{C}|} TCC_i} \tag{2.12}$$

$$\mu Precision = \frac{\sum_{i=1}^{|\mathcal{C}|} CFC_i}{\sum_{i=1}^{|\mathcal{C}|} TCF_i} \tag{2.13}$$

The above measures are then used to evaluate the *microaverage* of both BEP and $f_1$ , i.e.

$$\mu BEP = \frac{\mu Precision + \mu Recall}{2} \tag{2.14}$$

$$\mu f_1 = \frac{2 \times \mu Precision \times \mu Recall}{\mu Precision + \mu Recall} \tag{2.15}$$

## 2.5 Support Vector Machines and Rocchio Classifier

One of the aim of our study is to measure the impact of richer document representations on TC. Such representations could produce different results on different TC approaches such as *Decision Trees*, *k-Nearest Neighbor heuristics*, *probabilistic frameworks*, *Disjunctive Normal Form rules* and *neural architectures* (see [Sebastiani, 2002] for a survey on the subject). Thus, the choice of some representative models is not trivial. The idea is that a richer representation can be really useful only if: (a) it produces an increase of the target classifier accuracy, that overcomes all other models, fed with the simple *bag-of-words* or

[10] A separate portion of the *training-set* used for parameterization purposes

[11] A *binary classifier* is a decision function that assigns or rejects a unique category $C_i$ to an input document.

(b) it allows the accuracy of a very efficient classifier (in term of time complexity) to be close to the best figure classifier. In both cases an improvement of the *state-of-the-art* will be obtained in accuracy or efficiency.

In this perspective, we have adopted two different TC approaches: Rocchio [Ittner *et al.*, 1995] and SVM [Vapnik, 1995] classifiers. The former is a very efficient TC, so, it would be very appealing (especially for real scenario applications) to bring its accuracy *near* the best figure classifier. The second is one of the best figure TC, consequently, improving it causes an improvement of the *state-of-art*.

### 2.5.1 The Classification Function

Rocchio and SVM are based on the Vector Space Model. Again the document $d$ is described as a vector $\vec{d} =< w^d_{f_1}, .., w^d_{f_N} >$ in a $N$-dimensional vector space. The axes of the space, $f_1, .., f_N$, are the features extracted from the training documents and the vector components $w^d_{f_j} \in \Re$ are the weights evaluated as described in Section 2.2.

Rocchio and SVM learning algorithm use the vector representations to derive a hyperplane, $\vec{a} \times \vec{d} + b = 0$, that separates the positive from negative document vectors in the *training-set*. More precisely, $\forall \vec{d}$ positive examples, $\vec{a} \times \vec{d} + b \geq 0$, otherwise $\vec{a} \times \vec{d} + b < 0$. $\vec{d}$ is the equation variable, while the gradient $\vec{a}$ and the 0-intersect $b$ are determined by the target learning algorithm. Once the above parameters are available, it is possible to define the associated classification function, $\phi : D \rightarrow \{C, \emptyset\}$, from the set of documents $D$ to the binary decision (i.e., belonging or not to $C$). Such decision function is described by the following equation:

$$\phi(d) = \begin{cases} C & \vec{a} \times \vec{d} + b \geq 0 \\ \emptyset & otherwise \end{cases} \tag{2.16}$$

Eq. 2.16 shows that a category is accepted only if the product $\vec{a} \times \vec{d}$ overcomes the threshold $b$. This suggests that the hyperplane gradient $\vec{a}$ can be considered as a category profile, the scalar product is adopted to measure the similarity between profile and document, and $b$ is the threshold for the *Scut* policy, described in Section 2.4.1.

Thus, Rocchio and SVM are characterized by the same decision function[12]. Their difference is the learning algorithm to evaluate the threshold $b$ and the profile $\vec{a}$ parameters: the former uses a simple heuristic while the second solves an optimization problem.

### 2.5.2 Rocchio Learning

The learning algorithm of the Rocchio text classifier is the simple application of the Rocchio's formula (Eq. 2.6) presented in Section 2.2.2. The parameters

[12] This is true only for linear SVM. In the polynomial version the decision function is a polynomial of support vectors.

$\vec{a}$ is evaluated by the equation:

$$\vec{a}_f = \max\left\{0, \frac{\beta}{|P|}\sum_{d\in P} w_f^d - \frac{\gamma}{|\bar{P}|}\sum_{d\in \bar{P}} w_f^d\right\} \tag{2.17}$$

Eq. 2.17 shows that the components of the hyperplane gradient $\vec{a}$ are the weights assumed by the feature $f$ in the profile of $C$ and the 0-intersect $b$ is the threshold. This latter can be estimated by picking-up the value that maximizes the classifier accuracy on a training subset called *evaluation-set*.

The above learning algorithm is based on a simple heuristic that does not ensure the best separation of the training documents. Thus, the accuracy reflects the weakness of the approach. However, the simplicity of the learning algorithm makes the resulting TC system one of the best efficient ones.

### 2.5.3 Support Vector Machine learning

The major advantage of SVM model is that the parameters $\vec{a}$ and $b$ are evaluated applying the *Structural Risk Minimization principle* [Vapnik, 1995], stated in the statistical learning theory. The main feature of the above principle is that the probability $P(\phi(d) = C|d \in \bar{P})$ of a classifier $\phi$ will make an error is bounded by the following quantity:

$$e_0 + 2\sqrt{\frac{vc(ln\frac{2m}{vc}+1) - ln\frac{M}{4}}{M}} \tag{2.18}$$

Where $e_0$ is the error over the training set, $M$ is the number of training examples and $vc$ is the *VC-dimension*[13] [Vapnik, 1995] that depends on the classifier. The SVMs are chosen in a way that $|\vec{a}|$ is minimal. More precisely the parameters $\vec{a}$ and $b$ are a solution of the following optimization problem:

$$\begin{cases} Min \quad |\vec{a}| \\ \vec{a} \times \vec{d} + b \geq 1 \quad \forall d \in P \\ \vec{a} \times \vec{d} + b \leq -1 \quad \forall d \in \bar{P} \end{cases} \tag{2.19}$$

It can be proven that the minimum $|\vec{a}|$ leads to a maximal margin[14] (i.e. distance) between negative and positive examples.

In summary, SVM actually divides the positive from negative examples of the *training-set* and it attempts to make the best separation to reduce the probable error on *test-set*. Rocchio classifier enables the separation using a simple heuristic that does not ensure the best separation, at all. However, the notion of profile is better suited for the human interpretation of Text Categorization

[13] Technically the *VC dimension* is the maximal number of training points that can be divided in all possible bi-partitions by using linear functions (in our case).

[14] The software to carry out both the learning and classification algorithm for $SVM$ are described in [Joachims, 1999] and they have been downloaded from the web site *http://svmlight.joachims.org/*.

(i.e. it is possible to build such profiles manually). On one hand, SVM better exploits the indexing property of the feature set used, on the other hand Rocchio algorithm is nearer to the manual processing. This last property makes simpler the introduction in the model of more complex linguistic feature such as *proper nouns*, *complex nominals* or other conceptual information.

In next section, we present a parameter estimation method that allows Rocchio classifier to improve its $f_1$ measure at least of 4/5 percent points.

## 2.6 The Parameterized Rocchio Classifier

Machine learning techniques applied to text categorization (TC) problems have produced very accurate although computationally complex models. In contrast, systems of real scenario such as Web applications and large-scale information management necessitate fast classification tools. Accordingly, several studies (e.g., [Chuang *et al.*, 2000; Drucker *et al.*, 1999; Gövert *et al.*, 1999]) on improving accuracy of low complexity classifiers have been carried out. They are related to the designing of efficient TC models in Web scenarios: feature space reduction, probabilistic interpretation of $k$-Nearest Neighbor and hierarchical classifiers are different approaches for optimizing speed and accuracy.

In this perspective, there is a renewed interest in the Rocchio formula. Models based on it are characterized by a low time complexity for both training and operative phases. The Rocchio weakness in TC application is that its accuracy is often much lower than other more computationally complex text classifiers [Yang, 1999; Joachims, 1998].

In order to improve the Rocchio accuracy we have study a method to derive an optimal parameterization. The parameters of Rocchio formula (Eq. 2.17) are $\beta$ and $\gamma$. They control the relative impact of positive and negative examples and determine the weights of the features $f$ in the target profile. The setting used for any IR application was $\beta = 16$ and $\gamma = 4$. It was also used for the categorization task of low quality images [Ittner *et al.*, 1995]. However, neither a methodology nor a theoretical justification was followed to derive that setting. In [Cohen and Singer, 1999] has been pointed out that these parameters greatly depend on the training corpus and different settings produce a significant variation in performances. Recently, some researchers [Singhal *et al.*, 1997b] have found that $\gamma = \beta$ is a good parameters choice, but, again a systematic methodology for parameter setting were not definitively proposed.

In [Schapire *et al.*, 1998] Rocchio standard classifier has been shown to achieve the *state-of-the-art* performances, although its efficiency is penalized. Improvements in accuracy were produced by using more effective weighting schemes and *query zoning* methods, but a methodology for estimating Rocchio parameters was not considered.

Thus, the literature confirms the need of designing a methodology that automatically derives optimal parameters. Such a procedure should search parameters in the set of all feasible values. As no analytical procedure is available for deriving optimal Rocchio parameters, some heuristics are needed to limit the search space. Our idea to reduce the search space is to consider the feature selection property of the Rocchio formula. We will show that:

1. The setting of Rocchio parameters can be reduced to the setting of the ratio between parameters.
2. Different values for the ratio induce the selection of feature subsets.
3. Only the features in the selected subset affect the accuracy of Rocchio classifier parameterized with the target parameter rate.
4. The parameter rate is inversely-proportional to the cardinality of the feature subset.

Therefore, increasing the parameter ratio produces a subset collection of decreasing cardinality. Rocchio classifier, trained with these subsets, outcomes different accuracies. The parameter ratio seems affect accuracy in the same way a standard feature selector [Kohavi and John, 1997] would do. From this perspective, the problem of finding optimal parameter ratio can be reduced to the feature selection problem for TC and solved as proposed in [Yang and Pedersen, 1997]. Next section describes in details the adopted method.

### 2.6.1 Search space of Rocchio parameters

As claimed in the previous section, to improve the accuracy of the Rocchio text classifier, parameter tuning is needed. The exhaustive search of optimal values for $\beta$ and $\gamma$ is not a feasible approach as it requires the evaluation of Rocchio accuracy for all the pairs in the $\Re^2$ space.

To reduce the search space, we notice that not both $\gamma$ and $\beta$ parameters are needed as $\beta$ can be bound to the threshold parameter. The classifier accepts a document $d$ in a category $C$ if the scalar product between their representing vectors is greater than a threshold $\sigma$, i.e. $\vec{C} \times \vec{d} \geq \sigma$. Substituting $\vec{C}$ with the original Rocchio's formula we get:

$$\left( \frac{\beta}{|P|} \sum_{d' \in P} \vec{d'} - \frac{\gamma}{|\bar{P}|} \sum_{d' \in \bar{P}} \vec{d'} \right) \times \vec{d} \geq \sigma$$

and dividing by $\beta$,

$$\left( \frac{1}{|P|} \sum_{d' \in P} \vec{d'} - \frac{\gamma}{\beta |\bar{P}|} \sum_{d' \in \bar{P}} \vec{d'} \right) \times \vec{d} \geq \frac{\sigma}{\beta} \Rightarrow \left( \frac{1}{|P|} \sum_{d' \in P} \vec{d'} - \frac{\rho}{|\bar{P}|} \sum_{d' \in \bar{P}} \vec{d'} \right) \times \vec{d} \geq \sigma'.$$

Once $\rho$ has been set, the threshold $\sigma'$ can be automatically assigned by the algorithm that evaluates the BEP. Note that, to estimate the threshold from a *validation-set*, the evaluation of BEP is always needed even if we maintain both parameters. The new Rocchio formula is:

$$\vec{a}_f = \max \left\{ 0, \frac{1}{|P|} \sum_{d \in P} w_f^d - \frac{\rho}{|\bar{P}|} \sum_{d \in \bar{P}} w_f^d \right\} \qquad (2.20)$$

where $\rho$ represents the *ratio* between the original Rocchio parameters, i.e. $\frac{\gamma}{\beta}$.

Our hypothesis for finding *good* $\rho$ value is that it deeply depends on the differences among classes in term of document contents. This enables the existence of different optimal $\rho$ for different categories. If a correlation function between the category similarity and $\rho$ is derived, we can bound the search space.

We observe that in Equation 2.20, features with negative difference between positive and negative weights are set to 0. This aspect is crucial since the 0-valued features do not contribute in the similarity estimation (i.e. they give a null contribution to the scalar product). Thus, the Rocchio model does not use them. Moreover, as $\rho$ is increased *smoothly*, only the features having a *high* weight in the negative documents will be eliminated (they will be set to 0 value). These features are natural candidates to be irrelevant for the Rocchio classifier. On one hand, in [Kohavi and John, 1997; Yang and Pedersen, 1997] it has been pointed out that classifier accuracy can improve if irrelevant features are removed from the feature set. On the other hand, the accuracy naturally decreases if relevant and some weak relevant features are excluded from the learning [Kohavi and John, 1997]. Thus, by increasing $\rho$, irrelevant features are removed until performance improves to a maximal point, then weak relevant and relevant features start to be eliminated, causing Rocchio accuracy to decrease. From the above hypothesis, we argue that:

*The best setting for $\rho$ can be derived by increasing it until Rocchio accuracy reaches a maximum point.*

In Section 2.7, experiments show that the Rocchio accuracy has the above behavior. In particular, the $\rho$/accuracy relationship approximates a convex curve with a single max point.

An explanation of linguistic nature could be that a target class $C$ has its own specific set of terms (i.e. features). We define *specific-terms* as the set of words typical of one domain (i.e. very frequents) and at the same time they occur infrequently in other domains. For example, *byte* occurs more frequently in a *Computer Science* category than a *Political* one, so it is a *specific-term* for *Computer Science* (with respect to the *Politic* category).

The Rocchio formula selects *specific-terms* in $C$ also by *looking* at their weights in the other categories $C_x$. If the negative information is emphasized enough the *non specific-terms* in $C$ (e.g., terms that occur frequently even in $C_x$) are removed. Note that these *non specific-terms* are misleading for the categorization. The term *byte* in political documents is not useful for characterizing the political domain. Thus, until the *non specific-terms* are removed, the accuracy increases since noise is greatly reduced. On the other hand, if negative information is too much emphasized, some *specific-terms* tend to be eliminated and accuracy starts to decrease. For example, *memory* can be considered *specific-terms* in *Computer Science*, nevertheless it can appears in *Political* documents; by emphasizing its negative weight, it will be finally removed, even from the *Computer Science* profile. This suggests that the specificity of terms in $C$ depends on $C_x$ and it can be captured by the $\rho$ parameter.

In the next section a procedure for parameter estimation of $\rho$ over the *training-set* is presented.

### 2.6.2 Procedure for parameter estimation

We propose an approach that takes a set of training documents for profile building and a second subset, the *estimation-set*, to find the $\rho$ value that optimizes the Breakeven Point. This technique allows parameter estimation over data independent of the *test-set* ($TS$), and the obvious bias due to the training material is avoided as widely discussed in [Kohavi and John, 1997]. The initial corpus is divided into a first subset of training documents, called *learning-set* $LS$, and a second subset of documents used to evaluate the performance, i.e. $TS$.

Given the target category, estimation of its optimal $\rho$ parameter can be carried out according to the following *held-out* procedure:

1. A subset of $LS$, called *estimation set* $ES$ is defined.
2. Set $j = 1$ and $\rho_j =$ Init_value.
3. Build the category profile by using $\rho_j$ in the Eq. 2.20 and the *learning-set* $LS - ES$.
4. Evaluate the $BEP_j$ for the target classifier (as described in Section 2.4.2) over the set $ES$.
5. Optionally: if $j > 1$ and $BEP_{j-1} \geq BEP_j$ go to point 8.
6. if $\rho_j >$ Max_limit go to point 8.
7. Set $\rho_{j+1} = \rho_j + \Delta\rho$, $j = j + 1$ and go to point 3.
8. Output $\rho_k$, where $k = argmax_j(BEP_j)$.

The minimal value for $\rho$ (i.e. the Init_value) is 0 as a negative ratio makes no sense in the feature selection interpretation. The maximal value can be derived considering that: (a) for each $\rho$, a different subset of features is used in the Rocchio classifier and (b) the size of the subset decrease by increasing $\rho$. Experimentally, we have found that $\rho = 30$ corresponds to a subset of 100 features out of 33,791 initial ones for the *Acquisition* category of the Reuters Corpus. The above feature reduction is rather aggressive as pointed out in [Yang and Pedersen, 1997] so, we chose 30 as our maximal limit for $\rho$.

However, in the feature selection interpretation of $\rho$ setting, an objective maximal limit exists: it is the value that assigns a null weight to all features that are also present in the negative examples. This is an important result as it enables the automated evaluation of the maximum $\rho$ limit on training corpus in a linear time. It can be obtained by evaluating the ratio between the negative and the positive contributions in Eq. 2.20 for each feature $f$ and by taking the maximum value. For example we have found a value of 184.90 for the *Acquisition* category.

The values for $\Delta\rho$ also (i.e. the increment for $\rho$) can be derived by referring to the feature selection paradigm. In [Yang and Pedersen, 1997; Yang, 1999; Joachims, 1998] the subsets derived in their feature selection experiments have a decreasing cardinality. They start from the total number of unique features $N$ and then select $N - i \times h$ features in the $i$-th subset; $h$ varies between 500 and 5,000. When $\Delta\rho = 1$ is used in our estimation algorithm, subsets of similar sizes are generated. Moreover, some preliminary experiments have suggested that smaller values for $\Delta\rho$ do not select better $\rho$ (i.e., they do not produce better Rocchio accuracy).

A more reliable estimation of $\rho$ can be applied if steps 2-8 are carried out according to different, randomly generated splits $ES_k$ and $LS - ES_k$. Several values $\rho(ES_k)$ can thus be derived at step $k$. A resulting $\bar{\rho}$ can be obtained by averaging the $\rho(ES_k)$. Hereafter we will refer to the Eq. 2.20 parameterized with estimated $\rho$ values as the *Parameterized Rocchio Classifier* ($PRC$).

### 2.6.3 Related Work

The idea of parameter tuning in the Rocchio formula is not completely new. In [Cohen and Singer, 1999] it 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, a procedure for their estimation was not proposed as the parameters chosen to optimize the classification accuracy over the training documents were, in general, different from those optimizing the *test-set* classification. A possible explanation is that the searching in parameter space was made at random: a group of values for parameters was tried without applying a specific methodology. Section 2.7.2 shows that, when a systematic parameter estimation procedure is applied (averaging over a sufficient number of randomly generated samples), a reliable setting can be obtained.

Another attempt to improve Rocchio classifier has been provided via probabilistic analysis in [Joachims, 1997]. A specific parameterization of the Rocchio formula based on the $TF \times IDF$ weighting scheme is proposed. Moreover, a theoretical explanation within a vector space model is provided. The equivalence between the probability of a document $d$ in a category $C$ (i.e. $P(C|d)$) and the scalar product $\vec{C} \times \vec{d}$ is shown to hold. This equivalence implies that the following setting for the Rocchio parameters: $\gamma = 0$ and $\beta = \frac{|C|}{|D|}$, where $|D|$ is the number of corpus documents. It is worth noting that the main assumption, at the basis of the above characterization, is $P(d|w, C) = P(d|w)$ (for words $w$ descriptors of $d$). This ensures that $P(C|d)$ is approximated by the expectation of $\sum_{w \in d} P(C|w)P(w|d)$. The above assumption is critical as it assumes that the information brought by $w$ subsumes the information brought by the pair $<w, C>$. This cannot be considered generally true. Since the large scale empirical investigation, carried out in Section 2.7, proves that the relevance of negative examples (controlled by the $\gamma$ parameter) is very high, the approach in [Joachims, 1997] (i.e., $\gamma = 0$) cannot be assumed generally valid.

In [Singhal *et al.*, 1997b; 1997a] an enhanced version of the Rocchio algorithm has been designed for the problem of document routing. This task is a different instance of TC. The concept of category refers to the important document for a specific query. In that use of the Rocchio's formula, $\beta$ parameter cannot be eliminated as it has been in Section 2.6.1. Moreover, an additional parameter $\alpha$ is needed. It controls the impact of the query in routing the relevant documents. The presence of three parameters makes difficult an estimation of a good parameter set. The approach used in [Singhal *et al.*, 1997b] is to try a number of values without a systematic exploration of the space. The major drawback is that the selected values could be only the local max of some document sets. Moreover, no study was done about the parameter variability. A set of values that maximize Rocchio accuracy on a *test-set* could minimize the performance over other document sets.

In [Schapire *et al.*, 1998] an enhanced version of Rocchio text classifier has been designed. The Rocchio improvement is based on better *weighting schemes* [Singhal *et al.*, 1995], on *Dynamic Feedback Optimization* [Buckley and Salton, 1995] and on the introduction of *Query Zoning* [Singhal *et al.*, 1997b]. The integration of the above three techniques has shown that Rocchio can be competitive with state-of-the art filtering approaches such as *Adaboost.* However, the problem of parameter tuning has been neglected. The simple setting $\beta = \gamma$ is adopted for every category. The justification given for such choice is that the setting has produced good results in [Singhal *et al.*, 1997b]. The same reason and parameterization has been found even in [Arampatzis *et al.*, 2000] for the task of document filtering in TREC-9.

In summary, literature shows that improvements can be derived by accurately setting the Rocchio parameters. However, this claim is neither proven with a systematic empirical study nor is a methodology to derive the good setting given. On the contrary, we have proposed a methodology for estimating parameters in a bound search space. Moreover, in the next section we will show that our approach and the underlying hypotheses are supported by the experimental data.

## 2.7 Performance Evaluations: PRC, Rocchio and SVM

The experiments are organized in three steps. First, in Section 2.7.1 the relationship between the $\rho$ setting and the performances of Rocchio classifier has been studied. Second, in Section 2.7.2 the statistical distribution of $\rho$ parameter has been extracted from samples in order to study its variability for each category. Third, *PRC* as well as the Rocchio performances have been evaluated over the *Reuters-21578* fixed *test-set* in Section 2.7.2. These results can be compared to other literature outcomes, e.g., [Joachims, 1998; Yang, 1999; Tzeras and Artman, 1993; Cohen and Singer, 1999]. Additionally, experiments of Section 2.7.3 over different splits as well as different corpora in two languages definitely assess the viability of the $PRC$ and the related estimation proposed in this paper. Finally, an evaluation of $SVM$ on Ohsumed and Reuters corpora is given. This enables a direct comparison between $PRC$ and one *state-of-the-art* TC model.

Three different collections have been considered: the *Reuters-21578*, the Ohsumed collection and the ANSA collection. Performance scores are expressed by means of interpolated BEP breakeven point and $f_1$ (see Section 2.4.2). The global performance of systems is always obtained by *microaveraging* the above measure over all categories of the target corpus, i.e., $\mu BEP$ and $\mu f_1$ of equations 2.14 and 2.15. The sets of features used in these experiments are all $Tokens$ that do not appear in the $SMART$ [Salton and Buckley, 1988] stop list[15]. They are 33,791 for Reuters, 42,234 for Ohsumed and 55,123 for ANSA. No feature selection has been applied. The feature weight in a document (for all TC models) is evaluated with Eq. 2.3 (i.e. the SMART *ltc* weighting scheme [Salton and Buckley, 1988]).

### 2.7.1 Relationship between accuracy and $\rho$ values

In these experiments we adopted the fixed split of the Reuters corpus as our *test-set* ($RTS$). The aim here is simply to study as $\rho$ influences the Rocchio accuracy. This latter has been measured by systematically setting different values of $\rho \in \{0, 1, 2, ..., 15\}$ in Eq. 2.20 and evaluating the BEP for each value.

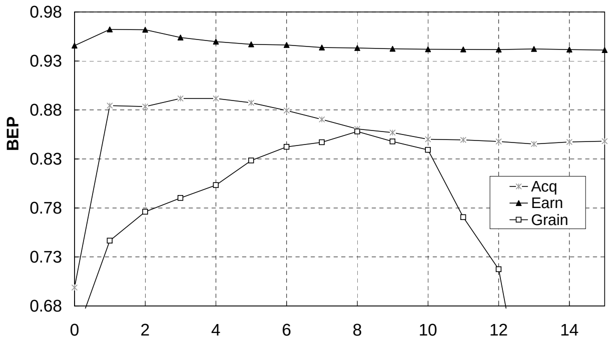


Figure 2.1: BEP of the Rocchio classifier according to different $\rho$ values for *Acq*, *Earn* and *Grain* classes of the Reuters Corpus.

Figures 2.1, 2.2 and 2.3 show the BEP curve on some classes of the Reuters Corpus with respect to $\rho$ value. For *Earn*, *Acq* and *Grain* there is available a large number of training documents (i.e. from 2,200 to 500). For them, the BEP increases according to $\rho$ until a max point is reached, then it begins to decrease for higher values of the parameter. Our hypothesis is that after BEP reaches the

[15] No stop list was applied for Italian corpus.

max point, further increase of $\rho$ produces relevant or weakly relevant features to be removed. In this perspective, the optimal $\rho$ setting would correspond to a quasi-optimal feature selection.

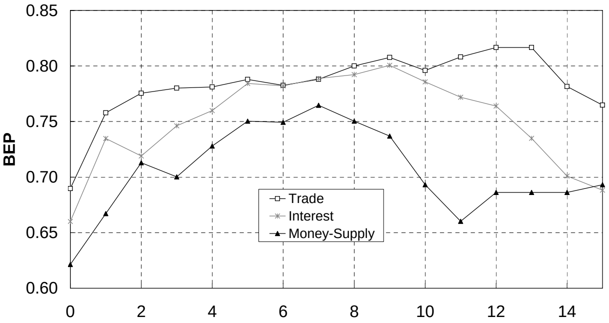


Figure 2.2: BEP of the Rocchio classifier according to different $\rho$ values for *Trade*, *Interest*, and *Money Supply* classes of the Reuters Corpus.

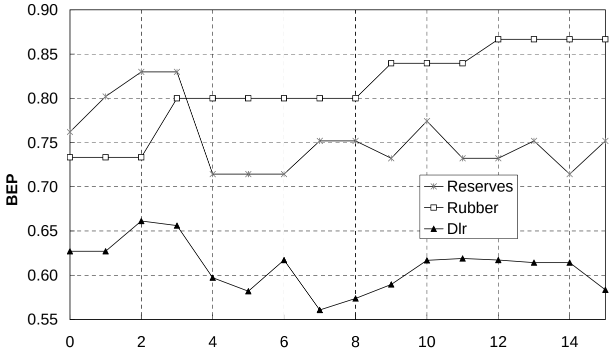


Figure 2.3: BEP of the Rocchio classifier according to different $\rho$ values for *Reserves*, *Rubber* and *Dlr* classes of the Reuters Corpus.

The *Trade*, *Interest* and *Money Supply* categories have a smaller number of documents available for training and testing (i.e. from 500 to 100). This reflects less regularity in $\rho$/BEP relationship. Nevertheless, it is still possible to identify convex curves in their plots. This is important as it allows us to infer that the absolute max is into the interval $\left[0, 15\right]$. The very small categories (i.e. less than 50 training documents) *Reserves*, *Rubber* and *Dlr* show a more chaotic relationship, and it is difficult to establish if the absolute maximum is in the target interval.

It is worth noting that the optimal accuracy is reached for $\rho > 1$. In contrast, it is a common belief that for a classifier the positive information should be more relevant than negative information. This suggests that (a) in Rocchio classifier, the contribute of the feature weights in negative examples has to be emphasized and (b) the $\gamma$ of Eq. 2.6 should not be interpreted as negative information control but as a simple parameter.

### 2.7.2 Performance Evaluation on the Reuters fixed *test-set*

In this experiment the performance of $PRC$ model over the fixed Reuters *test-set* ($RTS$) has been measured. The aim is to provide direct comparison with other literature results (e.g., [Yang, 1999; Joachims, 1998; Cohen and Singer, 1999; Lam and Ho, 1998]).

Twenty estimation sets $ES_1, ..., ES_{20}$ have been used to estimate the optimal ratio as described in Section 2.6.2. Once $\bar{\rho}$ is available for the target category, its profile can be built and the performance can be measured. The $PRC$ accuracy on $RTS$ is a $\mu f_1$ of 82.83%. This score outperforms all literature evaluations of the original Rocchio classifier: 78% obtained in [Cohen and Singer, 1999; Lam and Ho, 1998], 75% in [Yang, 1999] and 79.9% in [Joachims, 1998]. It is worth noting that this latter result has been obtained optimizing the parameters on $RTS$ as the aim was to prove the $SVM$ superiority independently on the parameters chosen (e.g., $\gamma$, $\beta$ and thresholds) for Rocchio.

To investigate the previous aspect we have measured directly the original Rocchio parameterized as in literature: $\gamma = 4$ and $\beta = 16$ ($\rho = .25$) and with $\gamma = \beta$ ($\rho = 1$). The results are shown in columns 2 and 3 of Table 2.5. When $\rho = 1$ is used, the global performance (78.79%) replicates the results in [Cohen and Singer, 1999; Lam and Ho, 1998] while for $\rho = .25$, it is substantially lower (72.61%). The explanation is the high number of features used in our experiments without applying any feature selection algorithm. A low ratio $\rho$ cannot filter an adequate number of irrelevant features and, consequently, the performances are low. As $\rho$ increases, a high number of noised features is removed and the performances improve. $PRC$, by determining the best parameter $\rho$ for each category, improves the Rocchio performance at least by 5 percent points.

To confirm the generality of the above results, cross validation experiments on Reuters and other corpora are presented in next section.

#### Variability of $\rho$ values across samples.

In this section we study the variability of $\rho$ which supports the explanation for the improved $PRC$ performances. The analysis of the distribution of the $\rho$ values requires an $ES$, i.e. the *estimation-set.*

$\rho$ values have been estimated over 20 samples $ES_1, ..., ES_{20}$. For each category $i$ and for each sample $k$ the best $\rho_i(ES_k)$ values has been estimated. The results are shown in Table 2.4. The values are reported for 14 categories of the Reuters Corpus, that includes more than 100 example documents. The name of categories is shown in column 1, while their sizes (expressed in number of documents) appear in column 2. The median, the means and standard deviation of $\gamma_i(ES_k)$ over the 20 samples are reported in columns 3,4 and 5.

When larger classes are available, the pointwise estimators (median and mean) seem represent the optimal $\rho$ values well. They are *near* the last column that represents the optimal $\rho$ evaluated on $RTS$. In other words whatever is the source information (i.e. the sample used for evaluating $\rho$) the resulting vector ranges in very small intervals. It approximates a *general* setting that, from one side, seems to reflect universal properties of the categories of a given collection, and, from the training point of view, can be derived via estimation (e.g., the median) over suitably large and numerous samples.

Table 2.4: Mean, Standard Deviation and Median of $\rho$ values estimated from samples.

| Categories | Size | Me | $\mu$ | Std.Dev. | Test-Set |
|---|---|---|---|---|---|
| earn | 2544 | 1 | 0.8 | 0.8 | 1 |
| acq | 1520 | 3 | 3.8 | 2.4 | 3 |
| money-fx | 456 | 10 | 6.0 | 5.1 | 10 |
| grain | 374 | 7 | 6.9 | 2.0 | 8 |
| crude | 366 | 10 | 7.3 | 4.7 | 12 |
| interest | 312 | 9 | 8.0 | 2.6 | 9 |
| trade | 312 | 9 | 6.0 | 4.8 | 12 |
| ship | 181 | 1 | 3.0 | 4.5 | 7 |
| wheat | 181 | 10 | 7.8 | 5.1 | 15 |
| corn | 151 | 10 | 10.0 | 1.7 | 15 |
| dlr | 111 | 0 | 0.0 | 0.0 | 0 |
| Money-supply | 110 | 4 | 4.3 | 4.0 | 7 |
| oilseed | 110 | 10 | 7.9 | 4.1 | 11 |
| sugar | 108 | 10 | 6.7 | 4.9 | 11 |

### 2.7.3 Cross evaluation

In order to assess the general performances of the $PRC$ and of the original Rocchio classifier, wider empirical evidences are needed on different collections and languages. Moreover, to estimate the best TC accuracies achievable on the target corpora, we have also evaluated the Support Vector Machine ($SVM$) classifier [Joachims, 1998].

Performance figures are derived for each category via a cross validation technique applied as follows:

1. Generate $n = 20$ random splits of the corpus: 70% for training ($LS^\sigma$) and 30% for testing ($TS^\sigma$).

2. For each split $\sigma$

   (a) Extract 20 sample[16] ${ES^\sigma}_1 ... {ES^\sigma}_{20}$ from $LS^\sigma$.

   (b) Learn the classifiers on $LS^\sigma - {ES^\sigma}_k$ and for each ${ES^\sigma}_k$ evaluate: (i) the thresholds associated to the BEP and (ii) the optimal parameters $\rho$.

   (c) Learn the classifiers Rocchio, $SVM$ and $PRC$ on $LS^\sigma$: in case of $PRC$ use the estimated $\bar{\rho}$.

   (d) Use $TS_\sigma$ and the estimated thresholds to evaluate $f_1$ for the category and to account data for the final processing of the global $\mu f_1$.

3. For each classifier evaluate the mean and the Standard Deviation for $f_1$ and $\mu f_1$ over the $TS_\sigma$ sets.

It is worth noting that the fixed *test-set* ($RTS$) and the *learning-set* of the Reuters Corpus have been merged in these experiments to build the new random splits.

Again, original Rocchio classifier has been evaluated on two different parameter settings selected from the literature (i.e. $\gamma = \beta$ and $\gamma = 4$ and $\beta = 16$). Tables 2.5 and 2.6 reports the $\mu f_1$ over 90 categories and the $f_1$ (see Section 2.4.2) for the top 10 most populated categories. Original Rocchio accuracy is shown in columns 2, 3, 4 and 5 of the first table. In the second table, columns 2 and 3 refer to $PRC$ while columns 4 and 5 report $SVM$ accuracy. The $RTS$ label indicates that only the Reuters fixed *test-set* has been used to evaluate the results. In contrast, the $TS^\sigma$ label means that the measurements have been derived averaging the results on 20 splits.

The symbol ± precedes the Std. Dev. associated to the mean. It indicates the variability of data and it can be used to build the confidence limits. We observe that our $SVM$ evaluation on Reuters $RTS$ (85.42%) is in line with the literature (84.2 %) [Joachims, 1998]. The slight difference in [Joachims, 1998] is due to the application of a stemming algorithm, a different weighting scheme, and a feature selection (only 10,000 features were used there). It is worth noting that the global $PRC$ and $SVM$ outcomes obtained via cross validation are higher than those evaluated on the $RTS$ (83.51% vs. 82.83% for $PRC$ and 87.64% vs. 85.42% for $SVM$). This is due to the non-perfectly random nature of the fixed split that prevents a good generalization for both learning algorithms.

The cross validation experiments confirm the results obtained for the fixed Reuters split. $PRC$ improves about 5 point (i.e. 83.51% vs. 78.92%) over Rocchio parameterized with $\rho = 1$ with respect to all the 90 categories ($\mu f_1$). Note that $\rho = 1$ (i.e. $\gamma = \beta$) is the best literature parameterization. When a more general parameter setting [Cohen and Singer, 1999] is used, i.e. $\rho = .25$, $PRC$ outperforms Rocchio by $\sim$ 10 percent points. Tables 2.5 and 2.6 shows a high improvement even for the single categories, e.g., 91.46% vs. 77.54% for

[16] Each $ES_k$ includes about 30-40% of training documents, depending on the corpus.

Table 2.5: Rocchio $f_1$ and the $\mu f_1$ performances on the Reuters corpus. $RTS$ is the Reuters fixed *test-set* while $TS^\sigma$ indicates the evaluation over 20 random samples.

| | Rocchio | | | |
|---|---|---|---|---|
| Category | RTS | | $TS^\sigma$ | |
| | $\rho = .25$ | $\rho = 1$ | $\rho = .25$ | $\rho = 1$ |
| earn | 95.69 | 95.61 | 92.57±0.51 | 93.71±0.42 |
| acq | 59.85 | 82.71 | 60.02±1.22 | 77.69±1.15 |
| money-fx | 53.74 | 57.76 | 67.38±2.84 | 71.60±2.78 |
| grain | 73.64 | 80.69 | 70.76±2.05 | 77.54±1.61 |
| crude | 73.58 | 80.45 | 75.91±2.54 | 81.56±1.97 |
| trade | 53.00 | 69.26 | 61.41±3.21 | 71.76±2.73 |
| interest | 51.02 | 58.25 | 59.12±3.44 | 64.05±3.81 |
| ship | 69.86 | 84.04 | 65.93±4.69 | 75.33±4.41 |
| wheat | 70.23 | 74.48 | 76.13±3.53 | 78.93±3.00 |
| corn | 64.81 | 66.12 | 66.04±4.80 | 68.21±4.82 |
| $\mu f_1$ (90 cat.) | 72.61 | 78.79 | 73.87±0.51 | 78.92±0.47 |

Table 2.6: $PRC$ and $SVM$ $f_1$ and the $\mu f_1$ performances on the Reuters corpus. $RTS$ is the Reuters fixed *test-set* while $TS^\sigma$ indicates the evaluation over 20 random samples.

| | $PRC$ | | $SVM$ | |
|---|---|---|---|---|
| Category | RTS | $TS^\sigma$ | RTS | $TS^\sigma$ |
| earn | 95.31 | 94.01±0.33 | 98.29 | 97.70±0.31 |
| acq | 85.95 | 83.92±1.01 | 95.10 | 94.14±0.57 |
| money-fx | 62.31 | 77.65±2.72 | 75.96 | 84.68±2.42 |
| grain | 89.12 | 91.46±1.26 | 92.47 | 93.43±1.38 |
| crude | 81.54 | 81.18±2.20 | 87.09 | 86.77±1.65 |
| trade | 80.33 | 79.61±2.28 | 80.18 | 80.57±1.90 |
| interest | 70.22 | 69.02±3.40 | 71.82 | 75.74±2.27 |
| ship | 86.77 | 81.86±2.95 | 84.15 | 85.97±2.83 |
| wheat | 84.29 | 89.19±1.98 | 84.44 | 87.61±2.39 |
| corn | 89.91 | 88.32±2.39 | 89.53 | 85.73±3.79 |
| $\mu f_1$ (90 cat.) | 82.83 | 83.51±0.44 | 85.42 | 87.64±0.55 |

the *grain* category. The last two columns in Table 2.6 reports the results for the linear version of $SVM$[17].

[17] We have tried to set different polynomial degrees (1,2,3,4 and 5). As the linear version has shown the best performance we have adopted it for the cross validation experiments.

Table 2.7: Performance Comparisons among Rocchio, $SVM$ and $PRC$ on Ohsumed corpus.

| | Rocchio (BEP) | | $PRC$ | | $SVM$ |
|---|---|---|---|---|---|
| Category | $\rho = .25$ | $\rho = 1$ | BEP | $f_1$ | $f_1$ |
| Pathology | 37.57 | 47.06 | 48.78 | 50.58 | 48.5 |
| Cardiovascular | 71.71 | 75.92 | 77.61 | 77.82 | 80.7 |
| Immunologic | 60.38 | 63.10 | 73.57 | 73.92 | 72.8 |
| Neoplasms | 71.34 | 76.85 | 79.48 | 79.71 | 80.1 |
| Digestive Syst. | 59.24 | 70.23 | 71.50 | 71.49 | 71.1 |
| MicroAv. (23 cat.) | 54.4±.5 | 61.8±.5 | 66.1±.4 | 65.8±.4 | 68.37±.5 |

Table 2.8: Performance comparisons between Rocchio and $PRC$ on ANSA corpus

| | Rocchio (BEP) | | $PRC$ | |
|---|---|---|---|---|
| Category | $\rho = 0.25$ | $\rho = 1$ | BEP | $f_1$ |
| News | 50.35 | 61.06 | 69.80 | 68.99 |
| Economics | 53.22 | 61.33 | 75.95 | 76.03 |
| Foreign Economics | 67.01 | 65.09 | 67.08 | 66.72 |
| Foreign Politics | 61.00 | 67.23 | 75.80 | 75.59 |
| Economic Politics | 72.54 | 78.66 | 80.52 | 78.95 |
| Politics | 60.19 | 60.07 | 67.49 | 66.58 |
| Entertainment | 75.91 | 77.64 | 78.14 | 77.63 |
| Sport | 67.80 | 78.98 | 80.00 | 80.14 |
| MicroAverage | 61.76±.5 | 67.23±.5 | 72.36±.4 | 71.00±.4 |

Tables 2.7 and 2.8 report the results on the other two corpora, respectively Ohsumed and ANSA. The new data on these tables is the BEP evaluated directly on the $TS^{\sigma}$. This means that the estimation of thresholds is not carried out and the resulting outcomes are upperbounds of the real accuracies. We have used these measurements to compare the $f_1$ values scored by $PRC$ against the Rocchio upperbounds. This provides a strong indication of the superiority of $PRC$ as both tables show that Rocchio BEP is always 4 to 5 percent points under $f_1$ of $PRC$. Finally, we observe that $PRC$ outcome is close to $SVM$ especially for the Ohsumed corpus (65.8% vs. 68.37%).

### $PRC$ complexity

The evaluation of Rocchio classifier time complexity can be divided into three steps: *pre-processing*, *learning* and *classification*. The *pre-processing* includes the document formatting and the extraction of features. We will neglect this extra time as it is common in almost all text classifiers.

The learning complexity for original Rocchio relates to the evaluation of weights in all documents and profiles. Their evaluation is carried out in three important steps:

1. The $IDF$ is evaluated by counting for each feature the number of documents in which it appears. This requires the ordering of the pair set <*document, feature*> by feature. The number of pairs is bounded by $m \times M$, where $m$ is the maximum number of features in a documents and $M$ is the number of training documents. Thus, the processing time is $O(m \times M \times log(m \times M))$.

2. The weight for each feature in each document is evaluated in $O(m \times M)$ time.

3. The profile building technique, i.e. Rocchio formula, is applied. Again, the tuple set <*document, feature, weight*> is ordered by feature in $O(m \times M \times log(m \times M))$ time.

4. All weights that a feature $f$ assumes in positive (negative) examples are summed. This is done by scanning sequentially the <*document, feature, weight*> tuples in $O(M \times m)$ time. As result, the overall learning complexity is $O(m \times M \times log(m \times M))$.

The classification complexity of a document $d$ depends on the retrieval of weights for each feature in $d$. Let $N$ be the total number of unique features; it is an upperbound of the number of features in a profile. Consequently, the classification step takes $O(m \times log(N))$.

In the $PRC$ algorithm, an additional phase is carried out. The accuracy produced by $\rho$ setting has to be evaluated on a *validation-set* $V$. This requires the re-evaluation of profile weights and the classification of $V$ for each chosen $\rho$. The re-evaluation of profile weights is carried out by scanning all <*document, feature, weight*> tuples. Note that the tuples need to be ordered only one time. Consequently, the evaluation of one value for $\rho$ takes $O(m \times M) + O(|V|m \times log(N))$. The number of values for $\rho$, as described in the previous section, is $k = Max\_limit/\Delta\rho$. The complexity to measure $k$ values is $O(mM \times log(mM)) + k(O(m \times M) + |V| \times O(m \times log(N)))$. The cardinality of the *validation-set* $|V|$ as well as $k$ can be considered constants. In our interpretation, $k$ is an intrinsic property of the target categories. It depends on feature distribution and not on the number of documents or features. Moreover, $N$ is never greater than the product $M \times m$. Therefore, the final $PRC$ learning complexity is $O(mM \times log(mM)) + k \times O(mM) + k|V| \times O(m \times log(mM)) = O(mM \times log(mM))$, i.e. the complexity of the original Rocchio learning.

The document classification phase of PRC does not introduce additional steps with respect to the original Rocchio algorithm, so it is characterized by a very efficient time complexity, i.e. $O(m \times log(N))$.

## 2.8 Conclusions

In this Chapter the basic steps for the designing of a general classifier have been described. In particular new weighting schemes and a novel score adjustment techniques have been presented. Two representative model, Rocchio and SVM, have been introduced: the former is one of the most efficient classifier whereas the latter has the highest accuracy.

The high efficiency of Rocchio classifier has produced a renewed interest in its application to operational scenarios. Thus, we have study a methodology for setting the Rocchio parameters that improves accuracy and keeps the same efficiency of the original version. This methodology reduces the search space of parameters by considering that: (a) in TC only one parameter is needed, i.e., the ratio $\rho$ between $\gamma$ and $\beta$, and (b) $\rho$ can be interpreted as a feature selector. This has allowed us to bind the search space for the ratio values since the $\rho$ maximal value corresponds to the selection of 0 features. Moreover, empirical studies have shown that the $\rho$/BEP relationship can be described by a convex curve. This suggests a simple and fast estimation procedure for deriving the optimal parameter (see Section 2.6.1).

The resulting model, the Parameterized Rocchio Classifier ($PRC$) has been validated via cross validation, using three collections in two languages (Italian and English). In particular, a comparison with the original Rocchio model and the $SVM$ text classifiers has been carried out. This has been done in two ways: (a) on the Reuters fixed split that allows $PRC$ to be compared with literature results on TC and (b) by directly deriving the performance of Rocchio and $SVM$ on the same data used for $PRC$.

Results allow us to draw the following conclusions:

- First, $PRC$ systematically improves original Rocchio parameterized with the best literature setting by at least 5 percent points, and it improves the general setting by 10 percent points. Comparisons with $SVM$ show the performances to be relatively close (-4% on Reuters and -2.5% on Ohsumed).
- Second, the high performance, (i.e., 82.83%) on the Reuters fixed *test-set* collocates $PRC$ as one of the most accurate classifiers on the Reuters corpus (see [Sebastiani, 2002]).
- Third, the low time complexity for both training and classification phase makes the $PRC$ model very appealing for real (i.e. operational) applications in Information Filtering and Knowledge Management.

Finally, the feature selection interpretation of parameters suggests a methodology to discover the *specific-term* of a category with respect to the other ones.

# Chapter 3

# NLP for Text Categorization

Chapter 1 has summarized some of the attempts to use advanced document representation to improve document retrieval. The conclusive results were that current NLP slightly improves the basic retrieval systems. When pure statistical *state-of-the-art* models are adopted either NLP is not useful or a comparison cannot be carried out as the less efficiency of NLP. For TC are available fewer studies as it is a relatively new research area (compared to IR) and some of these, e.g., [Raskutti *et al.*, 2001; Tan *et al.*, 2002] have shown improvements by using very basic language processing techniques. Thus, deriving a final conclusion on the role of NLP is more difficult than for document retrieval.

In this chapter advanced document representations, introduced in Section 1.2, have been investigated. Several experiments have been carried out:
First, efficient NLP techniques are used in fast TC models, the profile-based, to derive efficient categorization systems. Several weighting schemes, inference methods and adjustment score techniques have been considered. The aims were (a) to study how NLP impacts some different versions of profile-based classifiers, and (b) to design efficient and accurate NLP-driven TC tmodels. Second, more complex and less efficient NLP algorithms have been studied. They include the extraction of *terminological expressions*, i.e., important domain complex nominals and the selection of the correct *word senses* by using three WSD algorithms. These last tests allow us to verify the hypothesis claimed in [Voorhees, 1998; Smeaton, 1999], i.e., when the correct senses are used in IR the resulting system highly improves. The above study complete the NLP for TC survey. In fact, almost all *trendy* NLP techniques for IR have been studied and experimented.

Section 3.1 describes the NLP techniques applied to extract feature for indexing. Section 3.2 shows the impact of efficient derived features such as lemma, Proper Nouns and POS-tag in efficient statistical profile-based models. Section 3.3 reports the experiments for Rocchio, $PRC$ and $SVM$ on the advanced NLP document representations. In particular sections 3.3.2 and 3.3.1 report experiments on syntactic information, i.e., lemmatization, POS-tagging, proper nouns and terminological expressions whereas Section 3.3.4 shows the impact of semantic representation using word senses. Related work has been examined in Section 3.3.6. Finally, Section 3.4 derive the conclusions on using NLP for TC.

## 3.1 Natural Language Feature Engineering

The role of linguistic content in TC is twofold: from one side it is embodied by specific information with respect to entities and facts cited in documents. Proper Nouns for Companies, Location and Persons, or the events involving those entities (e.g., managing succession events as indicators of topics like *Industry News*) are example of such type of linguistic content. This information is widely used within the IE area, e.g. MUC-6, MUC-7 [1] and [Pazienza, 1997] , involved in very granular and specific recognition. On the other hand, content refers also to the set of *typical* words, i.e. expressions and terminological units that co-occur in a document or in documents of the same class. This provides an overall picture of what a topic is, and what it deals with. This second form of linguistic content is based on:

- A tight separation between content words (i.e. open syntactic classes such as nouns, verbs and adjectives) and other less relevant information (e.g., functional classes like prepositions or complex functional expressions *as far as* or *in order to*). The need of this separation is known since the early research in IR [Salton, 1989] that motivated the use of *stoplists*.

- The identification of the syntactic role of each word in its corresponding context: for example verbal from nominal uses of a lemma can be distinguished (*ready to land* vs. *suitable public lands*). The syntactic role allows to select the more informative class of words, i.e. Nouns, and to perform a first level of word disambiguation, e.g., *book* and *to book*. The syntactic category of the word *book*, clearly, decides which is the most suitable choice between categories like *Book Sales* and *Travel Agency*.

- The identification of linguistically motivated structures that behave non-compositionally, and thus require a completely different process with respect to other phenomena. Possibly complex Proper Nouns (e.g., *Shell Transport & Trading Co. PLC*) are an example, as they should not be modeled similarly to common nouns in TC. This less granular form of linguistic content could be very useful to enhance the document representation. As it provides core information that the single words may not capture. When used for TC, the accuracy in the recognition of the different components reflects in the accuracy of the classification processes. Empirical evidences on this relationship are still necessary, and our study aims to add further information to this issue.

[1] The Message Understanding Conference focused on the task of Information Extraction `http://www.cs.nyu.edu/cs/faculty/grishman/muc6.html`.

The above structures add to the simple words syntactic information, which could be useful to improve the accuracy in TC. A different source of linguistic information are the senses of words. These give a more precise sketch of what the category is concerning. For example, a document that contains the nouns *share*, *field* and the verb *to raise* could refer to agricultural activities, when the senses are respectively: *plowshare*, *agricultural field* and *to cultivate by growing*. At the same time, the document could concern economic activities when the senses of the words are: *company share*, *line of business* and *to raise costs*. This shows that the availability of word senses in document representation could improve the TC accuracy.

The next section will describe our system to extract the above sets of syntactic and semantic information.

### 3.1.1 Language processing for Text Categorization in TREVI

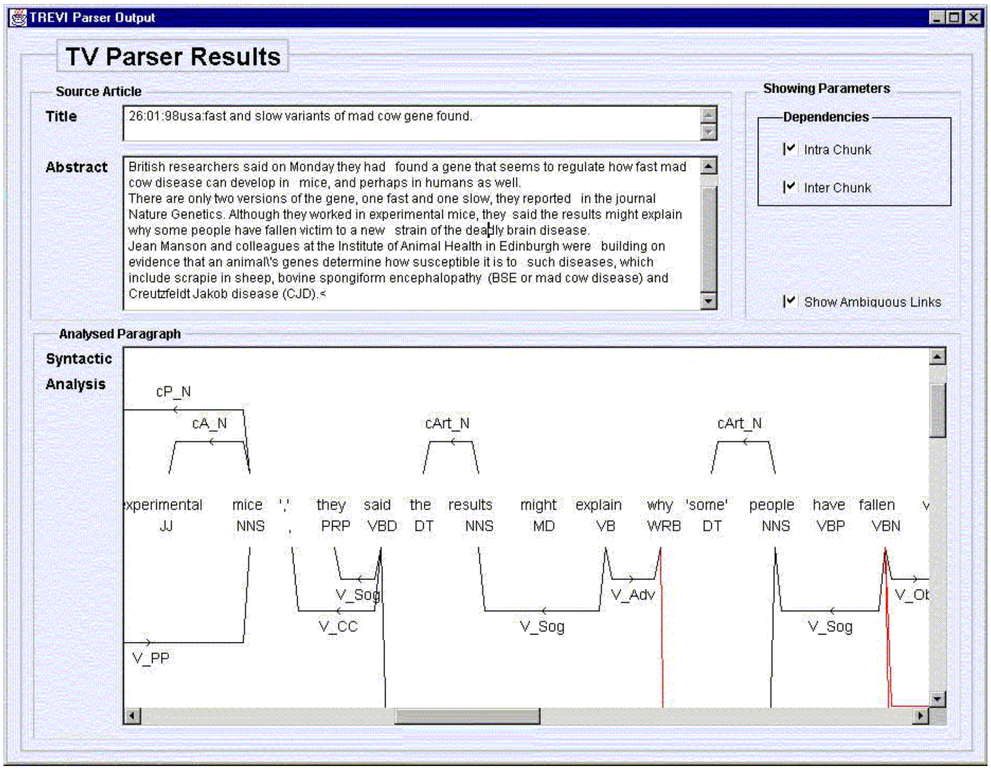
The linguistic information described in the previous section requires accurate recognition/extraction capabilities during the corpus-preprocessing phase. The linguistic processor adopted in our studies is TREVI. TREVI is a system for Intelligent Text Retrieval and Enrichment of news agency texts. In TREVI specific NLP technologies deal with the required linguistic content. All the experiments analyzed in this paper are based on the TREVI NLP components, described in the rest of this section.

The TREVI target application is to provide support to agencies in the management of different, multilingual and geographically distributed streams of news. Reuters, as a member of the Consortium, has been used as a main *User Case* for the released prototype. An editorial board is usually in charge of managing news, i.e. classifying and enriching them in order to facilitate their management, future retrieval and delivery. The TREVI components are servers cooperating to the *processing*, *extraction*, *classification*, *enrichment* and *delivery* of news. Mainly, two TREVI components contribute to the TC (sub-)task:

- the *Parser*, i.e. a full linguistic preprocessor that takes a normalized versions of a news item and produces a set of grammatical (e.g., subj/obj relations) and semantic (e.g., word senses in an ontology) information related to that 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 [Basili *et al.*, 1998b] is a complex (sub)system combining tokenization, lemmatization (via an independent lexical server), Part-of-Speech tagging [Brill, 1992; Church, 1988] and robust parsing [Basili *et al.*, 1998c]. Details on the linguistic methods and algorithms for each phase can be found in the related publications [Basili *et al.*, 1998b; 1998c; Basili and Zanzotto, 2002] and will not be here described as they go beyond the purposes of this thesis.



Title: 26:01:98usa:fast and slow variants of mad cow gene found.

Abstract: British researchers said on Monday they had found a gene that seems to regulate how fast mad cow disease can develop in mice, and perhaps in humans as well.
There are only two versions of the gene, one fast and one slow, they reported in the journal Nature Genetics. Although they worked in experimental mice, they said the results might explain why some people have fallen victim to a new strain of the deadly brain disease.
Jean Manson and colleagues at the Institute of Animal Health in Edinburgh were building on evidence that an animal\'s genes determine how susceptible it is to such diseases, which include scrapie in sheep, bovine spongiform encephalopathy (BSE or mad cow disease) and Creutzfeldt Jakob disease (CJD).<

☑ Intra Chunk
☑ Inter Chunk
☑ Show Ambiguous Links

Figure 3.1: Screendump: TREVI Parser output

Figure 3.1 shows the GUI of the TREVI parser on a Reuters news. The GUI shows the title, the full text, some complementary information and in the large panel the output produced by the parser in form of an annotated syntactic graph. The shown sentence in the panel is:

*Although they worked in experimental mice, they said the results might explain why some people have fallen victim to a new strain of the deadly brain disease.*

The graph is based on the word sequence (from *experimental* to *fallen* in the visible segment). Each word is tagged by its own Part-Of-Speech: for example *experimental*, *mice*, *explain*, *why* and *fallen* are tagged respectively as adjective ($JJ$), plural noun ($NNS$), base verb ($VB$), $Wh$-adverb ($WRB$) and past participle ($VBN$). The grammatical links within chunks (i.e. kernels of nouns or verb phrases following [Abney, 1996]) are shown above the sentence: complete noun phrases like *the results* or *some people* are recognized as valid chunks and grammatical relations between their participants (e.g., determiner-noun relations) are annotated via syntactic types (*Art_N*). Under the sentence other relations are shown. Subjects and objects of verbs are described as grammatical relations among the head words of chunks[2]. The material shown in Fig. 3.1 includes the subject of the verb *to fall* in the fragment *... some people have fallen victim ...* (see relation typed `V_Sog`). Although no Proper Noun is shown in the example of Fig. 3.1, the employed specific Named-Entity grammars in TREVI provide the detection and tagging ($NNP$ is the specific NE tag) of units like *New York*, *Jean Mason* or *Institute of Animal Health*.

The $Parser$ thus detects in every documents the following set of information:

1. Possibly complex **Tokens** and **lemmas**. Simple words (e.g., *bank*, *match*) as well as complex terminological expressions (e.g., entire noun phrases as *bond issue* or functional expressions as *in order to*) are detected and treated as atomic units during the later phases;

2. **Proper Nouns** (PNs). Set of domain (i.e. User) specific Named-Entities are recognized by accessing extensive catalogs as well as by special-purpose grammars. Typed proper nouns are derived from news, e.g., `company` and `person` are valid types for *Oracle* and *Woody Allen* respectively.

3. **Syntactic Categories** of lemmas in text. Each unit of text (i.e. simple or complex) is assigned with a single Part-of-Speech (POS). Indexes can be thus built over POS, so that verbal and nominal occurrences of a given lemma are independent (e.g., *results*/`VB` is different from *results*/`NNS`)

4. **Major grammatical relations** (i.e. `Subj/Obj` relations among words) are detected with a significant accuracy (about 80%, see [Basili *et al.*, 1998c] for an evaluation of the robust parser). Each news is thus annotated also with basic structures made of significant constituents (verbs and their modifiers).

The example in Fig. 3.1 is a simple case aiming to suggest the basic information extracted by the employed linguistic process. It is to be noticed that the TREVI parser is based on a modular architecture. Its average processing time is more than 80 words per second (see [Basili *et al.*, 1998c] for extensive evaluation in English and Italian). This speed, although quite reasonable for a variety of NLP tasks, could not be compatible with time constraints in some operational scenarios for TC. However, when a higher speed is required the parser can be scaled down to increase the computational speed. For example some linguistic processors such as the chunk-based parsing component can be deactivated.

### 3.1.2 Basic NLP-derived feature set

The problem of using NLP techniques to extract relevant indexes relates not only to the designing of effective models but mainly in keeping efficiency as lower as possible. The models proposed in this section focus on some linguistic levels

[2] The *head* of a *chunk* is the main meaning carrier of the entire structure, as for *results* in the chunk *the results*). Only the head enters in grammatical relations between two chunks.

that are currently supported by an efficient technology allowing fast processing of huge amount of data. Part of speech tagging, lemmatization and proper nouns recognition are simple linguistically motivated techniques that can be applied efficiently with a very high accuracy [Brill, 1992; Church, 1988]. This linguistic information provides a richer knowledge about texts and is expected to improve the selectivity exhibited by text categorizers.

The relevant information derived during parsing is used in TREVI for TC by the $Subject\ Identifier$ component. In TREVI, only nouns, verbs and adjectives are considered candidates features. We define the basic NLP-features the pairs:

$$\texttt{<lemma, POStags>} \tag{3.1}$$

where the valid `POStag` labels express noun, verb or adjective tags (e.g., `NN`, `NNS` `VBN`, `VB` or `JJ`). Proper Nouns (PNs) are also included in the feature set like the other lemmas so that they do not have a different treatment. Notice that stop lists are not required as POS tagging supplies the corresponding, and linguistically principled, filtering ability. It is expected that the overall process (i.e. recognition of functional units, proper nouns and POS tag assignment) supports the selection of a better set of candidate features.

The representation defined in (3.1) is able to express linguistic as well as non linguistic feature sets. Document as well as profile vectors are obtained by weighting the above pairs. Non linguistic feature sets (as discussed and tested in Section 3.2.1) are obtained by simply ignoring the second component in (3.1) (i.e. the POS label) and merging the pairs with identical lemmas. However, notice that the POS labels are always used for feature selection: non significant word classes (e.g., $WH$-adverbs with tag $WRB$) are preliminarily eliminated in any experiment. We refer to the non linguistic feature set as the *TREVI-tokens.*

In synthesis, we can say that the basic NLP-derived features are characterized by three important properties:

- First, information significant for TC is extracted via a modular NLP approach. This is able to isolate a variety of linguistic levels (ranging from simple lemmas to complex proper nouns or grammatical units, e.g., irrelevant functional expressions like *in order to* correctly POS-tagged).
- Second, the adopted technology reflects current *state-of-the-art* in NLP (e.g., modular design and engineering of a large scale system (TREVI) and chunk-based parsing) thus providing efficient and suitable (and configurable) processing for each selected level.
- Finally, a combination of language processing within IR is defined via an enriched feature representation (definition in (3.1)). It is designed to naturally support a quantitative model (i.e. metrics in feature vector spaces) and preserve the expressiveness of the extracted linguistic information.

### 3.1.3 Terminology-based Document Representation

One of the objectives of our research is to study the role of linguistic information in the description (i.e. feature extraction) of different classes in a TC task. By linguistically analyzing documents in the target categories, we noticed that these latter are often characterized by sets of *typical* concepts usually expressed by specific phrases, i.e. linguistic structures synthesizing widely accepted definitions (e.g., *bond issues* in topics like *Finance* or *Stock Exchange*). Such complex nominals express information useful to capture semantic aspects of a *topics*. Phrases that could be useful for TC belong to the following general classes:

- Proper Nouns (PN), which identify entities participating to events described by a text. Most named entities are locations, persons or artifacts and are tightly related to the topics. PN-based features can improve performances, as reported in [Basili *et al.*, 2001].
- Terminological expressions, i.e. complex nominal expressing domain concepts. Domain concepts are usually identified by multiwords (e.g., *bond issues*). Their detection results in a more precise set of features to be included in the target vector space.

The above phrases embody domain specific knowledge [Basili *et al.*, 1997] so that they can provide selective features in TC. In fact, phrases specific to a given topics $C_i$ can be learnt from the training material so that their matching in test documents $d$ is a trigger for classifying $d$ in $C_i$.

The availability of linguistically motivated terminological structures is usually ensured by external resources, i.e. thesauri or glossaries. However, extensive repositories are costly to be developed or simply missing in most domains. An enumerative approach cannot be fully applied. Automated methods for learning both Proper Nouns and terminological expressions from texts have been thus introduced and they can play a key role in content sensitive TC. While, the detection of Proper Nouns is easier achieved by applying a grammar that takes into a account capital letters of nouns, e.g., *George Bush*, terminology extraction requires a more complex process. Next section describes the adopted terminology acquisition method. The result is a self-adapting process that tunes its behavior to the target domain (i.e. the set of $C_i$ topics).

### Corpus-driven terminology extraction

The automated compilation of a domain specific terminological dictionary is a well know problem in NLP. Several methods for corpus-driven terminology extraction have been proposed (e.g., [Daille, 1994; Arppe, 1995; Basili *et al.*, 1997]). The terminology extraction algorithm that we used is an inductive (batch) method early introduced in [Basili *et al.*, 1997]. It is based on an integration of symbolic and statistical modeling along three major steps:

- First, a set of relevant atomic terms $ht$ (i.e. singleton words, e.g., *issue*) are identified by means of traditional techniques[3]. These terms are potential grammatical heads of complex terminological expressions (e.g., *bond issues*).

[3] The $TF \times IDF$ score early suggested in [Salton and Buckley, 1988] is here employed.

- Linguistically principled grammars are then applied to identify full linguistic structures (i.e. complex forms headed by *ht*) as admissible candidates. In this phase, simple noun phrase grammars (e.g., `NP <- [Det] [Adj*] N NP`) are applied to train texts previously preprocessed. Preprocessing here applies tokenization, Part-of-Speech tagging and lemmatization of incoming texts. The outcome of this phase includes candidate terms expressing genuine terminological entries, e.g., *bond issues*, *financial institution*, *chief executive*, *congenital heart defect* as well as generic, i.e. irrelevant, expressions, e.g., *last week*, *high rates*, *three time*, *early age.*
- 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.

The critical mechanism in the above process is the interaction between the NP grammars and the statistical filters. Term candidates extracted during the second step are couples $(x, \vec{y})$, where $\vec{y}$ represents the sequence of (left and/or right) modifiers, e.g., *(issue, (-1,bond))*, *(defect, ((-2,congenital),(-1,heart))* for *bond issue* and *congenital heart defect*, respectively. Mutual information (MI), [Fano, 1961], has been often used to capture linguistic relations between words (e.g., [Church and Hanks, 1990; Dagan *et al.*, 1994]):

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

The stronger is the relation between $x$ and $y$ the larger is the joint with respect to marginal probabilities[4]. The basic problem is that MI (and its estimation) is concerned with only two events, and is better suited with bigrams, e.g., *bond issue.* Longer expressions usually require an iterative estimation ([Basili *et al.*, 1997; Daille, 1994]), where first (sub)bigrams of longer structures are accepted, re-estimation of their occurrences in the corpus is run and then filtering of a new binary event is applied. 5-grams in this case would require 4 re-estimations.

In [Basili *et al.*, 1998a] 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 an entire (more than binary) relation is considered between word $x$ and the vector $\vec{y} = (y_1, y_2, ..., y_n)$ of its modifiers. The MI estimation $I(x, \vec{y})$ is carried out in two steps. First each $i$-th component, $\hat{I}(x, y_i)$, is estimated. Then, graphical comparison among the resulting $\hat{I}(x, y_i)$ is applied. The $\hat{I}(x, y_i)$ determine points in an histogram describing a full complex noun phrase. If a semantic relation holds between the modifiers $\vec{y}$ and the head $x$, then the obtained plot should be as flat as possible, i.e. no significant difference between the $\hat{I}(x, y_i)$

[4] A variety of estimations and extension of MI have been proposed, [Church and Hanks, 1990], like: $\hat{I}(x, y) = \log_2 N \frac{f_i(x,y)}{f(x)f(y)}$, where $f_i(x, y)$ is the frequency of co-occurrence of words $x$ and $y$ at distance $i$.

values should be observed. In this way each candidate term $(x, \vec{y})$ is analyzed looking "in parallel" to all its different MIs (i.e. $\hat{I}(x, y_i) \quad \forall i$). Thresholding on the differences provides a straightforward and efficient decision criteria applied without iteration.

The above methods has been largely applied to English texts in [Basili *et al.*, 1997]. Evaluation of the performances of the above acquisition method are very complex, as it is difficult to establish a clear separation among terms and non terms. However, the result is always a more or less precise set of complex nominals significant for the underlying domain, i.e. a terminological dictionary. The relevance of each term/feature for TC can be assessed by means of the feature selection and weighting method described in Chapter 2.

The terminology extraction for TC should include these additional steps:

1. Terminological dictionary, namely $T_i$, is obtained by applying the above method to training data for each single category $C_i$.

2. The global terminology set $T$ is obtained by merging the different $T_i$, i.e. $T = \cup_i T_i$. As test data are distributed in an unknown manner throughout different classes, a single terminological dictionary $T$ is needed during testing.

3. The TREVI processor can thus rely on $T$ during the matching of features within incoming test documents. Notice that when a given term $f$ is included in different category dictionaries $T_i$, it is likely to receive, from the learning model, a different weight $\vec{a}^i_f$ (i.e., $\vec{W}^i_f$) for each class $C_i$.

### 3.1.4 Semantic representation

Text Categorization and Word Sense Disambiguation are areas of language processing that have recently received a great deal of attention. This is because of the impact they have on harnessing the ever-growing textual information posted on the Internet or other on-line document collections. In the study we report in this thesis, we tried to see if the accuracy of TC could be improved when more sophisticated linguistic representations based on word meanings would also be available.

Word Sense Disambiguation is a Natural Language Processing (NLP) technique that assigns meanings to content words (e.g., nouns, verbs, adjectives or adverbs) based on dictionary definitions. In general, words may be ambiguous both syntactically and semantically. For instance, the word *hit* may be either a noun, or a verb. When a noun, *hit* may have 6 senses, as defined in WordNet (*http://www.cogsci.princeton.edu/∼wn/*), whereas when it is a verb it has 15 senses. Part-of-speech (POS) taggers like Brill's POS-tagger [Brill, 1992] assign POS-tags to words with fairly high precision (95 %). Recent WSD evaluations performed in SENSEVAL [Kilgarriff and Rosenzweig, 2000] show that current unsupervised learning methods for WSD achieve a precision of 80% for nouns, 70% for verbs and 75% for adjectives.

By adding features representing POS and senses of words, the document representation of text becomes richer, and intuitively, it may enhance the accuracy of the TC task. The problem however is that WSD algorithms also need massive annotation data, thus they incur an overhead over the TC learning approach. But this problem can be minimized by developing WSD techniques that can be tested on some seed annotated data. Similarly, as reported in [Nigam *et al.*, 2000], the same idea of using minimal data annotated for TC was successfully applied before. To our knowledge, this is the first study on the impact WSD has on the accuracy of TC.

Assigning the meaning of a content word depends on the definition of word senses in semantic dictionaries like WordNet. There are two ways of defining the meaning of a word. First, the meaning may be explained, like in a dictionary entry. Second, the meaning may be given through other words that share the same meaning, like in a thesaurus. WordNet encodes both forms of meaning definitions. Words that share the same meaning are said to be *synonyms* and in WordNet, a set of synonym words is called a *synset*. WordNet encodes a majority of the English nouns, verbs, adjectives and adverbs (146,350 words grouped in 111,223 synsets). A word that has multiple senses belongs to many different synsets. More importantly, for each word, its senses are ordered by their frequency in the Brown corpus. This property enables the development of a simple, baseline WSD algorithm that assigns to each word its most frequent sense[5].

The most accurate current WSD algorithm [Yarowsky, 2000] uses the observation that the meaning of words is given by the context in which they are used. There are multiple ways of modeling context, ranging from the window of words surrounding the target word in the document to combining various forms of collocations with the frequency of each word sense in the entire document. The more accurate algorithms rely on sophisticated modeling of the word context, thus resulting in processing-intensive technique that add up significant overhead to the TC task. Since it is not known how much WSD impacts on accuracy of TC, we have implemented additionally to the baseline algorithm, two different WSD algorithms, of increasing complexity of the context modeling. Additionally, we used the WSD algorithm developed by the *Language Computer Corporation* (`www.languagecomputer.com`). This is an enhancement of the WSD algorithm that won the SENSEVAL competition [Kilgarriff and Rosenzweig, 2000].

### Algorithm 1: Gloss-based WSD

In WordNet, each synset is associated with a gloss that defines its meaning. For example, the gloss of the synset $\{hit, noun\}_{\#1}$ which represents the first meaning of the noun *hit* is:

*(a successful stroke in an athletic contest (especially in baseball); "he came all the way around on Williams' hit").*

[5] In WordNet the most frequent sense is the first one.

Typically, the gloss of a synset contains three different parts: (1) the definition, e.g., a *successful stroke in an athletic contest*; (2) a comment *(especially in baseball)*; and (3) an example *"he came all the way around on Williams' hit"*. Since each of these three parts can be easily distinguished by the punctuation that separates them, we process only the definition part. If we consider the gloss as a *local context*, whereas the document where the words appears as a *global context*, we could learn a semantic disambiguation function by selecting the sense whose local context (or gloss) best matches the global context. The matching is performed by considering only the nouns both in the gloss and in the document. The algorithm has the following steps:

```
1. For every $noun_i \in N_d$, the set of nouns from document d
2.     For every sense j of $noun_i$
3.          Consider $N_j$, the set of nouns from the gloss of sense j
           of $noun_i$
4.          Assign $noun_i$ the sense S such that
            $s = argmax_{j \in senses(noun_i)} |N_j \cap N_d|$
```

The algorithm models the context of a noun by considering the nouns used in their WordNet gloss for defining each of their senses. The sense, which is selected, is the one having the nouns from its gloss more frequently used in the document.

### Algorithm 2: Collocation-based WSD

Words that appear in the context of a target word are said that they collocate with the target word. There are many types of collocations, some that comprise words that are at small distance from the target word, some that involve functional relations with the target word, such as predicate-argument relationships. As we focus on semantically disambiguating only nouns, for collocations we consider two nouns to the left of the target and two nouns to its right. To find which senses the target word has, the collocation nouns are matched against the glosses of each sense. The algorithm has the following steps:

```
1. For every $noun_i \in N_d$, the set of nouns of document d
2.     Collect its noun collocations, $N_d^{-2}(i)$,$N_d^{-1}(i)$,
       $N_d^{+1}(i)$ and $N_d^{+2}(i)$ in document d
4.     Assign $noun_i$ the sense s such that
            $s = argmax_{j \in senses(noun_i)} |N_c \cap N_H^i|$
```

Where $Nc$ is the set of nouns in the collocation and $N_H^i$ is the set of nouns in the gloss of $synset(noun_i^{sense=j})$ as well as the glosses of all its hyponyms[6].

This algorithm combines the modeling of context as a collocation window and the glosses of the WordNet sub-hierarchy determined by each possible sense

[6] In WordNet, if there is an IS-A relation between synsets $s_1$-IS-A-$s_2$, then $s_1$ is called a hyponym of $s_2$ whereas $s_2$ is a hypernym of $s_1$.

of a $noun_i$ from the document. It takes into account the notion of one sense per collocation by combining collocations of the same target noun that have common components. Moreover, since the combined collocations belong to the same document, it account also for the observation of learning one sense per document or discourse.

### 3.1.5 Computational Aspects

One of target issue in operational text classification systems is the *applicability* to large scale tasks and to computationally intensive tasks (e.g., filtering and delivery of Web multimedia documents). The overall computational complexity is thus very important. we provide some details about the main technologies employed as well as their complexities.

The proposed linguistic text classification framework depends basically on two main subsystems:

- The feature extraction model that includes the linguistic processors, i.e., the basic NLP-feature extractor, the terminology extractor and the WSD algorithms.
- The text classification model that refers to the (profile) learning and to the classification components. That has already been examined in Section 2.7.3.

#### Extraction of basic NLP-features

The overall complexity of the language processor strictly depends on the employed lexical resources as well as on the processing models for two language levels: morphsyntactic and grammatical recognition. Morphological recognition is the activity of detecting the canonical lemma associated to a text unit and it is usually carried out according to extensive dictionaries combined with generative (i.e. rule based) components for expressing legal linguistic derivations. These processes are usually optimized to (almost) linear pattern matching algorithms and do not represent a real issue for complexity.

A second phase is syntactic disambiguation, i.e. POS tagging. This process has been largely studied since late eighties (i.e. [Brill, 1992; Church, 1988]). It has an important role in efficient NL processing as it reduces the complexity of later grammatical recognition. The approach adopted in our processor is inspired by [Brill, 1992] where large sets of transformational rules are applied over an ambiguous textual context in cascade. This (almost) deterministic approach has a linear complexity in the number of text windows analyzed (i.e. the number of tokens in a document). Learnability of the transformational rules also ensures the scalability to large-scale document set, lexicons and portability throughout domains.

Finally, the third step employed in the proposed TC framework is named entity recognition. This is carried out as the recognition of specific phenomena driven by possibly large scale grammars. These grammars usually differ from domain to domain although a general set of classes for named entities (e.g., location, organization and person names) are common practice. However, in general they are expressed via regular expressions that can be easily modeled by means of finite state computational devices. The real source of complexity here is thus only the size of the grammar issue that can be easily dealt with suitable optimization techniques (e.g., look-head or hashing). Again these processing stages are almost linear in the size of the input texts.

The above processing steps represent a subset of the system discussed recently in [Basili and Zanzotto, 2002], where a large scale evaluation of the parsing architecture for two languages (Italian and English) over several domains is extensively reported. A processing time[7] of about 250 words per second is the result of effective (i.e. non analytical) measures over realistic collections. Moreover, processing time increases as a linear function of the corpus size for both languages. This clearly suggests that the adopted linguistic processor is usable for large scale scenarios (e.g., hypertextual linking and Web publishing as in [Basili *et al.*, 2003]) and does not represent an obstacle to the application of the proposed TC technique.

### Terminology Extraction

The Terminology Extraction requires the following phases:

1. *Head selection*; to apply the target complex nominal grammar the candidate head of terminological expressions has to be selected. For such purpose statistical filters based-on $TF \times IDF$ are evaluated. Given $M$ documents and $m$ the maximal number of words for each documents, this step can be carried out in $O(m \times M \times log(m \times M))$ time, as previously described in Section 2.7.3 .

2. *Grammar Application*; all windows of $n$ words around the head are considered. The algorithm attempts to apply the target complex nominal grammar in such windows. All sub-sequences of words around the head that match the grammar are stored in a database. The number of subsequences to be processed is less than the number of word occurrences. Moreover, the number of window words is considered constant, thus, the application of the grammar can be done in constant time. Keeping constant the size of the word window limit the length of the possible expressions but it allows to have a linear extraction algorithm.

3. *Statistical filtering*; after the processing of all documents in the target category, statistical filter (discussed in Section 3.1.3) are applied to select the most suitable terminological expressions. Even this phase requires linear times.

The above points prove that the terminology extraction is carried out in $max\{O(m \times M \times log(m \times M)), O(k)\}$ time, where $k$ is the total occurrences of words in the

[7] This refers to an old Personal Computer Pentium II 100 Mhz. equipped with 120 Mbytes RAM.

training documents. The number of words inside the window determine the multiplicative constants of the complexity.

### WSD algorithms

We have presented 3 algorithms for WSD, all of them have to look for the word senses in the WordNet database. The searching time is $log(W)$, where $W$ is the cardinality of the set of WordNet Synsets. The final complexities are described in the following:

(0) The Baseline algorithm is very simple for each word of the target document it chooses the first sense, thus the complexity is $O(N \times log(W))$, where $N$ is the number of unique features and $log(W)$ is the time required for the binary searching of the word synset.

(1) The Algorithm 1 for each noun $n$ of the target document $d$ and for each sense $s_n$ of $n$ tests if the gloss words for the sense $s_n$ are in $d$. This requires $O(m \times k_s \times k_g) \times O(log(m))$, where:

   - $m$ is the maximum number of words in a document,
   - $k_s$ is the maximum number of senses in a synset,
   - $k_g$ is the maximum number of words in a gloss and
   - $O(log(m))$ is the time required by the binary search to verify if a gloss word is in $d$.

   $k_s$ and $k_g$ are constants, consequently, the overall complexity for the $M$ corpus documents is $O(M \times m \times log(m))$.

(2) The Algorithm 2 requires to extract the collocations that precede and follow each noun (occurrence). This can be done by simple scanning all document $d$ in $O(m)$. The collocation nouns are then matched against the gloss nouns of each noun sense (with all its hyponym hierarchy) in $O(k_s \times k_g \times m \times log(m))$. The resulting complexity is $M \times [O(m \times log(m)) + O(m)] = O(M \times m \times log(m))$. Moreover, the multiplicative constants are higher than those of the Algorithm 1 as the searching is extended to all hyponym hierarchy.

The complexity of the Algorithm 3 could not be evaluated as it refers to a complex and secret WSD system kindly made available for these experiments by the *Language Computer Corporation*. In any case to carry out the disambiguation of 12,902 documents of *Reuters-21578*, the system employed approximately one week on a PC Pentium III 300 MhZ.

## 3.2 Experiments on basic NLP-derived indexes

In this section, experiments using efficient NLP indexing techniques over efficient TC models have been carried out. The aim is to discover the most effective combination of weighting schemes, inference policies, score adjustment techniques with the basic NLP information (i.e., lemmatization, Proper Nouns and POS-tagging). For this purpose we made the following experiments:

- the accuracy evaluation for different TC models adopting the weighting schemes of Section 2.2.
- the comparison of score adjustment techniques ($LR$ vs. $RDS$)
- The comparison of the above *real* corpus performances with those obtained over traditionally employed benchmarking *test-sets*;
- The analysis of the role of linguistic information with respect to the different model for designing features;

For a large-scale evaluation, we used three different corpora: *TREVI-Reuters* and HOS, provided from users involved in the TREVI project and *Reuters3* corpus to enable the comparison with other literature work (at least with [Yang, 1999; Yang and Pedersen, 1997]). Every evaluation test has made use of microaveraged Breakeven Point ($\mu BEP$) (see Section 2.4.2), over all the target categories in the underlying corpus. The set of features employed are those described in Section 3.1.2.

Two sets of experiments have been carried out: The first aims to provide a cross-domain analysis of weighting schemes, score adjustment techniques and inference policies. Here results over "real" data (i.e. TREVI-Reuters and HOS) have been compared with those obtained over the Reuters benchmarking corpus (Experiments 1-3). Moreover, tests on the Reuters benchmark are helpful to assess the contributions of original aspects (i.e. the $IWF$ weighting model and the $RDS$ score adjustment technique) against approaches previously presented in literature. A second set of experiments aimed to evaluate the contribution of POS-tag information, which has been measured via the most accurate models determined in the first tests.

### 3.2.1 Efficient NLP on Efficient TC models

Any test has been carried out over a specific design choice among the different approaches proposed for *Feature Design and Extraction*, *Document/Profile Weighting*, *Score Adjustment* and *Inference Policy.* Almost all the proposed models in Section 2.2, 2.3 and 2.4 could be combined in a target TC architecture. In order to investigate the implementation choices as well as the impact of linguistic features, we defined a subset of possible TC architectures summarized in Table 3.1. Here, each system is defined by means of a set of characteristics listed in the respective columns. In column 1, the model name is reported. It is obtained by forming a sequence of labels in the following order:

1. *Roc* or *SMART*, i.e., the profile weighing scheme, *Rocchio* and *summing-up*, of Section 2.2 . We call the latter *SMART* as it was firstly used for SMART IR model.

2. *Scut* or *Rcut* inference policies.

3. $IDF$, $IWF$ or $log$, they indicate the document weighing schemes: $TF \times IDF$, $TF \times IWF$, and $log(TF) \times IDF$. This latter is used only in combination with Rocchio.

4. $NL$, it indicates the use or not of the linguistic information defined in Section 3.1.2. If $NL$ is not present, the model is tested over the set of *TREVI-tokens*, i.e., nouns, verbs and adjectives without the POS-tag information.

5. $LR$ or $RDS$ score adjustment techniques.

As an example, line 4 (i.e. the $SMART^{Scut}_{IWF}/_{RDS}$ system) refers to our implementation of the standard SMART $IR$ model for TC: it adopts a *summing-up* policy for profile building and the *Scut* inference policy to classify test documents. The $TF \times IWF$ (Eq. 2.4) scheme weights the feature inside the documents, no linguistic information is used and *Relative Difference Score* (Eq. 2.3.2) is applied as score adjustment technique.

Table 3.1: Text Categorization System: Experimental Parameter setting

| Systems | Profile | Inference | Document | NLP | Score |
|---|---|---|---|---|---|
| | Weighting | Policy | weighting | | Adjust. |
| $SMART^{Scut}_{IDF}/^{NL}$ | *Summing-up* | *Scut* | $TF \times IDF$ | yes | None |
| $SMART^{Scut}_{IDF}/^{NL}_{RDS}$ | *Summing-up* | *Scut* | $TF \times IDF$ | yes | $RDS$ |
| $SMART^{Scut}_{IWF}/_{RDS}$ | *Summing-up* | *Scut* | $TF \times IWF$ | no | $RDS$ |
| $SMART^{Scut}_{IWF}/^{NL}$ | *Summing-up* | *Scut* | $TF \times IWF$ | yes | None |
| $SMART^{Scut}_{IWF}/^{NL}_{RDS}$ | *Summing-up* | *Scut* | $TF \times IWF$ | yes | $RDS$ |
| $Roc^{Scut}_{log}/_{RDS}$ | *Rocchio* | *Scut* | $log(TF) \times IDF$ | no | $RDS$ |
| $Roc^{Scut}_{log}/^{NL}$ | *Rocchio* | *Scut* | $log(TF) \times IDF$ | yes | None |
| $Roc^{Scut}_{log}/^{NL}_{RDS}$ | *Rocchio* | *Scut* | $log(TF) \times IDF$ | yes | $RDS$ |
| $Roc^{Rcut}_{log}/^{NL}$ | *Rocchio* | *Rcut* | $log(TF) \times IDF$ | yes | None |
| $Roc^{Rcut}_{log}/^{NL}_{LR}$ | *Rocchio* | *Rcut* | $log(TF) \times IDF$ | yes | $LR$ |

The next section provides a cross-domain analysis of weighting schemes, score adjustment techniques and inference policies.

### 3.2.2 Experiment 1.1: Performances in TREVI-Reuters corpus

In these experiments performances of the classifiers, by adopting different weighting schemes over the *Scut* threshold policy, have been measured. The Table 3.2

Table 3.2: Classifier Performances on the TREVI-Reuters

| | $SMART_{IDF}^{Scut}/^{NL}$ | $SMART_{IDF}^{Scut}/_{RDS}^{NL}$ | $SMART_{IWF}^{Scut}/_{RDS}^{NL}$ |
|---|---|---|---|
| $\mu BEP$ | 63% | 72% | 76% |

| | $Roc_{log}^{Scut}/^{NL}$ | $Roc_{log}^{Scut}/_{RDS}^{NL}$ | |
|---|---|---|---|
| $\mu BEP$ | 62.78% | 71.60% | |

shows that, whatever is the scheme used for document ($IDF$ or IWF) and profile (*Rocchio* or $SMART$) weighting, the $RDS$ technique improves accuracy. Moreover, we observe that the two approaches to profile building (*Rocchio* or $SMART$) have the same performances. It is worth noticing that Rocchio's formula has been parameterized with standard values $\gamma = 4$ and $\beta = 16$ [Cohen and Singer, 1999]. We recall that Chapter 2 has shown that other parameterization can improve Rocchio accuracy.

Table 3.3 reports only the *Rocchio* model performances. The aim here is the comparison between the score adjustment techniques $RDS$ and $LR$. The first and second column of the Table 3.3 show the low breakeven point achieved by the models that use neither $LR$ nor $RDS$. They differ for the adopted threshold policy (*Rcut* and *Scut*). The third and fourth column assess the benefits of using the $LR$ and the $RDS$ techniques as the performances of the *Rocchio* model improve significantly. This affects especially the *Rcut* inference policy for which the cross-categorical comparison of scores is crucial.

Table 3.3: $RDS$ vs. $LR$ technique on the TREVI-Reuters

| | $Roc_{log}^{Rcut}/^{NL}$ | $Roc_{log}^{Scut}/^{NL}$ | $Roc_{log}^{Rcut}/_{LR}^{NL}$ | $Roc_{log}^{Scut}/_{RDS}^{NL}$ |
|---|---|---|---|---|
| $\mu BEP$ | 47.04% | 62.78% | 66.55% | 71.60% |

### 3.2.3 Experiment 1.2: Performances in HOS

In these experiments the best weighting models of previous section (i.e. $SMART_{IWF}$ and *Rocchio*) have been evaluated for the HOS corpus. Table 3.4 confirms the results of the previous test about the benefits of $RDS$ as for both weighting schemes it produces an increase in $\mu BEP$.

It is worth noticing that the $SMART_{IWF}$ model shows lower performances (with or without $RDS$) than *Rocchio* which is in contrast with the Experiment 1.1 where $SMART_{IWF}^{Scut}/_{RDS}^{NL}$ outperformed all models. The reason is that the weighting scheme seems to depend on different corpora. Similar issues have inspired works about Meta Text classifier in [Yang *et al.*, 2000;

Table 3.4: Classifier Performances on HOS Corpus with *Scut*

| | $Roc_{log}^{Scut}/^{NL}$ | $Roc_{log}^{Scut}/^{NL}_{RDS}$ | $SMART_{IWF}^{Scut}/^{NL}$ | $SMART_{IWF}^{Scut}/^{NL}_{RDS}$ |
|---|---|---|---|---|
| $\mu BEP$ | 64.09 % | 67.75% | 45.85% | 59.15% |

Lam and Lai, 2001], which assesses the need of integrating multiple classification models within a single text classifier architecture. Thus, some heuristics should be applied for selecting the suitable classifier for a given corpora or document.

### 3.2.4 Experiment 1.3: Assessments over the Reuters corpus

In order to compare our classifier framework with other results from the literature, evaluation against the *Reuters3* corpus has been carried out. The breakeven points are reported in Table 3.5. In line with the previous results $RDS$ produces a significant improvement on both weighting schemes. Note that the performance of the basic *Rocchio* trained with our linguistic features (column 1) is higher than other results obtained in literature (e.g., 75% in [Yang, 1999]). This suggests that the linguistic processing (i.e. the only difference among other experiments (e.g., [Yang, 1999]) and our measurement) provides additional positive information. The $SMART_{IWF}^{Scut}/^{NL}_{.}$ (”.” means for any argument) model still shows performances lower than *Rocchio*. This is due to the similar structures of HOS and Reuters corpora (on which $SMART_{IWF}^{Scut}/^{NL}_{.}$ poorly performs). They have smaller classes than the TREVI-Reuters so, in line with exactly the same observation made in [Cohen and Singer, 1999], the $Roc_{log}^{Scut}/^{NL}_{.}$ model is more robust with respect to categories, which have a poorer *training-set*.

Table 3.5: Classifier Performances on *Reuters3* Corpus

| | $Roc_{log}^{Scut}/^{NL}$ | $Roc_{log}^{Scut}/^{NL}_{RDS}$ | $SMART_{IWF}^{Scut}/^{NL}$ | $SMART_{IWF}^{Scut}/^{NL}_{RDS}$ |
|---|---|---|---|---|
| $\mu BEP$ | 78.46% | 80.52% | 62.21 | 66.80 |

### 3.2.5 Experiment 2: Part-of-Speech information

In all the above experiments, the linguistic information has been entirely taken into account in the adopted TC architecture, i.e. all lemmas, proper nouns and $POS$ information have been used for feature engineering. In order to better understand the role of $POS$ information further evaluation is needed. Accordingly, we applied the best performing classifier architecture with and without accessing POS information. Table 3.6 shows the results of this experiment for the TREVI-Reuters corpus: Column 2 reports the performances using POS information whereas Column 3 shows the performances without POS information.

Table 3.6: Syntactic Information vs. Classification Accuracy on Trevi-Reuters.

| | $SMART^{Scut}_{IWF}/^{NL}_{RDS}$ | $SMART^{Scut}_{IWF}/_{RDS}$ |
|---|---|---|
| Rec. | 83.70% | 83.02% |
| Prec. | 70.86% | 70.56% |
| $\mu BEP$ | 76.75% | 76.28% |

It has to be stressed that, in the TREVI-Reuters corpus, among the 37,069 different features only 4,089 (11%) refer to ambiguous lemmas (i.e. lemmas with more than one POS tag)[8]: in this case the amount of information introduced by POS tags (i.e. the distinction between linguistic (i.e. lemma+POS tags pairs) and non-linguistic information (i.e. $Tokens$) is rather poor and, consequently, its impact on accuracy results low.

Table 3.7 describes recall and precision of the two indexing modalities over the Reuters corpus. Here we obtained 21,975 different indexes, and only 1,801 out of them (8%) refer to lemmas with more than one POS tag.

Table 3.7: Syntactic Information vs. Classification Accuracy on Reuters

| | $Roc^{Scut}_{log}/^{NL}_{RDS}$ | $Roc^{Scut}_{log}/_{RDS}$ |
|---|---|---|
| Rec. | 80.39% | 79.91% |
| Prec. | 80.68% | 79.95% |
| $\mu BEP$ | 80.54% | 79.93% |

### 3.2.6 Discussion

The large-scale experiments provide data for analyzing three relevant aspects:

- The impact of weighting schemes on the performances of profile-based text classifiers.
- The contribution of score adjustment techniques (e.g., $RDS$) over different inference policies ($Rcut$ and $Scut$).
- The role of linguistic processing in feature extraction, selection and their contribution to TC performances.

[8] It should be noticed that lacks in the POS tagger dictionary, e.g., several technical terms, imply that a generic "unknown noun" (NN) tag is assigned. This is often used for missing or new words thus reducing the overall ambiguity.

### Weighting Schemes

The three evaluated corpora have shown that it is very difficult to find out a profile-based classifier model that is optimal over any corpus. The Rocchio's model performs better when well characterized profiles for *smaller* and more numerous classes are available. This is shown in the results of Table 3.4 and 3.5, respectively. The $IWF$ scheme is better performing on the corpus that includes very generic classes poorly characterized by the profiles (as in the TREVI-Reuters corpus, described in Table 3.2).

### The Role of RDS in TC

A first result has been that $RDS$ establishes as an effective adjustment method that improves the TC performance. In fact, it always produces a meaningful increment of the $\mu BEP$ whatever is the adopted weighting scheme. This has been shown over all the three large and heterogeneous corpora (see tables 3.2, 3.4 and 3.5).

$RDS$ improvements vary from 13% (Table 3.2) to 2% (Table 3.5) with respect to any indexing policy. In Table 3.2 the effect is exactly the same for the two weighting models, $SMART$ and *Rocchio*. This systematic behavior suggests that $RDS$ has a corpus-dependent effect proportional to the inherent limitations of the weighting model. In Table 3.3 and 3.4 the weaker weighting policy (i.e. $IWF$) receives the best contribution (4.6% and 13.3% improvement).

In Table 3.3 we also observe that the *Rcut* policy has a poor performance [9]. The performance increases when $Scut$ is used as the comparison among scores is carried out only within a class, where variability is less important. The $LR$ technique, projecting all scores on the same [0,1] interval, allows a direct comparison thus improving the system performance of about 19%.

$RDS$ is more effective than $LR$ as, from one side, it has characteristics similar to $Scut$ (i.e. applying a threshold internally to each class) and, more importantly, it summarizes cross-categorical information (i.e. direct comparison among scores $s_{di} \forall i$). An explanation for such empirical evidence, has been already discussed in Section 2.3.2 (see Table 2.3). $RDS$ allows to accept those "odd" documents that have low scores in all classes that are usually rejected by a direct application of the $Scut$ policy. The $RDS$ technique, by using the relative difference among scores, links the decision for a class to all the others, thus capturing more "information" than the $Scut$ policy.

Analogously, $LR$ projects all scores in the [0,1] range and is sensitive, via an *Rcut* policy, to the contribution of all classes. According to our extensive experimental results, we may state that, when used alone, the *Rcut* policy (although it links the decision for a class to all the others) is not effective: the adopted similarity (and weighting) models are not providing in fact comparable values. This justifies the major beneficial effects of $LR$ (+19%).

[9] This is due to the complexity of the task in our Trevi-Reuters *test-set*. In fact classes are very rich and they need more than one profile to be suitably represented.

A direct comparison between Logistic Regression and $RDS$ (see Table 3.3) shows that both are robust with respect to "high-variability" phenomena in score assignment. In both cases the transformed similarity scores depend on all classes. According to our extensive testing, this property systematically improves the $\mu BEP$ of a profile-based TC system. However, $RDS$ is more expressive as its adjustment function depends on individual scores ($s_{di}$) as well as on each document behavior. Moreover, the $RDS$ technique is simpler and more efficient to implement. $LR$ requires a more costly implementation (for estimating $\alpha_i$ and $\beta_i$) and current results suggest that its impact is weaker.

$RDS$ is a natural way to model the overall task of classification. It is more flexible than the threshold policy ($Scut$ or $Pcut$): it is less biased by the *training-set* and can be easily adaptable to dynamically changing frameworks of use. $RDS$ is independent from the document stream (i.e. the overall set of incoming data) as it applies individually to documents. $RDS$ is expected to improve (and in fact it does) the system recall, keeping the same precision if compared with other policies. $RDS$ is not influenced by the average membership scores of documents in the *training-set* (it is thus less biased by the training data). It does not fix the number of classes ($k$) to be retained for a document. $RDS$ has been shown to be more robust with respect to categories with different specificity.

### RDS and the Parameterized Rocchio Classifier

As it has been described in Section 2.6, the accuracy of Rocchio model can be highly improved by estimating the optimal $\rho$ parameter. We have studied the relation between the $\rho$ parameters and $RDS$ technique. The Figure 3.2 shows the $\mu BEP$ curves of the $Roc_{log}^{Scut}/^{NL}$ and $Roc_{log}^{Scut}/^{NL}_{RDS}$ architectures: each point is obtained by varying the $\rho$ parameter. Notice how in the first range ($0 < \rho < 1$) the $RDS$ curve is stabilizing on high $\mu BEP$ values. This higher stability makes the selection of the parameter less critical: any value is stabilized around similar performance levels. The parameter setting derived in literature (that is not well suited for TC, as discussed in [Basili and Moschitti, 2001]) is an example where bad tuning is smoothed by $RDS$.

For the optimal $\rho$ parameters $Roc_{log}^{Scut}/^{NL}$ outperforms $Roc_{log}^{Scut}/^{NL}_{RDS}$, i.e., the positive effect of parameterization (as the negative ones) is also smoothed by $RDS$. This seems to suggest to not use RDS in conjunction with $PRC$. However, Section 2.6 has shown that the estimation of $\rho$ parameter can be carried out only if the number of training documents is enough ($> 500$). Thus, when the parameter estimation of $PRC$ is not applicable, we could use $RDS$.

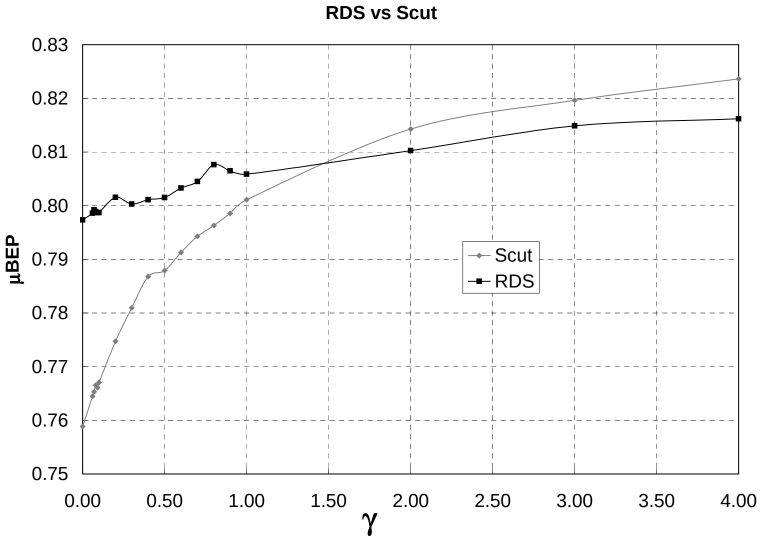


Figure 3.2: $\mu BEP$ of the $Roc_{log}^{Scut}/^{NL}$ and $Roc_{log}^{Scut}/_{RDS}^{NL}$ classifiers according to different $\rho$ values

### Analyzing the Impact of Linguistic Information

Table 3.5 reports *Rocchio* model[10] with a breakeven point of 78.46% which is relevantly higher than 75% found by Yang. Note that the only difference with those experiments is the TREVI technology used for feature extraction and selection. As discussed in Section 3.1.1, the linguistic preprocessing differs from traditional methods as (*i*) no stoplist and (*ii*) no stemming is applied, while (*iii*) recognition of proper nouns and (*iv*) POS information is available. It is evident that lemmatization and POS tagging supply information similar to that obtained via stemming and stoplist adoption: in fact, only words POS-tagged as nouns, verbs, adjectives and Proper Nouns are used for indexing.

This improvement seems suggest that the higher performances of the Rocchio's model on Reuters are related to the greater accuracy of the overall linguistic process and on the clear separation between lemmas (i.e. content words) and proper nouns.

However, other literature evaluations of Rocchio on more *difficult* Reuters versions (see Section 2.1.1) are around 78% (see Section 2.7.2), thus we cannot entirely attribute to our linguistic processing the 3.46% (78.46% vs. 75%) percent points of improvement.

The evaluation in Table 3.7 suggests that POS information, when added to

[10] It is worth to note that the Rocchio-based classifier , that we have implemented, uses the same weighting schemes adopted in [Yang, 1999]. Moreover, as the *Reuters3* corpus has been downloaded from Yang site, our results differ from the [Yang, 1999] only for the linguistic processor adoption.

the indexes, produces small improvements (see also Table 3.6). This is mainly due to the small number of truly ambiguous lemmas (10% or 8%), so that the overall effect is expected to be small.

Given the above indications a quasi-optimal version of a linguistic Profile-based Classifier can be obtained. It depends on a suitable use of document and profile weighting schemes, on the TREVI linguistic capabilities and on the $RDS$ score adjustment mechanism. We call this architecture *Language-driven Profile-based Classifier* ($LPBC$).

As it has been shown in section 3.1.5 the proposed $LPBC$ architecture has a set of suitable computational properties. It is viable even on a large scale as it has a low complexity and makes use of a robust and efficient language processor. Efficiency is also good for TC as a profile-based approach has been used. It supported an efficient processing of the test corpora in support to the different measurements requiring a very small time. This critical aspect shows the applicability of the method to operational scenarios where the number of documents requires a very high throughput.

$LPBC$ produces an accuracy on the Reuters data set of 80.52%. For its relatively simple nature and its applicability to different corpora, the $LPBC$ model was successfully adopted within the TREVI real application scenarios (i.e. users Reuters and HOS). Its good performances are retained also in the above new domains even if a simpler profile weighting model was applied (i.e. *Summing-up* in Table 4 and 5).

Section 2.6 has shown that the $PRC$ performs, using the simple $Tokens$ only, 82.83% on *Reuters-21578*, i.e. about 2.3 points (82.83% vs. 80.52%) over $LPBC$. Thus, the basic NLP does not seem improve the best Rocchio model trained with $Tokens$. In order to verify this aspect we have experimented $PRC$ on the same feature sets used in this section as well as on more advanced NLP representations, based on terminological expressions and word senses.

## 3.3 Experiments on advanced NLP-features

With the aim of studying if the NLP-derived features better impact the accuracy of TC than the simplest *bag-of-words* comparative analysis has been run. We designed three evaluation phases. Experiments in next section measure the performance of the *Rocchio* model fed with the advanced linguistic features, then, the $PRC$[11] is similarly evaluated. Different sets of features (ranging from the simplest ones (words) to the most complex (POS-tagged lemma, $PN$ and

[11] The $PRC$ interpretation claims that the optimal $\rho$ values represent the optimal feature selection for the Rocchio classifier. When richer NLP-derived representation is used, this kind of selection is more crucial, i.e. without optimal $\rho$ the extended feature set cannot be effective. Thus, when $PRC$ is fed with the richer representation it has been called the Generalized Rocchio Classifier (GRC) [Basili *et al.*, 2002; Basili and Moschitti, 2002]. The parameterization technique allows Rocchio to be a more general approach as it can be effectively trained with the linguistic features. Without the optimal $\rho$, as it is proven in what follows, Rocchio performances on NLP-features would be under the *bag-of-words*.

*TE*) have been here employed. Finally, two further collections (in Italian and English) have been used for extensive cross evaluation (Section 3.3.3).

In order to obtain the most general results we have considered two set of tokens:

- *Tokens* set defined in Section 2.1.1, which is the most general as it contains a larger number of features, e.g., numbers or string with special characters.
- *TREVI-tokens*, i.e. the nouns, verbs or adjectives. These are the tokens selected by TREVI and used in the previous section.

The NLP-feature sets are designed by adding to the above sets the NLP-information, e.g., the POS-tags to the tokens, or including the terminological expressions. The Table 3.3 summarizes the corpora information.

Table 3.8: Characteristics of Corpora used in the experiments

| Name | # Docs | # Cat | *Tokens* | *TREVI-tokens* | NLP feat. | Lang. | *test-set* Corpus % |
|---|---|---|---|---|---|---|---|
| *Reuters*3 | 11,077 | 93 | 35,000 | 19,000 | 38,000 | Eng. | 30% |
| Ohsumed | 20,000 | 23 | 42,000 | - | 42,000 | Eng. | 40% |
| ANSA | 15,000 | 8 | 55,000 | - | 60,000 | Ita. | 30% |

### 3.3.1 *PRC* for measuring different NLP-feature sets

In the following experiment, the novel sets of features described in Section 3.1.3 have been investigated according to the following distinctions:

- Proper Nouns: +PN indicates that the recognized proper nouns are used as features for the classifiers.
- Terminological Expressions (+TE), e.g., *bond issues*, *chief executive*.
- Lemmas (-POS), i.e. simple lemma without syntactic categorization, e.g., *operate*, *transform* but also the ambiguous lemmas like *check*, *stock* or *drive*.
- Lemmas augmented with their POS tags in context (+POS), e.g., *check*/N vs. *check*/V.

**+PN+TE** denotes a set obtained by adding to lemmas all features detected as Proper Nouns or terminological expressions. This results in atomic features that are simple lemmas or chunked multiwords sequences (PN or TEs), for which POS tag is neglected. Notice that due to their unambiguous nature, the POS tag is not critical for PN and TE. **+POS+PN+TE** denotes the set obtained by taking into account POS tags for lemmas, Proper Nouns and Terminological Expressions.

The $PRC$ classifiers is here, adopted to make an accurate evaluation of the improvement caused by the above feature sets. The fixed *Reuters3 test-set* has been used for estimating the performances.

In Table 3.9 the $\mu BEP$ obtained by the use of the above feature sets is reported. As baseline we use again the token set generated by TREVI system, i.e. all nouns, verbs and adjectives, i.e., the *TREVI-tokens*.

We observe that both POS-tag and terminological expressions produce improvements when included as features. The best model is the one using all the linguistic features, which increases the performance of $\sim 1.5$.

Table 3.9: Breakeven points of $PRC$ on three feature sets provided by NLP applied to *Reuters3* corpus.

| | Baseline | +PN+TE | +PN+TE+POS |
|---|---|---|---|
| $\mu BEP$ | 82.15% | 83.15% | 83.60% |

However, as our baseline has been evaluated on a subset of the $Tokens$ set, it could produce lower performance then the *bag-of-words*. To investigate this aspect, in next sections we have used the whole $Tokens$ set as initial feature set to be extended with the NLP. It contains a large number of non linguistic features, e.g., numbers or string with special characters. We expect a reduction of the positive NLP impact as many tokens cannot be correctly processed by our NLP techniques: POS-tagging, lemmatization and the complex nominal grammars could not be applicable.

### 3.3.2 The impact of basic NLP-features and Terminology.

The aim of these experiments is to measure the performance of *Rocchio* classifier based on two feature sets: $Tokens$ and the merging among $Tokens$, basic NLP-features and Terminological Expressions. These latter have been derived from the training material of each of the 93 classes: for example, in the class *acq* (i.e. *Acquisition*), among the 9,650 different features about 1,688 are represented by terminological expressions or complex Proper Nouns (17%).

The Rocchio classifier performances has been observed by systematically varying $\rho \in \{0, 1, 2, ..., 15\}$ and setting thresholds to obtain the $\mu BEP$. In Figure 3.2 the plot of $\mu BEP$s with respect to $\rho$ is shown.

Higher performances characterize the NLP-driven model for any $\rho$: this suggests an inherent superiority of such source features. The single $\rho$ can be tuned so that quasi-optimal $\mu BEP$ values are also obtained for the $Tokens$-based model (i.e. $\rho$=5). However, a different setting ($\rho$=11) allow the other (NLP) model to outperform it. The impact of the more selective $PRC$ model (i.e. $\rho$ $\forall i$) on this aspect is discussed in the next section.

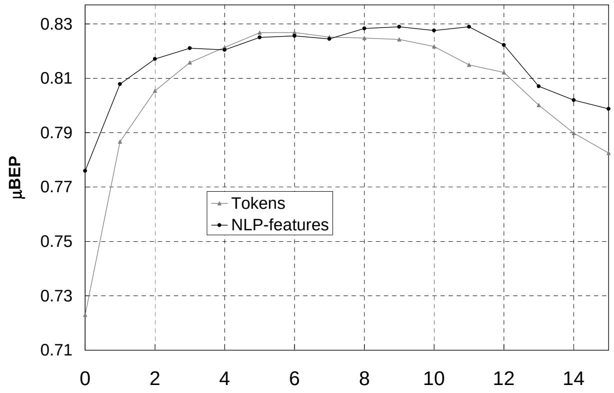


Figure 3.3: $\mu BEP$ comparisons between Rocchio classifier trained with $Tokens$ and *NLP-features*, according to different $\rho$ values.

For studying the impact of the source linguistic information on performances, independent analysis for each category has been run. Figures 3.4, 3.5, 3.6, 3.7, 3.8 and 3.9 show separate performances over some classes.

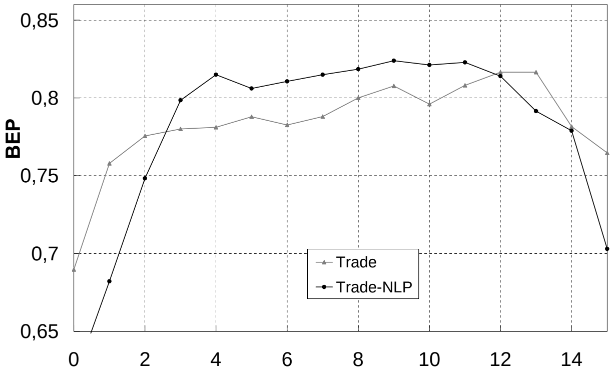


Figure 3.4: BEP of the Rocchio classifier over two feature sets (i.e. $Tokens$ and NLP-derived) according to different $\rho$ values for *Trade* category of the Reuters Corpus

It can be observed that the Rocchio takes advantage of NLP-features as slight improvement it is obtained over the $Tokens$. The richest categories in terms of the number training and test documents receive the lowest performance increase. Small categories like *Rubber* and *Dlr* are instead improved significantly by the $NLP$ process. This may suggest that poorer category profiles are modeled

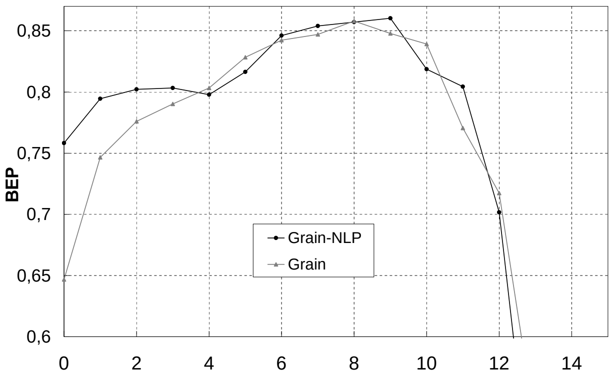


Figure 3.5: BEP of the Rocchio classifier over two feature sets (i.e. $Tokens$ and NLP-derived) according to different $\rho$ values for *Grain* category of the Reuters Corpus

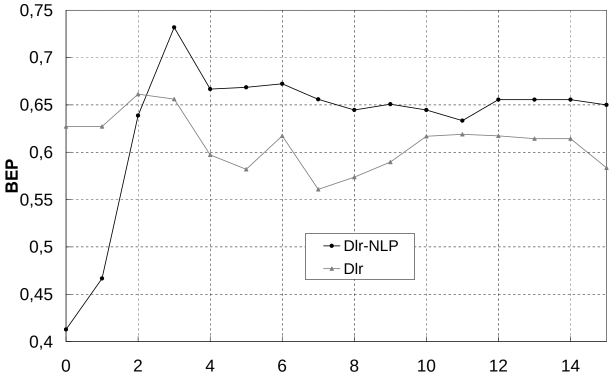


Figure 3.6: BEP of the Rocchio classifier over two feature sets (i.e. $Tokens$ and NLP-derived) according to different $\rho$ values for *Dlr* category of the Reuters Corpus

better by linguistic information.

### 3.3.3 Cross-corpora/classifier validations of NLP-features

In order to achieve the most general results cross validation has been carried out on three corpora: *Reuters3*, Ohsumed and ANSA. We have evaluated Rocchio, $PRC$ and $SVM$ classifiers to measure the impact of the NLP-features in TC. These latter were merged together with the $Tokens$ set to test if they improve the performances of the most general *bag-of-words* set.

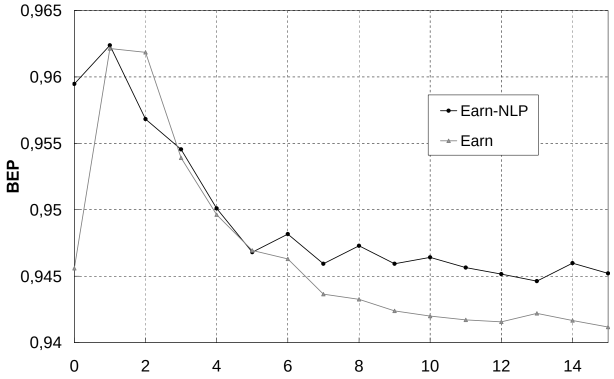


Figure 3.7: BEP of the Rocchio classifier over two feature sets (i.e. *Tokens* and NLP-derived) according to different $\rho$ values for *Earn* category of the Reuters Corpus

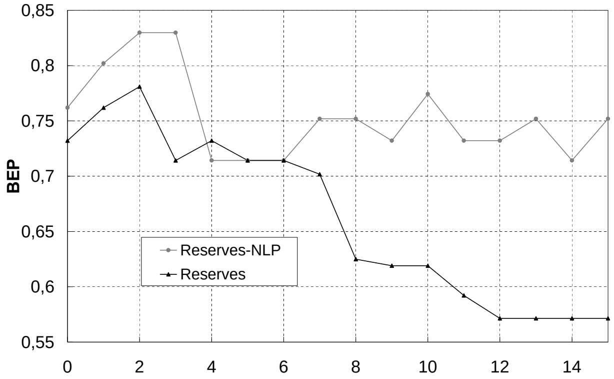


Figure 3.8: BEP of the Rocchio classifier over two feature sets (i.e. *Tokens* and NLP-derived) according to different $\rho$ values for *Reserves* category of the Reuters Corpus

For the evaluation we have adopted the same technique of Section 2.7.3 to estimate performances from several samples. Tables 3.10, 3.11 and 3.12 report the BEP, $f_1$, $\mu BEP$ and $\mu f_1$ (defined in Section 2.4.2). The accuracy of the Rocchio classifier parameterized with $\rho = .25$ has been measured by means of the BEP. Only experiments over *Tokens* are reported for Rocchio (column 2 of each table).

*PRC* has been experimented with three feature set: *Tokens*, *Tokens+TE* and *Tokens+TE+POS*. Tables 3.10 shows the uselessness of POS information

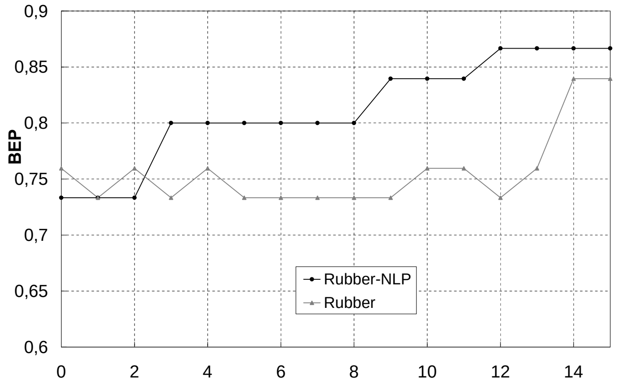


Figure 3.9: BEP of the Rocchio classifier over two feature sets (i.e. $Tokens$ and NLP-derived) according to different $\rho$ values for *Rubber* category of the Reuters Corpus

Table 3.10: Rocchio, $PRC$ and $SVM$ performances on different feature sets of the Reuters corpus

| | Rocchio | $PRC$ | | | | $SVM$ | |
|---|---|---|---|---|---|---|---|
| | Tokens | Tokens | | $+TE$ | POS+TE | Tokens | +TE |
| Category | $BEP$ | $BEP$ | $f_1$ | $f_1$ | $f_1$ | $f_1$ | |
| earn | 95.20 | 95.17 | 95.39 | 95.40 | 95.25 | 98.80 | 98.92 |
| acq | 80.91 | 86.35 | 86.12 | 87.83 | 87.46 | 96.97 | 97.18 |
| money-fx | 73.34 | 77.80 | 77.81 | 79.03 | 79.04 | 87.28 | 87.66 |
| grain | 74.71 | 88.74 | 88.34 | 87.90 | 87.89 | 91.36 | 91.44 |
| crude | 83.44 | 83.33 | 83.37 | 83.54 | 83.47 | 87.16 | 86.81 |
| trade | 73.38 | 79.39 | 78.97 | 79.72 | 79.59 | 79.13 | 81.03 |
| interest | 65.30 | 74.60 | 74.39 | 75.93 | 76.05 | 82.19 | 80.57 |
| ship | 78.21 | 82.87 | 83.17 | 83.30 | 83.42 | 88.27 | 88.99 |
| wheat | 73.15 | 89.07 | 87.91 | 87.37 | 86.76 | 83.90 | 84.25 |
| corn | 64.82 | 88.01 | 87.54 | 87.87 | 87.32 | 83.57 | 84.43 |
| MicroAv.(93 cat.) | 80.07 | 84.90 | 84.42 | 84.97 | 84.82 | 88.58 | 88.14 |
| Std. Dev. | ±0.51 | ±0.58 | ±0.52 | ±0.46 | ±0.49 | ±0.49 | ±0.47 |

for Reuters corpus as the measures in column 6 (+TE) and 7 (+POS+TE) assume similar values. SVM has been ran on simple tokens (column 7) and on terminological expressions (column 8) as they have been shown to bring more selective information in $PRC$. Similar type of measures are reported in tables 3.11 and 3.12. The global performances (microaverage) in the tables show small improvements wrt the *bag-of-words* approach (column $Tokens$) for $PRC$.

Table 3.11: Rocchio, $PRC$ and $SVM$ performances on different feature sets of the Ohsumed corpus

| | Rocchio | $PRC$ | | | | $SVM$ | |
|---|---|---|---|---|---|---|---|
| | Tokens | Tokens | | $+TE$ | | Tokens | $+TE$ |
| Category | $BEP$ | $BEP$ | $f_1$ | $f_1$ | $BEP$ | $f_1$ | |
| Pathology | 37.57 | 50.58 | 48.78 | 49.36 | 51.13 | 52.29 | 52.70 |
| Cardiovascular | 71.71 | 77.82 | 77.61 | 77.48 | 77.74 | 81.26 | 81.36 |
| Immunologic | 60.38 | 73.92 | 73.57 | 73.51 | 74.03 | 75.25 | 74.63 |
| Neoplasms | 71.34 | 79.71 | 79.48 | 79.38 | 79.77 | 81.03 | 80.81 |
| Digestive Sys. | 59.24 | 71.49 | 71.50 | 71.28 | 71.46 | 74.11 | 73.23 |
| Hemic & Lymph. | 41.06 | 65.75 | 65.80 | 65.93 | 65.85 | 63.39 | 63.39 |
| Neonatal | 41.84 | 49.98 | 50.05 | 52.83 | 52.71 | 48.55 | 51.81 |
| Skin | 47.93 | 60.59 | 60.38 | 60.53 | 60.80 | 65.97 | 64.98 |
| Nutritional | 53.23 | 60.20 | 60.08 | 60.66 | 60.75 | 71.17 | 71.34 |
| Endocrine | 39.80 | 48.76 | 44.80 | 43.96 | 48.87 | 54.24 | 53.14 |
| Disorders | 51.76 | 64.58 | 64.54 | 64.92 | 64.98 | 71.62 | 71.46 |
| Animal | 25.21 | 38.02 | 34.35 | 37.39 | 39.45 | 0 | 25.42 |
| Microaverage (23 cat.) | 54.36 | 66.06 | 65.81 | 65.90 | 66.32 | 68.43 | 68.36 |

Table 3.12: Rocchio, $PRC$ and $SVM$ performances on different feature sets of the ANSA corpus

| | Rocchio $\rho = 0.25$ | $PRC$ | | |
|---|---|---|---|---|
| | Tokens | Tokens | +TE | +POS+TE |
| Category | $BEP$ | $f_1$ | $f_1$ | $f_1$ |
| News | 50.35 | 68.99 | 68.58 | 69.30 |
| Economics | 53.22 | 76.03 | 75.21 | 75.39 |
| Foreign Economics | 67.01 | 61.72 | 61.12 | 62.37 |
| Foreign Politics | 61.00 | 75.59 | 75.32 | 76.36 |
| Economic Politics | 72.54 | 68.95 | 75.78 | 76.89 |
| Politics | 60.19 | 59.58 | 62.48 | 63.43 |
| Entertainment | 75.91 | 77.63 | 76.48 | 76.27 |
| Sport | 67.80 | 80.14 | 79.63 | 79.67 |
| Microaverage | 61.76 | 71.00 | 71.80 | 72.37 |

An explanation may be that the number of terminological expressions in these experiments is rather lower than the cardinality of $Tokens$: in Ohsumed we observed, in the feature dictionary, a ratio of about 15:1 between simple tokens and terminological expressions. This results obviously in a small impact on the microaverages.

The $SVM$ global performance are slightly penalized by the use of *NLP-derived* features. SVM seems to not need additional features derived from a combination of simpler words like phrases. If we look at the individual category performance, we observe that some classes take significant advantage from linguistic material (e.g., *Neonatal Disease & Abnormalities* in Ohsumed). The ANSA collection is more sensible to terminological information as some more specific categories, like *Politics* or *Economic Politics*, increase in BEP accuracies.

### 3.3.4 Experiments on word senses

In these experiments the performances over $Tokens$ have been compared against the performances over the semantic feature set. This latter has been obtained by merging the *Tokens* set with the set of disambiguated senses of all document nouns. We have used 3 different methods to disambiguate senses: the baseline, i.e. by picking-up the first sense, Alg1 that uses the gloss words, Alg2 that employs the notion of collocations and the Alg3 one of the most accurate commercial algorithm.

The *Reuters-21578* and 20 NewsGroups have been used to measure the accuracies. The latter was chosen as it is richer, in term of senses, than the other scientific or journalistic corpora. The performances are measured via $f_1$ for the single categories and $\mu f_1$ for the global results.

For the experiments, again, we have generated 20 splits between the training and the testing sets. For each split we have trained the classifiers and evaluated them on the test data. The performance reported in this paper is the average of all 20 splits.

Table 3.13: Performance of SVM text classifier on the Reuters corpus.

| Category | $Tokens$ | BL | Alg1 | Alg2 | Alg3 |
|---|---|---|---|---|---|
| | | | | | |
| earn | 97.70±0.31 | 97.82±0.28 | 97.86±0.29 | 97.90±0.29 | 97.68±0.29 |
| acq | 94.14±0.57 | 94.28±0.51 | 94.17±0.55 | 94.10±0.53 | 94.21±0.51 |
| money-fx | 84.68±2.42 | 84.56±2.25 | 84.46±2.18 | 84.67±2.22 | 84.57±1.25 |
| grain | 93.43±1.38 | 93.74±1.24 | 93.71±1.44 | 93.14±1.26 | 93.34±1.21 |
| crude | 86.77±1.65 | 87.49±1.50 | 87.06±1.52 | 87.30±1.67 | 87.91±1.95 |
| trade | 80.57±1.90 | 81.26±1.79 | 80.22±1.56 | 80.17±1.21 | 80.71±2.07 |
| interest | 75.74±2.27 | 76.73±2.33 | 76.28±2.16 | 76.52±2.00 | 78.60±2.34 |
| ship | 85.97±2.83 | 87.04±2.19 | 86.43±2.05 | 86.35±2.13 | 86.08±3.04 |
| wheat | 87.61±2.39 | 88.19±2.03 | 87.61±2.62 | 87.71±2.40 | 87.84±2.29 |
| corn | 85.73±3.79 | 86.36±2.86 | 85.24±3.06 | 85.40±3.00 | 85.88±2.99 |
| $\mu f_1$ (90 cat.) | 87.64±0.55 | 88.09±0.48 | 87.80±0.53 | 87.87±0.46 | 87.98±0.38 |

Table 3.13 shows the performance of $SVM$ for some categories of the Reuters corpus, measured by the $f_1$ score. *Tokens* is the usual set of tokens described in Section 2.1.2 (the adopted *bag-of-words*); BL stands for the baseline algorithm, Alg $i$ stands for Algorithm $i$. We can notice that the presence of semantic

information for each document word have globally enhanced the classifier. Surprisingly, the microaverage $f$-score ($\mu f_1$) of the baseline WSD method is higher than those of the more complex WSD algorithms. Nevertheless, the ranking among Alg1, Alg2 and Alg3 is that expected one. In fact, Alg3, i.e. the complex model of LCC, obtains an accuracy better than Alg2 and Alg1, which are simpler algorithm based on glosses. Alg2 that uses hyponym hierarchy is slightly better than Alg1. However, these are only speculative reasoning since the values of the Standard Deviations ([0.38, 0.53]) prevent a statistical assessment of our conclusions.

Table 3.14: PRC and SVM $\mu f_1$ performances on 20 NewsGroups.

| Category | *Tokens* | BL | Alg1 | Alg2 |
|---|---|---|---|---|
| SVM | 83.38±0.33 | 82.91±0.38 | 82.86±0.40 | 82.95±0.36 |

### 3.3.5 Discussion

The extensive empirical evidences provided in sections 3.3.1, 3.3.2, 3.3.3 and 3.3.4 provides themes for a wide discussion that will be attempted hereafter.

#### *Bag-of-words* results

First of all, the *PRC* model, again, produces a significant improvement in performance with respect to other proposed uses of the Rocchio formula. Tables 3.10, which provide the most general results, shows the superior accuracy of the *PRC* on *Reuters3* (80.07% vs. 84.90%). The difference of *PRC* accuracies measured on the Reuters fixed *test-set* and on cross validation is remarkable, e.g., 82.15% vs. 84.42% for the *Tokens* set. This is not due to the accuracy variability that is lower than 1% (the Std. Dev. is ~0.5 for every accuracy measures). The major reasons for such difference is the use of *TREVI-tokens* (about 19,000 features) in the experiments on *Reuters3* fixed *test-set* vs. the 35,000 (of the *Tokens* set) used for cross validation. As previously pointed out in our general performance evaluation we included numbers and strings containing special characters that helped the categorization of document containing numerical tables, e.g., many documents of the *Earn* category.

It is worth noticing that Rocchio, *PRC* and *SVM* accuracies, using the *Tokens* set over *Reuters3* corpus, are higher than those obtained, using *Tokens* set on *Reuters-21578* tested in Section 2.6 (80.07%, 84.42% and 88.58% vs. 78.92%, 82.83% and 87.64%). They differ approximately about 1 percent point. This suggests that removing the unlabeled documents [Yang, 1999] makes slightly easier the classification task.

### Syntactic information results

A second line of analysis focused on the role of syntactic information. The comparative evaluation of simpler with linguistically motivated features (carried out in the previous section) confirms the superiority of the latter (at least when $PRC$ model is used). The adoption of the effective selection and weighting method, as proposed in Equation 2.20, optimizes those meaningful features and limit the effect of sparse data often affecting linguistic approaches as derived in [Gale and Church, 1990]. This has been shown in Figures 3.4, ..., 3.9. The parameter setting of $\rho$ provides a systematic way to filter the source linguistic information. It has to be noted that in experiment +PN+POS+TE we obtained a source set of 9,650 features for the Reuters *acq* category. After $\rho_{acq}$ 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 Eq. 2.20.

The cross (corpus/language) evaluation of *linguistic performances* has added some important evidence. The results shown in Tables 3.10, 3.11 and 3.12 suggest that an improvement is always observed when linguistic features are employed in $PRC$. Although we observe a minor impact on some collection (i.e. the Ohsumed) we stably have higher results. It is worth noting that when the *TREVI-tokens* are extended by NLP-features the performance increase of ~1.5 points on the *Reuters3* fixed split (see Tab. 3.9). When the $Tokens$ set is used as basic feature set, such improvement decrease to 0.5 (see Tab. 3.10). As the ratio between terminological entries and simple tokens in the system dictionary is lowered to 1:15 the contribution of the latter is inherently weakened. Moreover, the numerical tables impact negatively on NLP-features as: (a) weakens their expressiveness, and (b) possibly caused errors in POS-tagging assignment, lemmatization and the application of complex nominal grammars.

SVM reaches high performances on many features. In a preliminary experiment with only *TREVI-tokens* over *Reuters3 test-set* we found the same accuracy ($\sim$ 85) measured in other works, e.g., [Joachims, 1998]. When the $Tokens$ were used it has increased its performance by ~2.5 percent points. On NLP-features, SVM decreases its accuracy. An explanation could be that SVM is negatively influenced by redundant features. In fact, terminological expressions contain the words already present in the $Tokens$ set and often they bring as much information as single words. For example *fetal_growth* and *early_pregnancy*, in the Neonatal category, have probably the same *indexing information* of *fetal* and *pregnancy* as single features. However, some categories (*Acq* for Reuters and $Neonatal$ for Ohsumed) show higher SVM $f_1$ when the advanced linguistic representation is adopted. We may argue that the NLP has selected relevant features as good performances are obtained even by $PRC$ on the same categories.

The syntactic NLP methods allow to includes as features $n$-grams not bound to a specific $n$. The adopted NLP use polynomial time complexity (see [Basili *et al.*, 1998c] for a description of the adopted robust parsing technique) and it selects more significant $n$-grams without overgeneration, thus limiting the size of the feature space. 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. On the contrary shallow techniques, to limit the exponential complexity of generating all possible $n$-grams, apply the selection[12] of word sequences according to minimal word frequencies. If the target word does not overcome such thresholds it cannot be part of any $n$-gram. This limits the quality of $n$-grams since relevant word sequences could contain some infrequent word. If we assume that word sequences useful for categorization are those that refer to important category concepts, the NLP derived phrases should be superior to the $n$-grams.

### Why does syntactic information not help?

NLP derived phrases seems to be superior to the *bag-of-words*, nevertheless, this section has shown that phrases produce small improvement for weak TC algorithms, i.e., Rocchio and $PRC$, and no improvement for theoretically motivated machine learning algorithm, e.g., $SVM$. The possible explanations are:

- Word information cannot be easily subsumed by the phrase information. As an example, suppose that in the target document representation *proper nouns* are used in place of their compounding words. Our task is to design a classifier that assigns documents to a *Politic* category, i.e. describing political events. The training documents could contain the feature `George_Bush` derived by the proper noun *George Bush*. If a political test document contains the `George_Bush` feature, it will have chances to be classified in the correct category. On the contrary, if the document contains only the last name of the president, i.e., *Bush*, the match of the feature `Bush` against the category feature `George_Bush` will not be enabled. In [Caropreso *et al.*, 2001] the approach of replacing the words compounding the $n$-grams with unique features has shown a decreasing of Rocchio accuracy.

- The information added by the sequence of words is very poor. Note that, a sequence of words classifies better than its compound words only if two conditions are verified:

  (a) The words of a sequence appear not sequentially in the wrong documents. For example the words *George* and *Bush* are included in a document not related to Political category.

  (b) Documents that contain the whole sequence *George Bush* are categorized in Political category.

  On one hand, the words *George Bush* is a strong indication of political category, on the other hand the single words *Bush* and *George* are not

[12] Most relevant $n$-grams can be selected by applying feature selection techniques [Caropreso *et al.*, 2001]. Even in this case the initial set of $n$-grams cannot be generated as $Tokens^n$.

related to the political category. Such situation is improbable in natural language documents, where many co-references between two referentials (in which at least one is a sequence of words) are triggered by specifying a common subsequence (e.g. *Bush* and *George Bush*). The same situation occurs frequently for the complex nominals, in which the head is used as a short referential. This suggests that terms are rarely not related to their compounding words.

- The role of phrases seem to make simpler the estimation of the TC parameters, in our case thresholds and $\rho$. For example Figure 3.3 shows that the maximal performances achieved with both $Tokens$ and $NLP$-features are approximately the same, but the convex curve is wider for the NLP-features. This allows $PRC$ a more *easy* estimation of *good* $\rho$ values. When phrases are used for $SVM$, which does not need the estimation of any critical parameter (e.g. thresholds or $\rho$), no improvement is produced.

**Semantic Information**

The experiments on WSD provide mixed results. On one hand, the sense representation obtained with the baseline WSD improves the TC accuracy using the *bag-of-words*. On the other hand, more accurate WSD algorithms does not produces better TC results than the WSD baseline algorithm. We may conclude that senses are effective for TC but these outcome should be analyzed considering our conclusive recommendations of Section 3.4.2.

In summary, NLP can be used to improve TC but the results are not impressive. Syntactic information seems to produce improvement only for *weak* TC algorithms. Semantic information still produces low improvement that enhances the best figure classifier. Next section examines the successful use of NLP in literature work.

### 3.3.6 Related Work

Previous section has revealed that NLP, especially efficient techniques, can be used to slightly improve efficient TC, i.e. profile-based classifiers. When more complex learning algorithms are used, e.g. $SVM$, only the semantic information can slightly improve the system. *Is this the role of NLP in TC?* To answer the question we examined some literature work that claim to have used language processing techniques to enhance TC. Hereafter, we attempt to explain the reasons for such successful outcomes:

- In [Furnkranz *et al.*, 1998] advanced NLP has been applied to categorize the HTML documents. The main purpose was to recognize the student home pages. For this task, the simple word *student* cannot be sufficient to obtain a high accuracy since the same word can appear, frequently, in other University pages. To overcome this problem, the AutoSlog-TS, Information Extraction system [Riloff, 1996] was applied to automatically extract syntactic patterns. For example, from the sentence *I am a student of computer science at Carnegie Mellon University*, the patterns: *I am* <->, <-> *is student*, *student of* <->, and *student at* <-> are generated. AutoSlog-TS was applied to documents collected from various computer science department and the resulting patterns were used in combination with the simple words. Two different TC models were trained with the above set of features: Rainbow, i.e. a bayesian classifier [Mitchell, 1997] and RIPPER. The positive results, reported by the authors, are higher precisions when the NLP-representation is used in place of the than *bag-of-words*. These improvements were obtained for recall lower than 20% only. The explanation was that the above NLP-patterns have low coverage, thus they can compete with the simple words only in low recall *zone*. This kind of result, even if important, cannot testify in favor of the thesis: *NLP improves TC*.

- [Mladenić and Grobelnik, 1998] reports the experiments using $n$-grams with $1 \leq n \leq 5$. These latter have been selected by using an incremental algorithm. The web pages in the Yahoo categories: *Education* and *References* were used as reference corpus. Both categories contain a sub-hierarchy of many other classes. An individual classifier was designed for each sub-category. The set of classifiers was trained with the $n$-grams contained in few training document available. The results showed that $n$-grams produce an improvement about 1 percent point (in terms of *Precision* and *Recall*) for *Reference* category and about 4 % on the *Educational* category. This latter outcome may represent a good improvement over the *bag-of-words*, but we have to consider that:

  - The experiment were done on 300 documents only, even if a cross validation was carried out.
  - The classifier adopted is *weak*, i.e. a *Bayesian* model, not very accurate. Its improvement using $n$-grams does not prove that the best figure classifier improves too.
  - The task is not standard: many sub-categories (e.g., 349 for *Educational*) and few features for each classifier. There are not other researches that have measured the performance on this specific task, i.e., it is not possible to compare the results.

  As best hypothesis we can claim that an efficient classifier (medially accurate) has been shown improving its performance by using $n$-grams. The task involved few data and many categories.

- In [Furnkranz, 1998] is reported the experimentation of $n$-grams for *Reuters-21578* and 20 NewsGroups corpora. $n$-grams were, as usual, merged with the words to improve the *bag-of-words* representation. The selection of features was done using the simple document frequency [Yang and Pedersen, 1997]. Ripper was trained with both $n$-grams and simple words. The improvement over the *bag-of-words* representation, for the Reuters corpus was less than 1%, i.e. similar to our evaluation of terminological expressions. For 20 NewsGroups no enhancement is reported.

- Other experiments of $n$-grams using Reuters corpus are reported in [Tan *et al.*, 2002]. Only bigrams were considered. Their selection is slightly different from the previous work, as Information Gain was used in combination with the document frequency. The experimented TC models were Naive Bayes and Maximum Entropy classifier [Nigam *et al.*, 1999] both fed with bigrams and words. On *Reuters-21578* the authors present an improvement of 2 % for both classifiers. The achieved accuracies were 67.07% and 68.90%[13] respectively for Naive Bayes and Maximum Entropy. What we are wondering is the following: *why to obtain an improvement using phrases have we to design TC models about 20% percent points less accurate than the best figure?* Unfortunately even the study in [Tan *et al.*, 2002] cannot be used to assess that some simple NLP-derived features as the $n$-grams, is useful for TC. A higher improvement was reported for the other experimented corpus, i.e. some *Yahoo* sub-categories. Again to validate these finding is necessary that some common corpora are adopted. This allows researchers to replicate the results. Note that it is not possible to compare the performances with [Mladenić and Grobelnik, 1998] as the set of documents and *Yahoo* categories are quite different.

- On the contrary, in [Raskutti *et al.*, 2001] were experimented bigrams using $SVM$ on the *Reuters-21578.* This enables the comparison with (a) the literature results and (b) the best figure TC. The selection algorithms that was adopted is interesting. They used the $n$-grams over characters to weight the words and the bigrams inside categories. For example, the sequence of characters *to build* produces the following 5-grams: "to bu", "o bui", "buil" and "build". The occurrences of the $n$-grams *inside* and *outside* categories were used to evaluate the $n$-gram scores in the target category. In turn $n$-gram scores are used to weight the characters of a target word. For instance, the character "o" in the word "score" in the context "to score by" receive a contribution from the 5-grams, "o scor", " score", "score", "core", and "ore b". The 5-grams scores are apportioned giving more ratio to the most centered $n$-gram, i.e. the scores are multiplied respectively by 0.05, 0.15, 0.60, 0.15, 0.5. These weights are used to select the most relevant words and bigrams. The selected sets as well as the whole set of words and bigrams were compared on *Reuters-21578* fixed *test-set.* According to the results $SVM$ improved about 0.6% when bigrams were added either to all words or to the selected words. This may be important because to our knowledge is the first improvement on SVM using phrases. However, we have to consider that:

  - No cross validation was applied. The fact that bigrams improve $SVM$ on the Reuters fixed *test-set* does not prove that they improve the

[13] Very low results as they used only the top 12 populated categories. Dumais reported for the top 10 categories a $\mu f_1$ of 92 % for SVM [Dumais *et al.*, 1998].

general $SVM$ accuracy. The major reason for the above claim is that in our cross validation over $Tokens$ (Section 2.7.3) and in [Dumais *et al.*, 1998], $SVM$ reaches an accuracy over 87%, that is higher than $\mu BEP = 86.2$ obtained by Raskutty et al. with bigrams. However, they used a larger number of categories and possibly this lowered the $\mu BEP$.

– The improvement on simple words reported in [Raskutti *et al.*, 2001] is 0.6% = 86.2% - 85.6%. If we consider that the Std. Dev. in our and other experiments [Bekkerman *et al.*, 2001] is $\sim$ 0.4, 0.6%, the improvement is not statistically sufficient to assess the superiority of the bigrams.

– Only, the words were used, special character strings and numbers were removed. As it has been proven in sections 3.3.1 and 3.3.3 they strongly affect the results by improving the unigram model. Thus the baseline could be higher than those reported (i.e. 85.6%).

According the above consideration, we can asses that on the Reuters corpus is not proven yet that phrases increase the best figure classifier accuracy. On the contrary, another corpus experimented in [Raskutti *et al.*, 2001], i.e., *ComputerSelect* shows higher $SVM$ $\mu BEP$ when bigrams are used, i.e. 6 percent points. But again the *ComputerSelect* collection is not standard. This makes difficult to replicate the results.

The above literature, favorable to the use of phrases in TC, shows that these latter do not affect the accuracy (or at least the best classifier accuracy) on the Reuters corpus. This could be related to the structure and content of its documents, as it has been pointed out in [Raskutti *et al.*, 2001]. Reuters news are written by journalists to disseminate information and hence contain precise words that are useful for classification, e.g., *grain* and *acquisition* whereas other corpora such as *Yahoo* or *ComputerSelect* categories contain words like *software* and *system*, which are useful only in context, e.g., *network software* and *array system*.

On the same line is the opinion expressed in [Bekkerman *et al.*, 2001]. They applied the Information Bottleneck (IB) feature selection technique to cluster similar features. The important idea was that a classical feature-filtering model cannot achieve good performances for the text classification problem as it is usually not related with the adopted machine learning algorithm. The IB allows to cluster words according to their relationship with categories. More precisely, it attempts to derive a good trade-off between the minimal number of word clusters and the maximum mutual information between the clusters and document categories. The information bottleneck method relates to the distributional clustering approach that has been shown not particularly useful to improve "weak" TC model performances (e.g., Naive Bayes TC). However, a more powerful TC model like SVM was shown to take advantage of word clustering techniques. Thus, $SVM$ fed with IB derived clusters was experimented on three different corpora: Reuters, WebKB and 20 NewsGroups.

Only 20 NewsGroups corpus showed an improvement of performances when IB method was used. This was explained by studying the "complexity" of the involved corpora. The above analysis revealed that Reuters and WebKB corpora require a small number of features to obtain optimal performance. The conclusion is that IB can be adopted to reduce the complexity of the problem as well as to increase the SVM performance by using a concise space representation. The improvement on 20 NewsGroups, using the cluster representation, was $\sim 3$ percent points.

In our own opinion, to correctly assess their improvement other experimentation is needed. In fact, their enhancement is related to a particular subset selection of simple word features. 15,000 features for the *bag-of-words* and 300 for the cluster representation were selected via mutual information. Other subsets of features may led to different results.

## 3.4 Conclusions

After the extensive experimentation carried out in this chapter some almost definitive conclusions can be derived about the use of NLP for improving TC accuracy. We have divided our conclusions in two parts: (a) The use of efficient NLP, i.e. the basic NLP-features and (b) the uses of more expensive technique such as phrases and word senses.

### 3.4.1 Efficient NLP for Efficient TC

In this chapter an extensive evaluation of different profile-based TC architectures has been reported. Real data (Health on line services and Reuters news agency) as well known benchmarking corpora have been used for comparative analysis. The results of such large-scale experiments allowed to systematically examine the following design choices for profile based TC:

- Two document weighting schemes ($IWF$ and $IDF$)
- Two weighting schemes for profile building (*Rocchio* and *Summing-up*)
- Two adjustment methods over similarity scores ($LR$ and $RDS$)
- Two inference methodologies (*Scut* and *Rcut*)

Data analysis has shown that different document weighing schemes can improve performance only if they are suitably combined with the related profile-weighting scheme. The best combination of them seems to depend on the nature of the target corpus. On the contrary, every corpus seems to require classification inferences depending on cross-categorical knowledge, i.e. information provided by all the categories. The improvements supplied by the $RDS$ technique and $LR$ confirms the above issues in every test.

The best text classifier combination is an original classifier sensitive to linguistic content, and characterized by a novel score adjustment method ($RDS$)

able to effectively approach the scale and dynamics of operational scenarios. The model obtained has been thus called Language-driven Profile-Based text Classifier (*LPBC*). *LPBC* exhibits good performance within linear statistical classifiers and throughout different corpora. Slightly increase of performances is also characterized by basic NLP-features that are straightforwardly integrated within the underlying statistical framework. The impact of Natural Language Processing in these experiments was based on:

- NLP functionalities that produce an inherent corpus reduction by pruning less informative units, like function words and functional expressions (e.g., *in order to*, *as well as.*.) from the candidate feature set;
- proper nouns (PN) are useful in order to determine significant complex features (e.g., *n*-grams expressing domain concepts, e.g., *bond issues*, or entities, *Shell Transport & Trading Co. PLC*);
- Lemmatization better supports feature representation: it focuses only on meaningful syntactic categories (e.g., nouns, verbs) and makes available for them canonical forms rather than stemmed strings;
- POS tagging augments the expressiveness of feature representation. It allows to better characterize the conceptual role of a feature resulting in higher retrieval precision;
- The linguistic features are declarative, so that manual validation is also viable. This can be especially useful in profile-based classifier where the category-specific features can be found in the profile itself.

The resulting *LPBC* model seems to have two appealing properties: (a) it maintains the efficiency in learning and classification typical of profile-driven system. (b) *RDS* emphasizes the linguistic information and it increases the performance to a good levels. Notice that, while *state-of-the-art* TC models are hardly applicable in operational scenarios, *LPBC* has already been used effectively within different "realistic" scenarios (e.g., the TREVI project).

### 3.4.2 The role of NLP for Text Categorization

The throughout study of the impact of advanced NLP-features on TC allows to derive these main conclusions:

First, the experimented NLP-features have a positive effect only if the number of *true* words has a high rate in the feature set. Non linguistic data highly influence the accuracy of TC and at the same time it prevents the consistency of the overall linguistic model. For instance, when *Tokens* including numbers and special strings are added to the NLP-features, these latter reduce their positive effect: high quality POS-tagging can improves TC accuracy only if the larger portion of features are words.

Second, the efficient extraction of advanced type of phrasing, i.e., terminological expressions, has been applied. The methods, for automated extraction of terminological information, extend the set of linguistic information derivable from the training texts, by making available terminological dictionaries for different target categories. The experiments of Section 3.3.3 demonstrate over three different collections (in two languages, English and Italian) that weak statistical TC models can be slightly improved with NLP tools whereas theoretical motivated text classifiers, such as $SVM$ do not receive any benefit. We have provided two possible explanations:

(a) Syntactic NLP information impact only the parameterization of TC models, i.e., they make easier the estimation of the optimal parameters. Since $SVM$ does not need critical parameterization (e.g. the setting of acceptance thresholds), its performance is not affected.

(b) $SVM$ better exploits the word representation, so redundant information is not useful for it. Phrases often bring as much information as single words. For example *fetal_growth* and *early_pregnancy* probably have the same relevance of *fetal* and *pregnancy* in the Neonatal category.

It is worth noting that some categories (e.g., *Neonatal Disease & Abnormalities* and *Acquisition*) show improvement for both $PRC$ and $SVM$ performances when linguistic features are adopted, but as we have discussed in Section 3.3.6, to prove the effectiveness of linguistic features, a more general data is needed, e.g. the improvement on $\mu f_1$.

Third, terminological expressions selected via NLP are more general than other generic $n$-grams, at least from a linguistic point of view. Thus, our experiments could be considered representative for different types of $n$-grams. As the global performance of $SVM$ over terminological expressions is shown to not improve the accuracy on the $Tokens$ features, there are few chances to obtain better results with the rougher $n$-grams.

Finally, some literature work have claimed that TC improves using $n$-grams representations. These approaches differ from the techniques that select the relevant $n$-grams set. A careful analysis has revealed that small improvements for particular corpora, classifiers and feature sets were obtained. In our own opinion if a richer representation produces a better accuracy in TC this should be verified for any feature subset and for any parameter used. Moreover, the improvement should be more or less the same on different corpora. If the enhancement is obtained only for one specific corpus, the overall impression is that the researcher has looked for finding an instance that satisfies its model rather than to design a model that satisfies all the instances. In order to avoid common pitfalls in finding useless (for TC) advanced document representations, we recommend to follow some steps in the experiments:

- Use corpora that have been already experimented for other TC researches and align your own baseline results to those reported in literature. If they differ too much something is going wrong.

- Consider the whole *bag-of-words*. If some portion is held-out, e.g., the tokens derived from numbers, the baseline could be lower than the real one. The corresponding improvements are, thus, not realistic.

- The improvement on the accuracy should be proven for the *new feature-set* or for the *new feature-set* ∪ *bag-of-words*. If the enhancement of TC model requires that feature selection is applied to the *bag-of-words* (or indirectly to the *new feature-set* ∪ *bag-of-words*), feature selection may be the responsible for the improvement. Text classifiers are not already properly parameterized for the target corpus (or *test-set*). Feature selection, sometime, has the effect to fit the corpus for the classifier default parameterization. Such effect could be neither prevented by cross-validation.

- The improvement should be obtained for the best TC figure otherwise we are simple making *stronger weak* models. Improving *weak* but efficient models can be useful if (a) the complexity of the new models do not relevantly increase and (b) the accuracy approach those of the best figure.

- Cross validation is essential to prove that a representation is better than another one. This because some feature sets could be suited for a particular split, i.e., the default parameters are suited for that particular feature proportion between test and training.

- The improvement should be at least of 3 percent point otherwise: (a) it is not really useful and (b) It may be due to the classifier parameters.

- Adopt all corpus categories. This makes more general the results as different conditions will be tested, e.g., category sizes, linguistic contents and feature distributions.

# Chapter 4

# Advanced NLP systems via Text Categorization

Chapter 3 has shown that using current NLP for general TC is not effective. On the contrary TC is often used to improve advanced NLP systems. A simple use of TC is the enrichment, with the category label, of the document presented to the final user. The TREVI system (discussed in Section 3.1.1) is such an example as its purpose is to provide as much as possible information to the final user. Other natural language oriented systems like *Yahoo.com*, use categorization schemes as a navigation method to locate the user needed data.

In this chapter we discuss three novel ways to use TC for subtasks of three important NLP applications. First, we show as TC could be used to enable the *Open Domain Information Extraction.* This latter has been approached via semantic labeling technique based on information encoded in FrameNet frames, introduced in Section 1.3.1. Sentences are labeled for frames using TC models. As frames are relevant to any new Information Extraction domain, they are used for the automatic acquisition of extraction rules for the new domains. The experimental results show that both the semantic labeling and the extraction rules enabled by the labels are generated automatically with a high precision.

Second, we present a study on a *Question/Answering* system that involves several models of question and answer categorization. Knowing the question category has the potential of enhancing a more efficient answer extraction mechanism. Moreover, the matching of the question category with the answer category allows to (1) re-rank the answers and (2) eliminate incorrect answers for improving the Q/A precision. Experimental results show the effects of question and answer categorization on the overall Question Answering performance.

Finally, we describe category-based summarization methods for fast retrieval of user information. The automatic delivery of textual information to interested users is often based on the notion of text categories. The approach generally adopted by news providers consists of categorizing news items in predefined classification schemes and, then selectively delivering information to interested consumers. We propose the use of indicative and informative summaries as explanations of the categorizer decision for the target document. The summaries are produced using the explicit information inside the category profile. This latter contains simple terms (i.e. words) as well as complex nominals and coarse event representations. Specific experiments over a medical corpus have been settled to evaluate the impact of the document explanation model on the users' comprehension of the categorization process.

This chapter is organized as follows: Section 4.1 discusses our approach to use TC for IE. Section 4.2.2 presents the use of TC for Q/A. Section 4.3 describes the summarization system that adopted TC information. Finally Section 4.4 summarizes the conclusions on using TC for NLP.

## 4.1 Text Categorization for Information Extraction

With the advent of the Internet, more and more information is available electronically. Most of the time, information on the Internet is unstructured, generated in textual form. One way of automatically identifying information of interest from the vast Internet resources is by employing Information Extraction (IE) techniques.

IE is an emerging NLP technology, whose purpose is to locate specific pieces of information called *facts* (e.g., events or finer grained data), from unstructured natural language texts. These information are used to fill some predefined database table. The current methods to extract information use linguistic motivated patterns. Typical patterns are regular expressions for which is provided a mapping to a logical form. More complex and general patterns can be obtained using semantic constraints, e.g. relations among WordNet concepts. Each topic, e.g., *bombing events* or *terrorist acts*, requires different customized pattern sets to extract the related *facts*. The construction of pattern base for new topics is a time-consuming and expensive task, thus methods to automatically generating the extraction pattern have been designed.

Categorized documents have been used to enable the unsupervised patterns extraction in AutoSlog-TS [Riloff, 1996] (see sections 1.3.1 and 3.3.6). This method allows the IE designers to save time as it generates the ranked list of patterns that can be validated quicker than the manual annotation of the extraction rule from texts. However, this type of IE is clearly domain based.

We propose an approach of *Open Domain Information Extraction* that is based on sentence categorization in semantic FrameNet[1] categories. The aim of the FrameNet project is to produce descriptions of words based on semantic frames. Semantic frames, as they have been introduced by [Fillmore, 1982], are schematic representations of situations involving various participants, properties and roles, in which a word may be typically used. This kind of information can

[1] FrameNet is a lexico-semantic database, made recently available in `www.icsi.berkeley.edu/~framenet`.

be successfully used for generating domain knowledge required for any new domain, i.e. Open-Domain Information Extraction. The corpus annotation available from FrameNet enables us (a) to design algorithms for learning the sentence categorization function in FrameNet frames and (b) once available the target frame, to define the extraction rules for any domain. Next sections describe in more details the adopted Information Extraction algorithm as well as the use of sentence categorization.

### 4.1.1 Information Extraction

IE is typically performed in three stages. First, the information need is abstracted and expressed as a structured set of inter-related categories. These structures are called templates and the categories that need to be filled with information are called slots. For example, if we want to extract information about natural disasters, we may be interested in the type of disaster, the damage produced by the disaster, in the casualties as well as in the date and location where the disasters occurred. Therefore, we may generate a template listing such categories as *DAMAGE*, *NUMBER_DEAD*, *NUMBER_INJURED*, *LOCATION* and *DATE*.

Second, as the extraction template is known, text snippets containing the information that may fill the template slots need to be identified. The recognition of textual information of interest results from pattern matching against extraction rules, which are very much dependent on the knowledge of the domain of interest. For example, if we want to extract information about natural disasters, we need to recognize (a) types of disasters, names of locations and dates; and (b) all the syntactic alternations of expressions that report to natural disasters, e.g.:
`"A tornado hit Dallas Monday at 8am."` or
`"Reports on a tornado touch down in Dallas came as early as 8 in the morning."` or
`"Two people were injured when a tornado touched down in Dallas last Monday."`

In the third phase, after information of interest is identified in the text of electronic documents, it needs to be mapped in the correct template slot. This mapping is not trivial, as rarely we can identify in the same sentence all fillers of a template.

All these phases of IE are dependent on knowledge about the events, states or entities that are of interest, also known as *domain knowledge*. Every time when the information of interest changes, new domain knowledge needs to be acquired and modeled in the extraction rules. This task is complex, as it has been reported in [Riloff and Jones, 1999; Harabagiu and Maiorano, 2000; Yangarber *et al.*, 2000; Basili *et al.*, 2000c], and it requires both high quality seed examples and texts relevant to the extraction domain. The two limitations hinder the extension of IE techniques to virtually any topic of interest, or Open-Domain IE. To solve this problem we have considered the knowledge extracted from FrameNet that can be used to model any new domain. Next section describes in more detail such above information.

### 4.1.2 Semantic Frames

The Semantic Frames available from FrameNet are in some way similar to efforts made to describe the argument structures of lexical items in terms of case-roles or thematic-roles. However, in FrameNet, the role names, which are called Frame Elements (FEs) are local to particular frame structures. Some of these FEs are quite general, e.g., *AGENT*, *PHENOMENON*, *PURPOSE* or *REASON*, while others are specific to a small family of lexical items, e.g., *EXPERIENCER* for *Emotion* words or *INTERLOCUTOR* for Communication words. Most of the frames have a combination of FEs, some are general, and some are specific. For example, the FEs of the Arriving frame are *THEME*, *SOURCE*, *GOAL* and *DATE*. They are defined in the following way: the *THEME* represents the object which moves; the *SOURCE* is the starting point of the motion; the *PATH* is a description of the motion trajectory which is neither a *SOURCE* nor a *GOAL*; the *GOAL* is the expression which tells where the theme ends up.

A frame has also a description that defines the relations holding between its FEs, which is called the *scene* of the frame. For example, the scene of Arriving is: the *THEME* moves in the direction of the *GOAL*, starting at the *SOURCE* along a *PATH*. Additionally, FrameNet contains annotations in the British National Corpus (BNC) of examples of words that evoke each of the frames. Such words are called *target words*, and they may be nouns, verbs or adjectives. Although all these three major lexical categories can be frame bearing, the most prominent semantic frame evoked in a particular sentence is usually one evoked by a verb. For example, the target words evoking the Arriving frame are: approach(v), arrival(v), arrive(v), come(v), enter(v), entrance(n), return(n), return(v), visit(n) and visit(v) [2].

S1: [Yorke]$^{\text{FE=THEME}}_{\text{PT=NP GF=Ext}}$ [returning]$_{\text{TARGET}}$ [home]$^{\text{FE=GOAL}}_{\text{PT=AVP GF=Comp}}$
[from a charity event]$^{\text{FE=SOURCE}}_{\text{PT=PP GF=Comp}}$ at 2am, the city's magistrates heard.

S2: [Returning]$_{\text{TARGET}}$ [across the square]$^{\text{FE=PATH}}_{\text{PT=PP GF=Comp}}$ [she]$^{\text{FE=THEME}}_{\text{PT=NP GF=Ext}}$
felt she was going home; not for one moment did she confuse such a place
with the Aber House Hotel.

S3: You heard [she]$^{\text{FE=THEME}}_{\text{PT=NP GF=Ext}}$ [returned]$_{\text{TARGET}}$ [heartlessly]$^{\text{FE=MANNER}}_{\text{PT=AVP GF=Comp}}$.

Figure 4.1: Example of sentences mapped in FrameNet.

[2] *n* stands for noun and v stands for verb.

In FrameNet the annotations seek to exemplify the whole range of syntactic and semantic dependencies that the target word exhibit with any possible filler of a FE. For example, Figure 4.1 shows four FrameNet annotations corresponding to the verb *return*. The FrameNet tagset used to annotate the BNC sentences contain different tags, which were described in [Johnson and Fillmore, 2000]. In our experiments we relied only on these tags: (1) the *target word* (*TARGET*); (2) the *phrase type* (PT); and (3) the grammatical function (GF). The first sentence illustrated in Figure 4.1 has annotations for the *THEME*, *GOAL* and *SOURCE* FEs, whereas the second sentence has an annotation for the *PATH* frame element. The annotations from Figure 4.1 also use different possible values from the phrase type (PT) tags and the grammatical function (GF) tag. These values are listed in Tables 4.1 and 4.2. Sentence S3 contains an annotation for *MANNER*. Figure 4.2 illustrates a part of the FrameNet hierarchy. Sometimes multiple frames have the same FEs, e.g., the Arriving and Departing frames, but their *scenes* contrast their semantic interpretation.

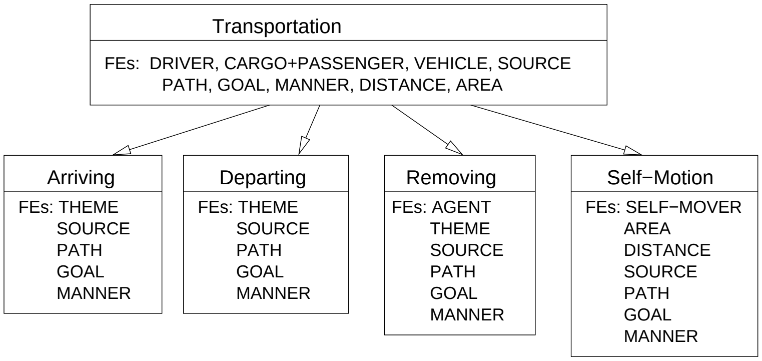


Figure 4.2: Hierarchical structuring of the Motion domain in FrameNet.

The FrameNet structures and their annotations can be used for extracting information in a topic that relates to the domains they encode. To experiment with the usage of FrameNet for IE, we have employed the extraction definitions used in the Hub-4 Event'99 evaluations [Hirschman *et al.*, 1999]. The purpose of this extraction task was to capture information on certain newsworthy classes of events, e.g., natural disasters, deaths, bombings, elections, financial fluctuations. Extraction tasks do not use frames, but instead they produce results in the form of templates. For example, let us consider the template devised for capturing the movement of people from one location to another. Individual templates were generated for fifteen different generic events.

We have used these templates for studying ways of mapping their slots into FEs of FrameNet frames. We have noticed that one Event'99 template is generally mapped into multiple FrameNet frames. The slots of the template are: *PERSON*, *FROM_LOCATION*, *TO_LOCATION* and *DATE*. Figure 4.3 illus-

Table 4.1: Phrase types annotated in FrameNet

| **Label** | **Phrase Type Description** |
|---|---|
| NP | Noun Phrase (*the witness)* |
| N | Non-maximal nominal (personal *chat*) |
| Poss | Possessive NP (*the child's* decision |
| There | Expletive *there* (*there* was a fight) |
| It | Expletive *it* (*it*'s nice that you came) |
| PP | Prepositional phrase (look at *me*) |
| Ping | PP with gerundive object (keep *from laughing*) |
| Part | Particle (look it *up*) |
| VPfin | Finite verb phrase (we *ate fish*) |
| VPbrst | Bare stem VP (let us *eat fish*) |
| VPto | To-marked infinitive VP (we want to *eat fish*) |
| VPwh | WH-VP (we know *how to win*) |
| VPing | Gerundive VP (we like *winning*) |
| Sfin | Finite clause (it's nice *that you came*) |
| Swh | WH-clause (ask *who won*) |
| Sif | *If/whether* clause (ask *if we won*) |
| Sing | ve clause (we saw *them running*) |
| Sto | To-marked clause (we want *them to win*) |
| Sforto | *For-to* marked clause (we would like *for them to win*) |
| Sbrst | Bare stem clause (we insist *that they win*) |

trates a mapping from the slots of this template to the FEs of two different frames encoded in FrameNet.

In our experiments we have manually produced the mappings. Since mappings are possible from any given template to FEs encoded in FrameNet, we developed a five-step procedure of acquiring domain information in the form of extraction rules for any topic. The procedure is:

```
Open-domain Information Extraction (Template)
1. Map Template slots into the FEs of frames from FrameNet.
2. Given a text, label each sentence either with $F_A$, if it
   contains information from the domain of frame A, or with $\phi$.
3. In each labeled sentence identify:
        3.1 the target word
        3.2 instantiations of FEs from frame A
4. For each verb identified as
      (a) target word or in a Subject-Verb-Object dependency
          with the target word; or
      (b) in a FE instantiation
   collect all Subject-Verb-Object triplets as well as all
   the prepositional attachments of the verb;
```

Table 4.2: Grammatical functions annotated in FrameNet

| Label | Grammatical Function Description |
|---|---|
| Ext | *External argument* (Argument outside phrase headed by target verb, adjective or noun) |
| Comp | *Complement* (Argument inside phrase headed by target verb, adjective or noun) |
| Mod | *Modifier* (Non-argument expressing FE of target verb, adj. or noun) |
| Xtrap | *Extraposed* (Verbal or clausal compl. extraposed to the end of VP) |
| Obj | *Object* (Post-verbal argument; passivizable or not alternate with PP) |
| Pred | *Predicate* (Secondary predicate compl. of target verb or adjective) |
| Head | *Head* (Head nominal in attributive use of target adjective) |
| Gen | *Genitive determiner* (Genitive determ. of nominal headed by target) |

5. *Generate extraction rules for the topic.*

The result of this procedure is that we obtain as many extraction rules as many different verbs we have identified. Their subjects, objects and prepositional objects are matched by any noun groups having the head in the same semantic category as those learned at training time from the FrameNet annotations. Central to this procedure is step 2, which identifies relevant sentences. Based on this categorization, we can perform step 3 with high precision, in a second labeling pass.

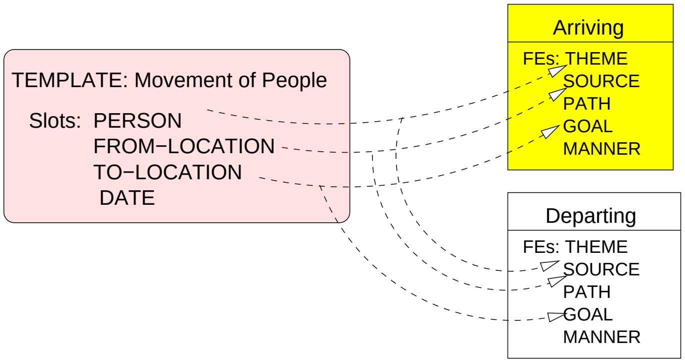


Figure 4.3: Mappings from an extraction template to multiple frames.

### 4.1.3 Semantic Labeling via TC models

The first pass of labeling concerns identifying whether a sentence contains information pertaining to a frame encoded in FrameNet or not. It is possible that a sentence is relevant to two or multiple frames, thus it will have two or multiple labels. In the second pass text snippets containing a target word and the instantiation of a frame elements are detected.

#### Sentence labeling

The problem of semantic labeling of sentences is cast as a classification problem that can be trained on the BNC sentences annotated in FrameNet.

To implement the classifier we have chosen the Support Vector Machine (SVM) model as previous chapters have known that it generally obtains high classification accuracy. Moreover, its learning algorithm allows to generalize well without requiring high quality training data [Vapnik, 1995]. In our SVM-based procedure we have considered the following set of features: each distinct word from the training set represents a distinct feature; additionally, each distinct <*Phrase Type - Grammatical function*> pair (<*PT-GT*>) that is annotated in the training set represents a distinct feature. In our experiments, we have used 14,529 sentences (60% of the corpus) containing 31,471 unique words and 53 distinct <*PT-GF*> pairs. The total number of features was $N$=31,524. The sentences were selected from the FrameNet examples corresponding to 77 frames.

For each frame $F_\alpha$ we have trained a different classifier $C_\alpha$. Considering each sentence $s$ from the training corpus $T_S$, we generate the Vector Space Model for the sentences. The dimensions are all the features $F = \{f_1, f_2, ..., f_N\}$ inside the sentences of $T_S$, and the sentences $s$ are represented as vectors of weights $\vec{w_s} =< w^s_{f_1}, ..., w^s_{f_n} >$. Similarly to the document weighting strategy of Section 2.2, we evaluate the weights for each feature $f$ observed in the sentence using:

- $M_s$, the number of sentences in $T_S$,
- $M_{sf}$, the number of sentences in which the features $f$ appears and
- $o^s_f$, the occurrences of the features $f$ in the sentence $s$.

Accordingly, the sentence weights are:

$$w^s_f = \frac{l^s_f \times ISF(f)}{\sqrt{\sum_{r \in F}(l^s_r \times ISF(r))^2}} \tag{4.1}$$

where $l^s_f$ is defined as

$$l^s_f = \begin{cases} 0 & \text{if } o^s_f = 0 \\ log(o^s_f) + 1 & \text{otherwise} \end{cases} \tag{4.2}$$

and the $ISF(f)$ is the *Inverse Sentence Frequency* evaluated similarly to the $IDF$, i.e., $log\frac{M_s}{M_{sf}}$. In other words, we adopted for the sentences the same weighting schemes used for the documents, by considering sentences as small documents.

To classify new sentences $s'$ for a frame $F_\alpha$ with $SVM$ we need to learn the gradient vector $\vec{a}$ and the threshold $b$ (see Section 2.5.3). This can be done by solving the equation 2.19 for the new Vector Space Model, i.e.:

$$\begin{cases} Min \quad |\vec{a}| \\ \vec{a} \times \vec{w_s} + b \geq 1 \quad \forall s \in T_c \ \ labeled \ \ for \ \ F_\alpha \\ \vec{a} \times \vec{w_s} + b \leq -1 \ \ \forall s \in T_c \ \ not \ \ labeled \ \ for \ \ F_\alpha \end{cases}$$

The SVM classifier $C_\alpha$ for the frame $F_\alpha$ applies the signum function ($sgn$) to the linear function $l_\alpha = \vec{a} \times \vec{w_s} + b$, i.e., $C_\alpha(\vec{s}) = sgn(l_a(\vec{s}))$. A sentence $s'$ is labeled for $F_\alpha$ only if $C_\alpha(\vec{w_{s'}}) = 1$.

The above classification algorithm requires two type of features: words and the pairs <*PT-GF*>. The former can be extracted with the usual TC techniques whereas for the latter we need some heuristics that discover the probable target word with its phrase type and grammatical function. Next section defines some heuristic that can be used for this second task.

### Refining Semantic Labels

For the purpose of open-domain IE, we need to know additionally which text snippets from a sentence stand for (a) a target word and (b) an instantiation of a frame element.

To identify the target words we simply collected all the words that evoke each frame and implemented a two-step procedure: (1) recognize any of these words in the text sentence; (2) if a word could not be recognized, rank all sentence words by semantic similarity to the evoking words and select the highest ranking word. Semantic similarity is computed with the same procedure employed for generating lexical chains as reported in [Barzilay and Elhadad, 1997].

The recognition of FE boundaries is based on a set of heuristics. For example, for the ARRIVING frame, we used a set of 4 heuristics. To describe them, we call *siblings* two phrases that have the same parent in the syntactic parse tree of the sentence being analyzed.

- *Heuristic 1* An instantiation of an FE is recognized as an adverbial phrase (ADVP) if:

  (a) The ADVP is a sibling of the target word;

  (b) The head of the ADVP identifies a physical location;

  For example, in the sentence:
  `"Amy arrived home from school early one afternoon."`,
  Heuristic 1 recognizes [home] as an instantiation of a FE because it is labeled as ADVP by the parser, it is a sibling of the target word *arrive* since they have a common parent (VP) and *home* is a location.

- *Heuristic 2* An instantiation of an FE is recognized as a verb phrase (VP) if:

  (a) The VP is a sibling of the target verb;

  (b) The VP's head is a gerund verb;

  For example, in the sentence `"The Princess of Wales arrived smiling and dancing at a Christmas concert last night."`,
  Heuristic 2 recognizes the verb phrase `"smiling and dancing"` as a FE instantiation because its head is a gerund verb and a sibling of the target word *arrived*.

- *Heuristic 3* An instantiation of an FE is recognized as a prepositional phrase (PP) if its leading preposition belongs to this list: *from*, *to via*, *through*, *in*, *at*, *on*, *at*, *of*, *towards* or *by*, and one of the following three conditions is true:

  (a) The PP is a sibling of the target word;

  (b) The target word is verbal and the PP is a child of one of its siblings; in one of the following;

  (c) The target word is nominal and the PP is a sibling of its parent.

  In the previous example, Heuristic 3 recognizes the prepositional phrase `"at a Christmas concert last night"` because it is a sibling of the target word and its preposition is *at*.

- *Heuristic 4* An instantiation of an FE is recognized as a noun phrase (NP) or a wh-phrase (WHNP) [3] if:

  (a) The right-end of the NP or wh-phrase precedes the target word and;

  (b) The NP or wh-phrase are siblings of an ancestor of the target word in the parse tree;

  (c) The NP or the wh-phrase is connected to the target word in the parse tree only through S, SBAR, VP or NP nodes. The NP nodes are allowed only if the target word is of a gerund.

  (d) The NP or the wh-phrase is the top-most and right-most phrase of these types that satisfy conditions (a), (b) and (c).

For example, in the sentence `"The first of the former concentration camp prisoners and their families will start arriving from the war-torn former Yugoslav republic within days"`,
Heuristic 4 recognizes the noun phrase `"The first of the former concentration camp prisoners and their families"` as an instantiation of a FE.

Once the boundaries have been discovered it is possible extract the pair <*PT-GF*> for the target word. Then the sentence classification algorithm is applied to determine the most suitable FrameNet frame for the sentence. Finally, the frame provides the information extraction patterns given the mapping between FrameNet and the target domains.

[3] a wh-phrase contains a relative pronoun like *who*, *what* or *which*

### 4.1.4 Experiments

The quality of the extraction rules required for any new domain depends on the accuracy with which sentences are labeled with semantic frames relevant to the domain. In our experiments, we measured the performance of sentence categorization in the same way it has been done for TC:

(a) the *Precision*, defined as the ratio between the number of correctly labeled sentences (by $C_\alpha$) for a frame $F_\alpha$ over the number of sentences processed;

(b) *the Recall* defined as the ratio between the number of sentences correctly labeled with a frame $F_\alpha$ (by $C_\alpha$) over the number of sentences processed that were labeled (by annotators) for $F_\alpha$.

(c) The *combined f-measure* defined as $f_1$ and the $\mu f_1$, by using equations 2.11, 2.12, 2.13 and 2.15.

In our experiments we have used 9687 sentences (40% of the corpus as test collection) from FrameNet annotations. Table 4.3 shows the result of our first pass of the sentence semantic labeling. The table shows the performance of SVM classifiers for 10 frames that had the largest number of examples annotated in FrameNet. Precision ranges between 73% and 90%, depending on the semantic frame, whereas recall ranges from 55% to 89%. In addition; to measure the average performance of the classifiers, we have computed the microaverage measures.

The results[4] listed in Table 4.3 show that the $\mu f_1$ of 80.94% distributed for the entire experiment involving 10 frames. It is close to the $f_1$ for some of the best-classified frames that lend the largest number of annotations in FrameNet, i.e. JUDGMENT, MENTAL PROPERTY OR PERCEPTION-NOISE

In each sentence labeled for a frame $F_\alpha$, we also identify (a) the target word and (b) the boundaries of the FEs that account for the semantic information pertaining $F_\alpha$. For this purpose we have employed 14 heuristics, many of them applicable across frames that share the same FE. In our experiments, the precision of identification of FEs was 92% while the recall was 78%. When 5624 sentences were processed for the following frames: SELF-MOTION, ARRIVING, DEPARTING and TRANSPORTATION, that we called MOVEMENT-Frames. From the sentences annotated for MOVEMENT-Frames, we have identified 285 verbs, called MOVEMENT-verbs, out of which 158 were target words whereas 127 are verbs identified in the boundaries of FEs. We have identified in the parse trees of the sentences labeled by MOVEMENT-Frames 285 Subject-Verb-Object triplets.

When applying these new extraction rules to the text evaluated in Event-99, they identified relevant text snippets with a precision of 82% and recall of 58%, thus an $f_1$ of 68%. This result is important because, as reported in [Yangarber *et al.*, 2000], if extraction rules perform with high precision, more rules can be

[4] In these experiments, we have used the *SVM* implementation from the Rainbow package [McCallum, 1996].

Table 4.3: Performance of SVM classifier on frame assignment

| Name | Recall | Precision | $f_1$ |
|---|---|---|---|
| self-motion | 89.74 | 87.81 | 88.76 |
| statement | 77.67 | 80.26 | 78.94 |
| judgment | 83.16 | 87.36 | 85.21 |
| perception_noise | 75.62 | 87.18 | 80.99 |
| experiencer-obj | 60.93 | 80.59 | 69.39 |
| body-movement | 68.56 | 81.95 | 74.66 |
| communication_noise | 68.74 | 73.90 | 71.23 |
| placing | 58.06 | 76.99 | 66.20 |
| mental-property | 79.72 | 90.81 | 84.90 |
| leadership | 55.89 | 79.74 | 65.72 |
| MicroAverage ($\mu$) | 77.71 | 84.46 | 80.94 |

learned, thus enhancing the recall. Additionally, the high precision of detecting boundaries of FEs is an essential pre-requisite of semantic parsing of texts, as reported in [Gildea and Jurasky, 2002]. To our knowledge, this identification is performed manually in current semantic parsers.

This section has shown an original way to exploit text categorization for an important NLP task, such as IE. The key concept was the use of text categorization algorithm to associate semantic information to sentences in open texts. The most important contribution is that small text fragments, such as the sentences, can be classified in the same way of documents with a high accuracy. This idea will be used in the next section for Question Answering systems. The challenge here is tougher as we enable the classification of even smaller text fragments, i.e. the questions.

## 4.2 Text Categorization for Question Answering

One method of retrieving information from vast document collections is by using textual *Question/Answering.* Q/A is an Information Retrieval (IR) paradigm that returns a short list of answers, extracted from relevant documents, to a question formulated in natural language. Another, different method of finding the desired information is by navigating along subject categories assigned hierarchically to groups of documents, in a style made popular by *Yahoo.com* among others. When the defined category is reached, documents are inspected and the information is eventually retrieved.

Q/A systems incorporate a paragraph retrieval engine, to find paragraphs that contain candidate answers, as reported in [Clark *et al.*, 1999; Pasca and Harabagiu, 2001]. To our knowledge no information on the text category of these paragraphs is currently employed in any of the Q/A systems. Instead, semantic information, e.g., the class of the expected answers, derived from the question processing, is used to retrieve paragraphs and later to extract answers. Typically, the semantic classes of answers are organized in hierarchical ontologies and do not relate in any way to semantic categories typically associated with documents. The ontology of expected answer classes contains concepts like PERSON, LOCATION or PRODUCT, whereas categories associated with documents are more similar to topics than concepts, e.g., acquisitions, trading or earnings. Given that categories indicate a different semantic information than the classes of the expected answers, we argue in this paper that text categories can be used for improving the quality of textual Q/A.

In fact, we show that by automatically assigning categories to both questions and texts, we are able to filter out many incorrect answers and moreover to improve the ranking of answers produced by Q/A systems.

### 4.2.1 Textual Question Answering

The typical architecture of a Q/A system is illustrated in Figure 4.4. Given a question, it is first processed for determining (a) the semantic class of the expected answer, (b) what keywords constitute the queries used for retrieving relevant paragraphs. Question processing relies on external resources for identifying the class of the expected answer, typically in the form of semantic ontologies (Answer Type Ontology). The semantic class of the expected answer is later used to (a) filter out paragraphs that do not contain any word that can be cast in the same class as the expected answer, and (b) locate and extract the answers from the paragraphs. Finally, the answers are extracted and ranked based on their unification with the question.

#### Question Processing

To determine what a question asks about, several forms of information can be used. Since questions are expressed in natural language, sometimes their stems, e.g., *who*, *what* or *where* indicate the semantic class of the expected answer,

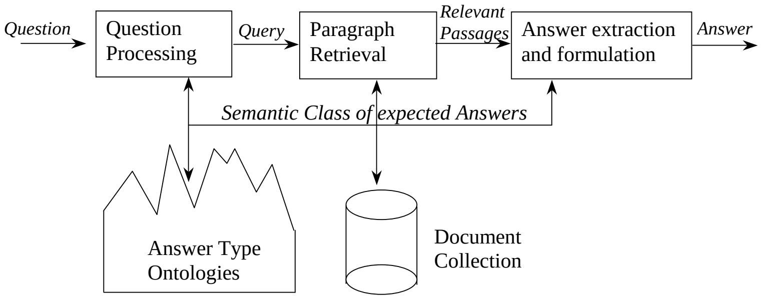


Figure 4.4: Architecture of a Q/A system.

i.e. PERSON, ORGANIZATION or LOCATION, respectively. To identify words that belong to such semantic classes, Name Entity (NE) recognizers are used, since most of these words represent names. NE Recognition is a natural language technology that identifies names of people, organizations, locations and dates or monetary values.

However, most of the time the question stems are either ambiguous or they simply do not exist. For example, questions having *what* as their stem may ask about anything and thus (1) another word from the question needs to be used for determining the semantic class of the expected answer; and (2) that word must be semantically classified against an ontology of semantic classes. To determine which word indicates the semantic class of the expected answer, the syntactic dependencies between the question words may be employed. By using any of the syntactic parsers publically available, e.g., [Charniak, 2000; Collins, 1997], the binary dependencies between the head of each phrase can be captured.

The formulation of questions typically uses $w_a$, the head of the first phrase from left to right that has the most binary dependencies as the word indicating the semantics of the answer. This result was previously reported in [Harabagiu *et al.*, 2000; Pasca and Harabagiu, 2001; Harabagiu *et al.*, 2001]. To find the semantic class of the answer, the word $w_a$ is identified in an ontology of possible classes of answers, comprising hierarchies of nouns and verbs imported from WordNet database [Fellbaum, 1998]. Such ontologies encode thousands of words, but (1) do not necessarily cover all the English words; or (2) sometimes miss-classify words because of the semantic ambiguity words have. Consequently, sometimes the semantic class of the expected answers cannot be identified, e.g., in the former case or is erroneously identified, e.g., in the latter case.

The above failure can cause some errors in retrieval the correct answers. The use of text classification aims to filter out the final set of answers that Q/A systems provide.

### Paragraph Retrieval

Once the question processing has chosen the relevant keywords of questions, some term expansion techniques are applied: all nouns and adjectives as well as morphological variations of nouns are inserted in a list. To find the morphological variations of the nouns, we used the CELEX [Baayen *et al.*, 1995] database. The list of expanded keywords is then used in the boolean version of the SMART system to retrieve paragraphs relevant to the target question. Paragraph retrieval is preferred over full document retrieval because (a) it is assumed that the answer is more likely to be found in a small text containing the question keywords and at least one other word that may be the exact answer; and (b) it is easier to process syntactically and semantically a small text window for unification with the question than processing a full document.

### Answer Extraction

The procedure for answer extraction that we used is reported in [Pasca and Harabagiu, 2001], it has 3 steps:

**Sentence-length Answer Extraction:**

*Step 1: Identification of Relevant Sentences:*
Knowledge about the semantic class of the expected answer generates two cases:

> Case 1 When the semantic class of the expected answers is known, all sentences from each paragraph that contain a word identified by the Named Entity recognizer as having the same semantic classes as the expected answers are extracted.
>
> Case 2 The semantic class of the expected answer is not known, therefore all sentences that contain at least one of the keywords used for paragraph retrieval are selected.

*Step 2: Sentence Ranking:*
We compute the sentence ranks as a by product of sorting the selected sentences. To sort the sentences, we may use any sorting algorithm, e.g., the quicksort, given that we provide a comparison function between each pair of sentences. To learn the comparison function we use a simple neural network, namely, the perceptron, to compute a relative comparison more between any two sentences. This score is computed by considering four different features for each sentence $S$:

> $f_1^s$ = number of question words matched in sentence $S$
>
> $f_2^s$ = number of question words that are matched in a window of ±5 words from the word having the same semantic class as the expected answer.
>
> $f_3^s$ = number of words occurring in the same order both in the question and in the sentence.

$f_4^s$ = the average distance between each question word and the sentence word having the same semantic class as the expected answer.

*Step 3: Answer Extraction* We select the top 5 sentences that are ranked and return them as answers. If we lead fewer than 5 sentences to select from, we return all of them.

Once the answers are extracted we can apply an additional filter based on text categories. The idea is to match the categories of the answers against those of the questions. Next section addresses the problem of question and answer categorization.

### 4.2.2 Text and Question Categorization

To exploit category information for Q/A we categorize both answers and questions. For the former, we define as categories of an answer *a* the categories of the document that contain *a*. For the latter, the problem is more critical as it is not clear what can be considered as categories of a question.

To define question categories we assume that users have in mind a specific domain when they formulate their requests. Although, this can be considered a strong assumption, it is verified in practical cases. In fact, if a question is sound it implies that the questioner knows some basic concepts about the application domains. As an example consider a random question from TREC-9[5]:

`"How much folic acid should an expectant mother get daily?"`

The concept *folic acid* and *get daily* are related to the concept *expectant mother* as medical experts prescribe such substance to pregnant woman with a certain frequency. The hypothesis that the questioner has randomly generated this question without knowing the relations among the question concepts is unlikely. In turn, specific relations are typical of the application domains, i.e. they often characterize domains. Thus, the user by referring to some relations automatically refers to some specific domains (categories). In summary, the idea of question categorization is (a) users cannot formulate a consistent question on a domain that do not know, and (b) specific questions that express relation among concepts automatically define domains.

Moreover, the specificity of the questions depends on the categorization schemes which documents are divided in. For example the following TREC question:

`"What was the name of the first Russian astronaut to do a spacewalk?"`

may be considered generic, but if the categorization scheme include categories like *Space Conquest History* or *Astronaut and Spaceship* the above question is clearly specific of the above categories.

The same rationale cannot be applied to very short questions like: `Where is Belize located?`, `Who invented the paper clip?` or `How far away`

[5] TREC-9 questions are available at `http://trec.nist.gov/data/topics_eng/qa_questions_201-893`.

`is the moon?`. In these cases we cannot assume that a question category exists. However, our aim is to provide an additional answer filtering mechanism for a stand-alone Q/A systems. This means that when question categorization is not applicable, we can recognize this case and we can deactivate the filtering mechanism.

Next section describes the automatic question categorization model that exploits word statistics on category documents.

### Question categorization

In Chapter 2 and 3 we have shown that modern TC algorithms are quite effective, whereas in Section 4.1 we have shown that natural language sentences can be accurately categorized in FrameNet frames. Thus, our idea is to consider questions (as we did for the sentencesin Section 4.1.3) as a particular case of documents, in which the number of words is rather small.

The question categorization task is more difficult than sentence labeling as there are not available strong relevant features like phrase type and grammatical function. This poses two important problems:

(a) *Can the question categorization models converge given the small number of words per questions?*

(b) *How big has to be the number of training questions to ensure the classifier convergence?*

This latter question is very interesting for practical cases where the cost and the designing time for the target Q/A system strongly depend on the number of manually generated train questions. Note that, intuitively to ensure the convergence, the number of questions should be such that the training data includes a large portion of words that occur in feasible questions (this may be more than 10 thousands).

In order to overcome the above problems we dropped the idea to learn the question categorization function directly from a set of learning questions. We observe that, when the training of the target document categorization model is applied, an explicit set of relevant words together with their weights is defined for each category. Our idea is to exploit Rocchio and SVM learning on category documents to derive question categorization function.

We define for each question $q$ a vector $\vec{q} =< w_1^q, .., w_n^q >$ where $w_i^q \in \Re$ are the weights associated with the question features, i.e. the question words. Ideally, the weights for the question features can be computed using the same formulae 2.2 and 2.3 and substituting: $o_f^d$ with the $o_f^q$, the frequency of feature $f$ in question $q$, and $IDF(f)$ with the *Inverse Questions Frequency*, i.e., $IQF(f) = log\frac{M_q}{M_{qf}}$, where $M_q$ is the total number of questions and $M_{qf}$ is the number of questions that contain $f$. However, this is not practical for two reasons: (1) Each question has far less words than each document, and hence fewer features; and (2) generally the number of questions is also smaller than the number of documents. To address these two problems we have developed four different methods computing the weights of question features, which in turn determine five models of question categorization:

Method 1: If the $o_f^q$ is the frequency of feature $f$ inside the question $q$, then

$$w_f^q = \frac{l_f^q \times IDF(f)}{\sqrt{\sum_{r \in F}^{n} (l_r^q \times IDF(r))^2}} \tag{4.3}$$

where

$$l_f^q = \begin{cases} 0 & \text{if } o_f^q = 0 \\ log(o_f^q) + 1 & \text{otherwise} \end{cases} \tag{4.4}$$

and $F$ is the set of the training document features. This weighting mechanism uses the Inverse Document Frequency (IDF) of features instead of computing the Inverse Question Frequency. The rationale is that the number of questions is assumed smaller than the number of documents. When this method is applied to the Rocchio-based Text Categorization Model, by substituting $w_f^d$ with $w_f^q$ we obtain a model call the $RTC0$ model. When it is applied to the SVM model, by substituting $w_f^d$ with $w_f^q$, we call it SVM0.

Method 2: The weights of the question features are computed by formulae 4.3 and 4.4 employed in Method 1, but they are used in the Parameterized Rocchio Model. This entails that after questions are categorized on the training set of 120 questions, $\rho$ from formula 2.20 as well as the threshold $b$ are chosen to maximize the accuracy of categorization. We call this model of categorization PRTC.

Method 3: The weights of the question features are computed by formulae 4.3 and 4.4 employed in Method 1, but they are used in an extended $SVM$ model, in which two additional conditions enhance the optimization problem expressed by Eq. 2.19. The two new conditions are:

$$\begin{cases} Min \quad ||\vec{a}|| \\ \vec{a} \times \vec{q} + b \geq 1 \quad \forall q \in P_q \\ \vec{a} \times \vec{q} + b \leq -1 \quad \forall q \in \bar{P}_q \end{cases} \tag{4.5}$$

where $P_q$ and $\bar{P}_q$ are the set of positive and negative examples of training questions for the target category $C$. We call this question categorization model QSVM.

Method 4: We use the output of the basic Q/A system to assign a category to questions. Each question has associated up to five answer sentences. In turn, each of the answers is extracted from a document, which can be categorized. The categories of documents containing the answers determine the question category in the following way:

Case 1: The most popular category associated with the answers is propagated back to the question;

Case 2: If categories are equally popular (e.g., out of the $1 \leq k \leq 5$ answers, each has a different category), they are all propagated back to the question.

Case 3: If no answers are generated, the question is not assigned any category.

We named this ad-hoc question categorization method that relies on the results of Q/A and the categorization of the documents containing the answers QATC.

### 4.2.3 Answers filtering and Re-Ranking based on Text Categorization

Many Q/A system extract and rank answers successfully, without employing any TC information. For such systems, it is interesting to evaluate if TC information improves the ranking of answers they generate. In fact, the question category can be used in two ways: (1) to re-rank the answer by pushing down in the list any answer that is labeled with a different category than the question; or (2) to simply eliminate answers labeled with a category different than the question category.

First, the basic Q/A system has to be trained on documents that are categorized (automatically or manually) in a predefined categorization scheme. Then, the target questions as well as the answers provided by the basic $Q/A$ system are categorized. The answers receive the categorization directly from the categorization scheme, as they are extracted from categorized documents. The questions are categorized using one of the models described in the previous section. Two different impacts of question categorization on Q/A are possible:

- Answers that do not match at least one of the categories of the target questions are eliminated. In this case the precision of the system should increase if the question categorization models are enough accurate. The drawback is that some important answers could be lost because of categorization errors.

- Answers that do not match the target questions (as before) get lowered ranks. For example, if the first answer has categories different from the target question, it could shift to the last position in case of all other answers have (at least) one category in common with the question. In any case, all questions will be shown to the final users, preventing the lost of relevant answers.

More formally, the above two models are described by the following steps:

1. Given a basic Q/A system, train it with the target set of documents $D$ that are categorized in a collection $\mathcal{C} = \{C_1, .., C_n\}$.

2. Let $\phi$ the question categorization function implemented by one of the following models: $RTC0$, $SVM0$, $PRTC$, $QSVM$ and $QATC$. $\phi$ maps questions $q \in Q$ in a subset of $\mathcal{C}$, i.e., $\phi : Q \rightarrow 2^{\{C_1,..,C_n\}}$.

3. Let $A_q$ be the answer set that the basic Q/A returns for the question $q$, $d_a$ be the document that contain the answer $a \in A_q$ and $Cat(d)$ be the

set of categories of $d$. The output of the answer elimination model for the question $q$ is the answer sequence: $(a_1, a_2, .., a_k) : a_i \in TC_A$ where $(a_1, a_2, .., a_n), \quad n \geq k$ is the ranking provided by the basic Q/A system and

$$TC_A = \{a \in A_q : \exists C \in \mathcal{C}, C \in cat(d_a), C \in \phi(q)\}.$$

4. The answer re-ranking system takes into account the answer ordering by providing the sequence:
$(R(a_1), R(a_2), .., R(a_n))$ where $(a_1, a_2, .., a_n)$ is the answer ranking provided by the basic Q/A system and $R : A_q \to A_q$ is a bijection function such that $\forall i, j : i < j$, $R(a_i) > R(a_j)$ *iff*, $a_i \notin TC_A$, $a_j \in TC_A$.

Table 4.4: Example of question labeled in the *Crude* category and its five answers.

| Rank | Category | Question: *What did the Director General* say *about the energy floating production plants?* |
|---|---|---|
| 1 | *Cocoa* | " Leading cocoa producers are trying to protect their market from our product , " *said* a spokesman for Indonesia 's directorate general of plantations. |
| 2 | *Grain* | Hideo Maki , Director General of the ministry 's Economic Affairs Bureau , *quoted* Lyng as telling Agriculture Minister Mutsuki Kato that the removal of import restrictions would help Japan as well as the United States. |
| 3 | *Crude* | Director General of Mineral and Energy Affairs Louw Alberts announced the strike earlier but *said* it was uneconomic . |
| 4 | *Veg-oil* | Norbert Tanghe, head of division of the Commission's Directorate General for Agriculture, told the 8th Antwerp Oils and Fats Contact Days " the Commission firmly believes that the sacrifices which would be undergone by Community producers in the oils and fats sector... |
| 5 | *Nat-gas* | Youcef Yousfi, director - general of Sonatrach , the Algerian state petroleum agency , indicated in a television interview in Algiers that such imports. |

An example of the answer elimination and answer re-ranking is provided by the Table 4.4. As basic Q/A system we adopted the LCC-Q/A system[6]. TREC conference provides data-set for testing Q/A system, but unfortunately texts and questions are not categorized. Thus we trained the LCC-Q/A system with all *Reuters-21578* documents. Table 4.4 shows the five answers generated

[6] It is an advanced question answering system developed at Language Computer Corporation `www.languagecomputer.com`. LCC-Q/A won the TREC 2002 competition and other past TREC editions on question answering track.

for one example question and their corresponding rank. The categories of the text from which the answer was extracted is displayed in column 1. The question classification algorithm automatically assigned the *Crude* category to the question.

The processing of the question identifies the word *say* as indicating the semantic class of the expected answer and for paragraph retrieval it used the keywords $k_1$ = *Director*, $k_2$ = *General*, $k_3$ = *energy*, $k_4$ = *floating*, $k_5$ = *production* and $k_6$ = *plants* as well as all morphological variations for the nouns. For each answer from Table 4.4, we have underlined the words matched against the keywords and emphasized the word matched in the class of the expected answer, whenever such a word was recognized (e.g., for answers 1 and 2 only). For example, the first answer was extracted because words producers, product and directorate general could be matched against the keywords production, Director and General from the question and moreover, the word *said* has the same semantic class as the word *say*, which indicates the semantic class of the expected answer.

The ambiguity of the word plants cause the basic Q/A system to rank answer related to *Cocoa* and *Grain* plantations higher than the correct answer, which is ranked as the third one. If the answer re-ranking or elimination methods are adopted, the correct answer reaches the top as it was assigned the same category as the question, namely the *Crude* category.

This example shows that question categorization captures extra important information that the weighting schemes and the heuristics of the basic Q/A system do not detect. The information added seems related to the relation among specific concepts contained in the question. *Cocoa* and *plantations* relation in the answer 1 is difficult to be detected as (a) the words are too much *distant* so they need a discourse interpreter to be related and (b) world knowledge is needed to derive that *Cocoa* can be a kind of plantation. On the contrary, the categorization function establishes that the question is related to *energy plant* whereas the category of the answer suggests that the stem *plant* (from plantation) refers to vegetable. This is enough to detect that the *plant* sense in the answer is different than the sense assumed by *plant* in the question. Text Categorization seems to provide an effective WSD.

Next section describes in detail our experiments to prove that TC add some important information for selecting relevant answers.

### 4.2.4 Experiments

The aim of the experiments is to prove that category information used as described in previous section is useful for Q/A system. For this purpose we have to show that the performance of a basic Q/A system is improved when the question filtering is adopted. To implement our Q/A and filtering system we need:

- A state of the art Q/A system. Low accurate systems may produce many wrong answers probably due to their weak weighting schemes. If we measure the improvements on these Q/A systems we cannot assess that we have added relevant information; probably we have added just a more accurate way to use the standard information. As previously stated we decided to use the Q/A LCC system that is the current *state-of-the-art*.

- A collection of categorized documents on which training our basic Q/A system. We cannot use the TREC corpora because they are not categorized. We decided to use the *Reuters-21578* corpus because it is very common in TC experiments and it contains many categories. This last property is crucial as the more specific is the application domain the more specific is the categorization of the questions. High specificity produces a high level of filtering. In contrast, a low granular categorization schemes (i.e. few categories) does not capture the differences among questions. These latter would result too much general.

- A set of questions categorized according to the Reuters categories. A portion of this set is used for learning PRTC and QSVM models, the other disjoint portion is used to measure the performance of the Q/A systems.

Next section, instead describes the technique used to produce the question corpus.

### Question generations

The idea of PRTC and QSVM models is to exploit a set of questions to improve the learning of the PRC and SVM text classifiers. This means that for each category of the Reuters corpus we need to have a set of questions that are categorized in it. If we choose to produce only 20 questions for each category, the total number of questions for 90 categories is $20 \times 90 \sim 2000$, thus, we decided to test our algorithms on 5 top-populated categories only. We chose *Acq*, *Earn*, *Crude*, *Grain*, *Trade* and *Ship* categories. To generate questions related to the above categories, we randomly selected a number of documents from each category. Then we tried to formulate questions related to the target documents. Three cases were found:

(a) The document does not contain feasible questions. We tried to formulate general questions. In contrast, many documents contain specific information that can be found in just one documents. Thus, some selected documents did not offer the possibility to create general questions.

(b) The document suggests general questions, in this case some of the words that are contained in the answer (of that document) are replaced with synonyms. This makes difficult the retrieval of the document from which the question was generated.

(b) A document $d$ categorized in the category $C$ suggests general questions. These latter are typical of categories different from $C$. We add these questions in our data-set associated with their true categories.

Table 4.5 lists a sample of the questions we derived from the target set of categories. It is worth noting that we include short queries also to maintain general our experimental set-up.

Table 4.5: Some training/testing Questions

| | |
|---|---|
| *Acq* | Which strategy aimed activities on core businesses?<br>How could the transpacific telephone cable between the U.S. and Japan contribute to forming a join venture? |
| *Earn* | What was the most significant factor for the lack of the distribution of assets?<br>What do analysts think about public companies? |
| *Crude* | What is Kuwait known for?<br>What supply does Venezuela give to another oil producer? |
| *Grain* | Why do certain exporters fear that China may renounce its contract?<br>Why did men in port's grain sector stop work? |
| *Trade* | How did the trade surplus and the reserves weaken Taiwan's position?<br>What are Spain's plans for reaching European Community export level? |
| *Ship* | When did the strikes start in the ship sector?<br>Who attacked the Saudi Arabian supertanker in the United Arab Emirates sea? |

We generated 120 questions and we used 60 for the learning and the other 60 for testing. To measure the impact that TC has on Q/A, we first evaluated the question categorization models presented in Section 2.5. Then we compared the performance of the basic Q/A system with the extend Q/A that adopts the answer elimination and re-ranking methods.

### Performance Measurements

The question categorization algorithms are evaluated by using the $f_1$ measure. This latter has been evaluated as it is done for the document categorization by considering questions as small documents.

The Q/A performance is computed by the reciprocal value of the rank (RAR) of the highest-ranked correct answer generated by the Q/A system. Given that only the first five answers for the question $i$ were considered, RAR is defined as $1/rank_i$, its value is 1 if the first answer is correct, 0.5 if the second answer is correct but not the first one, 0.33 when the correct answer was on the third position, 0.25 if the fourth answer was correct, and 0.1 when the fifth answer was correct. If none of the answers are corrects, RAR=0. The Mean Reciprocal

Answer Rank (MRAR) is used to compute the overall performance of Q/A[7], defined as $MRAR = \frac{1}{n}\sum_i \frac{1}{rank_i}$, where $n$ is the number of questions.

Since we believe that TC information is meaningful for preferring out incorrect answers, we defined a second measure for evaluating Q/A. For this purpose we replaced the MRAR measure with a signed reciprocal (SRAR), which is defined as $\frac{1}{n}\sum_{j \in A} \frac{1}{srank_j}$, where $A$ is the set of answers given for a set of questions, $|srank_j|$ is the rank position of the answer $j$ and $srank_j$ is positive if $j$ is correct and negative if it is not correct. The Mean Signed Reciprocal Answer Rank can be evaluated over a set of questions as well as over only one question. SRAR for a single question is 0 only if none answer was provided for it.

For example, given the answer ranking of Table 4.4 and considering that we have just one question for testing, the MRAR score is 0.33 while the SRAR is -1 -.5 +.33 -.25 -.1 = -1.52. If the answer re-ranking is adopted the MRAR improve to 1 and the SRAR becomes +1 -.5 -.33 -.25 -.1 = -.18. The answer elimination produces a MRAR and a SRAR of 1.

### Evaluation of Question Categorization

Table 4.6 lists the performance of question categorization for each of the models described in Section 2.5. We noticed better results when the PRTC and QSVM models were used. In the overall, we find that the performance of question categorization is not as good as the one obtained for TC (see Section 2.7.3).

Table 4.6: $f_1$ performances of question categorization.

| | RTC0 | SVM0 | PRTC | QSVM | QATC |
|---|---|---|---|---|---|
| acq | 18.19 | 54.02 | 62.50 | 56.00 | 46.15 |
| crude | 33.33 | 54.05 | 53.33 | 66.67 | 66.67 |
| earn | 0.00 | 55.32 | 40.00 | 13.00 | 26.67 |
| grain | 50.00 | 52.17 | 75.00 | 66.67 | 50.00 |
| ship | 80.00 | 47.06 | 75.00 | 90.00 | 85.71 |
| trade | 40.00 | 57.13 | 66.67 | 58.34 | 45.45 |

### Evaluation of Question Answering

To evaluate the impact of TC on Q/A we first scored the answers of a basic Q/A system for the test set, by using both MRAR and the SRAR measures.

Additionally, we evaluated (1) the MRAR when answers were re-ranked based on question and answer category information; and (2) the SRAR in the case when answers extracted from documents with different categories were eliminated. Table 4.8 shows that matching between the question category and the answer category improves both the MRAR (.6635 vs .6619) and the SRAR (-.0356 vs -.3724) score.

[7] The same measure was used in all TREC Q/A evaluations.

Table 4.7: Performance comparison between basic Q/A and Q/A using question categories information for answer extraction

| Quest. Categ. method | RTC0 | SVM0 | PRTC | QSVM | QATC |
|---|---|---|---|---|---|
| MRAR (QCQA) | .6203 | .6336 | .6442 | .6507 | .5933 |
| SRAR (QCQA) | -.4091 | -.3912 | -.3818 | -.3954 | -.4753 |
| MRAR (basic Q/A) | .6619 | | | | |
| SRAR (basic Q/A) | -.3724 | | | | |

Table 4.8: Performance comparison between the answer re-ranking and the answer elimination policies.

| Quest. Categ. method | RTC0 | SVM0 | PRTC | QSVM | QATC |
|---|---|---|---|---|---|
| MRAR (answer re-ranking) | .6224 | .6490 | .6577 | .6635 | .6070 |
| SRAR (answer elimination) | -.0894 | -.1349 | -.0356 | -.0766 | -.3199 |

In order to study how the number of answers impacts the accuracy of the proposed models, we have evaluated the MRAR and the SRAR score varying the maximal number of answers, provided by the basic Q/A system. We adopted as filtering policy the answer re-ranking.

Figure 4.5 shows that as the number of answers increases the MRAR score for QSVM, PRTC and the basic Q/A increases, for the first four answers and it reaches a plateau afterwards. We also notice that the QSVM outperforms both PRTC and the basic Q/A. This figure also shows that question categorization per se does not greatly impact the MRAR score of Q/A.

Figure 4.6 illustrates the SRAR curves by considering the answer elimination policy. The figure clearly shows that the QSVM and PRTC models for question categorization determine a higher SRAR score, thus indicating that fewer irrelevant answers are left. The results presented in Figure 4.6 show that question categorization can greatly improve the quality of Q/A when irrelevant answers are considered. It also shows that perhaps, when evaluating Q/A systems with MRAR scoring method, the "optimistic" view of Q/A is taken, in which erroneous results are ignored for the sake of emphasizing that an answer was obtained after all, even if it was ranked below several incorrect answers.

In contrast, the SRAR score that we have described in Section 4.2.4 produce a "harsher" score, in which errors are given the same weight as the correct results, but are affecting negatively the overall score. This explains why, even for a baseline Q/A, we obtained a negative score, as illustrated in 4.7. This shows that the Q/A system generates more erroneous answers then correct answers.

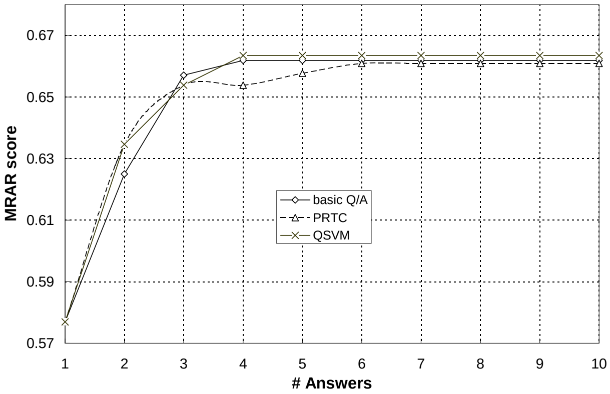


Figure 4.5: The $MRAR$ results for basic Q/A and Q/A with answer re-ranking based on question categorization via the PRTC and QSVM models.

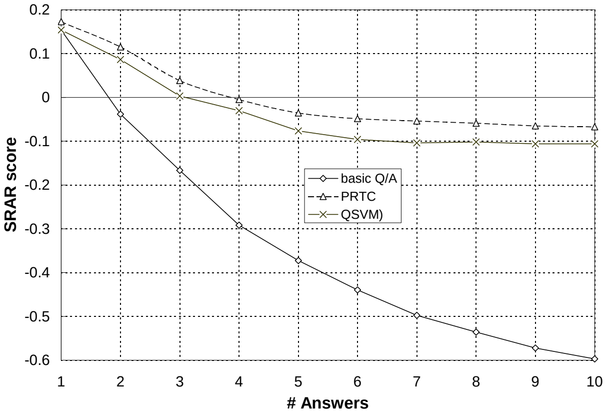


Figure 4.6: The $SRAR$ results for basic Q/A and Q/A with answer re-ranking based on question categorization via the PRTC and QSVM models.

This contrast between the MRAR scoring method and the SRAR scoring method is obvious in the results listed in Table 4.8. The five different text categorization methods generate MRAR scores that are quite similar. However, their SRAR scores vary more significantly.

If only the MRAR scores would be considered, two conclusions can be drawn:

1. text categorization does not bring significant information to Q/A for precision enhancement by re-ranking answers;

2. question categorization by using weighting scheme of text categorization does not perform correctly enough to be used for Q/A.

However, the results obtained with the SRAR scoring scheme, indicate that text categorization impacts on Q/A results, by eliminating incorrect answers. We plan to further study the question categorization methods and empirically find which weighting scheme is ideal.

In the next section a different use of Text Categorization is shown. Indicative and Informative summaries will be derived using categorical information.

## 4.3 Category-Based Text Summarization

One of use of TC is the automatic delivery of textual information to interested users based on the notion of text categories: first, news providers tag news items according to a predefined classification scheme and, second, they deliver the news to the interested users. On one hand, the more fine-grained is the classification structure the more specific information can be provided to the users. On the other hand, the more fine-grained is the category structure the less accurate is the system. Moreover, providers and consumers may have a different understanding of a huge classification scheme.

A common solution for this problem is the use of keywords or small summaries that gives and indication of which topics the target document is related to. The above information can be manually added to documents but this results in a high time consuming and costly activity. Automated method to generate keywords and summaries exploit traditional weighing scheme from $IR$. The relevant keywords can be considered as indicative summaries whereas relevant passages of a document or set of documents refer to as informative summary. Usually, the summaries are extracted based on queries, i.e. they are relevant passages and terms for the target query.

We introduce the concept of relevance with respect to a category. The indicative and informative summaries are extracted based on weighting schemes derived from the training data of the target category. In Chapter 3 has been shown that the *bag-of-words* representation is sufficient to achieve good performances. However, when an indicative explanation of document content is given in term simple words, it could not be sufficient to satisfy the users' information needs. On the contrary if the output keywords are terminological expressions or other complex nominals, their understandability improve. NLP cannot increase the classification accuracy but can improve the descriptive (at least for human point of view) quality of keywords.

A richer explanation can, also, help to recover misclassifications of the automatic categorization system. The user can better decide to thrust the system and read the news item, or, conversely, discard it. This may not be possible if he is exposed only to the document category and to the title of the current actual news. For instance, given the title:
`"Periventricular hyperintensity detected by magnetic resonance imaging in infancy."`,
it is not clear why the document of the medical domain in Tab. 4.9 is related to the *Nervous System Diseases* category of the Medical Subject Headings (MeSH[8]).

If the user is provided also with an indicative summary represented by the complex nominals such as *intracranial hemorrhage*, *cerebral palsy*, *brain damage* and *cerebral injuries*, he may better understand if this incoming document is related to the above class. This perception may be improved if an informative summary is presented. This latter is built using the sentences that contain the

[8] A complete description can be found in `http://www.nlm.nih.gov/mesh`

| **Title**: Periventricular hyperintensity detected by magnetic resonance imaging in infancy. |
|---|
| **Abstract**:<br>Twenty-one infants younger than 12 months of age were diagnosed as having *periventricular hyperintensity* (PVH) on T2-weighted *magnetic resonance* imaging.<br>Ten infants had experienced *neonatal asphyxia*, 6 intracranial hemorrhage, 2 bacterial meningitis, and 3 apnea.<br>PVH was classified according to its extent. Round foci of PVH surrounding the frontal and occipital horns of the *lateral ventricles* were observed in 4 infants (PVH pattern I). Continuous PVH was observed in 17 infants (PVH patterns II and III). Fourteen infants with continuous PVH had spastic diplegia or quadriplegia. Developmental delay was demonstrated in 15 infants with continuous PVH. No PVH pattern I infants had cerebral palsy; only 1 such infant had mild developmental delay. Our study suggests that the extent of PVH reflects the severity of brain damage in neonates with cerebral injuries. |

Table 4.9: Ohsumed sample news item

above concepts. Note that, to suggest the correct subject, the indicative and the informative summaries have to be related to the actual category. For example the complex nominals: *neonatal asphyxia*, *lateral ventricles* and *magnetic resonance* are useless or even misleading. The complex nominal and the simple nouns filtered by the profile weighting schemes are a kind of category-based explanation for the document content.

### 4.3.1 Representing documents for enriched categorization

In Chapter 2, we have shown that text classifiers based on a Vector Space Model represent documents as points in the space. The profile vector $\vec{a}$ produced by Rocchio or by $SVM$ learning algorithm contains the target category features ranked by their relevance for classifying documents in the target category. We speculate that if a feature $f$ is very relevant to correctly categorize documents in the category $C$, $f$ should be indicative also for a human being.

The expressiveness power of features can be improved if we use together with the simple words the complex nominal representations introduced in Section 3.1.3. In fact, we have shown in Chapter 3 that the indexing effectiveness of complex nominals is not lesser than the simple words. The problem to use them for TC is that they are subsumed by their compounding words. Anyhow, they are more meaningful for a human being than the bunch of words, apparently not related, that the classifier uses for categorization.

In Table 4.10 we show terminological features in the profile of the *Neonatal Diseases & Abnormalities* category of Ohsumed. Features are ranked according to the weight $\vec{a}_f$ produced by the $PRC$ model. In the Table the head and the tail of the list are shown in left and right columns, respectively.

Table 4.10: Complex Nominals extracted from the *Neonatal Dis.& Abnormalities* category texts and ranked according to the $PRC$ model (*only non null weights are reported*)

| Head List | Weight | Tail list | Weight |
|---|---|---|---|
| cystic_fibrosis | 0.017391 | ... | |
| pulmonary_artery | 0.005903 | shear_stress | 0.000074 |
| congenital_heart_disease | 0.005181 | 28_days | 0.000074 |
| birth_weight | 0.003942 | three_time | 0.000070 |
| premature_infant | 0.003646 | twin_transfusion | 0.000066 |
| congenital_anomalies | 0.003396 | significant_advantage | 0.000060 |
| intrauterine_growth_retardation | 0.003175 | lower_incidence | 0.000058 |
| fetal_growth | 0.003067 | data_collection | 0.000054 |
| cystic_fibrosis_gene | 0.002897 | lung_damage | 0.000052 |
| congenital_abnormalities | 0.002711 | structural_abnormalities | 0.000046 |
| outflow_tract | 0.002527 | imaging_technique | 0.000045 |
| double_inlet | 0.002335 | dose_group | 0.000045 |
| congenital_heart_defects | 0.002274 | 3_6 | 0.000045 |
| congenital_anomaly | 0.001890 | late_deaths | 0.000044 |
| early_pregnancy | 0.001888 | treatment_strategy | 0.000039 |
| full_term | 0.001258 | specific_binding | 0.000039 |
| 23_weeks | 0.001250 | early_age | 0.000035 |
| color_flow_mapping | 0.001180 | skin_cancer | 0.000034 |
| low_cardiac_output | 0.001115 | social_class | 0.000023 |
| pulmonary_artery_distortion | 0.001060 | 45_cases | 0.000016 |
| low_birth_weight_infants | 0.001027 | binding_proteins | 0.000014 |
| diabetic_women | 0.000991 | live_birth | 0.000012 |
| arch_obstruction | 0.000868 | bladder_wall | 0.000009 |
| ... | | young_woman | 0.000000 |

As comparison, Table 4.11 the words that compound the complex nominals of the Table 4.10. They have been alphabetically ordered to make more difficult the recognition of the complex nominals. We notice that the bunch of words is less meaningful. For example the number 23 or the word *weeks* have no sense if they are taken alone. Instead, the complex nominal *23_weeks* evokes a recurrent period of time of pregnancy. Other examples are *congenital_heart_disease*, *low_birth_weight_infants* and *intrauterine_growth_retardation*. Their compound words alone are not very meaningful.

Moreover, note that in Table 4.10 concepts relevant for the *Neonatal* class (e.g., *congenital_anomaly*, *premature_infant*) appear higher in the ranking (Head list), while less topics oriented multiwords (e.g., *social_class*) receive a very low

Table 4.11: Single words extracted from the complex nominals of Table 4.10. They have been alphabetically ordered to better separate the compounding words

| Single Words | | | |
|---|---|---|---|
| 23 | cystic | gene | output |
| abnormalities | defects | growth | pregnancy |
| anomalies | diabetic | heart | premature |
| anomaly | disease | infant | pulmonary |
| arch | distortion | infants | retardation |
| artery | double | inlet | term |
| birth | early | intrauterine | tract |
| cardiac | fetal | low | weeks |
| color | fibrosis | mapping | weight |
| congenital | flow | obstruction | women |
| | full | outflow | |

(although not null) weight. This shows that NLP derived features filtered by the TC algorithm result more meaningful for a human being. In the next sections, we presents another NLP technique that allows us to detect more general of complex nominals than those extracted by using the methods of Section 3.1.3.

**Extending the word-based document representation**

The VSM based on simple words lacks in expressiveness. In fact, words, considered independent, provide only singleton surface forms. These latter are only a small part of the key concepts expressed in the documents and, moreover, are generally polysemic, i.e. denote more than one concept. The consequence is a very poor representation from the user point of view.

A large part of relevant concepts in domains is expressed by collocations of more than one word (e.g., *interim dividend* in the financial domain). Collocations have also the positive property of denoting generally only one concept. This is also true for terminology expressions [Jacquemin, 2001]. Phrases like the *risk factor* or *interim dividend* that match both the *Noun Noun* and `Adjective Noun` constraints are less polysemic than the isolated compounding words, i.e. *risk*, *interim*, *factor*, and *dividend*.

Other important phrase are expressed by verb-governed surface forms such as *companies buy shares*. This information may be useful ”as it is” for the description of the class. Since the verb arguments may be very distant and in relatively free order, a normalized version may be used in the vector space model, to increase the number of matches.

The document representation that we want to produce is, thus, based on:

- concepts expressed with simple surface forms, i.e. words;
- concepts expressed with complex surface forms, i.e. complex terms;

- simple relations between concepts based on verbal contexts;

To support the discovery of such explicit descriptions some NLP tools have to be defined. Simple techniques based on barrier words are not sufficient. These approaches show their limits if applied to long distance dependencies such as the verb argumental relation.

### Description of the Extraction Algorithm

In the terminology extraction techniques [Jacquemin, 2001], a syntactic model of the textual phenomena is generally used. We have adopted the extended dependency-based representation formalism (XDG, [Basili *et al.*, 2000d]).

An XDG is a graph whose nodes are constituents and whose arcs are the syntactic relations among constituents. The constituents that we consider are *chunks* [Abney, 1996], i.e., non-recursive kernels of noun phrases ($NPK$), prepositional phrases ($PPK$) and verbal phrases ($VPK$) like *five patients*, *by non invasive methods*, *were evaluated*. Arcs indicate the syntactic relations between chunks, i.e. the inter-chunks relations such as *verb-subject*, *verb-object*, *verb-modifier*, and *noun-prepositional modifiers*.

Fig. 4.7 shows a sample XDG: chunks[9] are the words between square brackets (i.e. VPK, NPK and PPK) while inter-chunk dependencies are depicted as arrows, i.e.:

- SUBJ for the subject relation,
- V_PP for the verb prepositional modifier relation, and
- NP_PP for the noun-prepositional modifier relation.

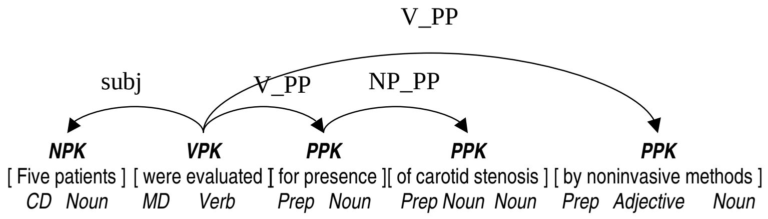

V_PP
subj
V_PP
NP_PP
NPK
VPK
PPK
PPK
PPK
[ Five patients ] [ were evaluated ] [ for presence ][ of carotid stenosis ] [ by noninvasive methods ]
CD Noun MD Verb Prep Noun Prep Noun Noun Prep Adjective Noun

Figure 4.7: Example of an XDG

The surface pattern candidates for the complex phrases can be detected by regular expressions like $\{NPK \quad PPK^*\}$ or $\{PPK^+\}$ on the XDG node sequence. A node sequence $N_1, .., N_k$ that satisfies one of the regular expressions is accepted if $\forall i : 1 \leq i < k$, the pair $<N_i, N_{i+1}>$ is an edge of the target XDG. It is worth noticing that we do not consider all the $PPK$, e.g., $PPK$s that contain pronouns are refused.

[9] The chunk layer is build on a part-of-speech tagged text.

The relations among concepts, instead, are extracted by verb-dependencies. Verb argument pairs are relevant for describing the target class. For example, *(buy,(dirobj,'share'))* or *(complete,(dirobj,'acquisition'))* in an economic corpus suggest that the text collection refers to the changes of company assets. The same information was used in [Strzalkowski *et al.*, 1998] to enrich the document representation for $IR$ tasks as described in Section 1.2.

The adoption of robust syntactic parsing techniques based on processing module cascades [Basili *et al.*, 2000d] makes possible the selection of the above surface forms on a large scale. The parser includes a tokenizer, a part-of-speech tagger, a chunker, and a shallow syntactic analyzer.

### 4.3.2 Explanation of categorization choices

Our aim is to provide two type of summaries as explanation of the target document categorization: one *indicative* and one *informative*. These summaries should show important concepts shared by both the document and the target category. For this purpose, we rank the features $f$ according the scores $sc^d_f = w^d_f \times \vec{a}_f$, i.e. the product between the document and the profile weights of $f$.

To generate the $\vec{a}$ we chose the $PRC$ since the feature selection interpretation in Section 2.6 has shown that:

- $PRC$ drastically reduces noise filtering out non-relevant features.
- The $\vec{a}_f$ weights depend on the target category and are directly used as components in the similarity estimation with the document (i.e., the scalar product).
- Simple words as well as complex linguistic features receive a weight proportional to their contribution in the classification accuracy. Note that as a parameter $\rho$ is provided for each category, features assume different weights in different categories. This defines the best suitable set of concepts (i.e. the features with higher weights) for the target category.

The indicative summary of the document $d$ is defined as the $R_k(d)$ set of the $k$ top features (*k-best features*) ranked by $sc^d_f$. The document features contain both complex terms and simple words thus the summary should be more descriptive than those based on words only. We call such an explanation as the *summary based on best features* ($S_{bf}$).

The informative summary should contain the more meaningful paragraphs (*m-best paragraphs*). The paragraphs that contain at least one of the best $k$ features are ranked according to $w^d_p$ weight defined in the following. Given a paragraph $p$ in a document $d$, the set of the best $k$ paragraph features are:

$$S_k(p, d) = \{f : f \in p, f \in R_k(d)\},$$

where $f$ is a feature in $p$. The paragraph weight is then defined as follows:

$$w^d_p = \sum_{f \in S_k(p,d)} sc^d_f = \sum_{f \in S_k(p,d)} w^d_f \times \vec{a}_f, \qquad (4.6)$$

where $p$ is a paragraph of $d$.

The informative summary based on the *best paragraphs* ($S_{bp}$) is obtained by picking-up the top $m$ paragraphs ranked according to Eq. 4.6. The parameter $m$ establishes the rate of the document paragraphs shown as an explanation.

A base-line version of the proposed explanation model can be obtained by replacing the $w_f^d \times \vec{a}_f$ score with the simpler *document frequency*, $M_f^c$ (i.e. the number of documents that contain $f$ and belong to the category $C$). Hereafter we will refer to these simpler explanation models as the *frequency summary based on features* ($S_{ff}$) and *frequency summary based on paragraphs* ($S_{fp}$).

In next section the above explanation models are contrastively evaluated.

### 4.3.3 Experiments

For evaluating the performance of our category-based summaries we adopted the Ohsumed corpus. In the first experiment we used the extended representation described in Section 4.3.1 to train $PRC$. Column 1 of Table 4.12 shows the top 31 complex terms of *Cardiovascular Disease* category profile generated by $PRC$. The features seem to be conceptually close to the target domain. Column 2 shows the complex terms ordered by frequency inside the category. We observe that some non-relevant features as well as non specific terms, i.e., *normal subject*, *control subject*, *risk factor*, *side effect*, *appo patients* and so on have reached the top of ranking positions. As suggested in [Daille, 1994], frequency seems to be a good indicator of domain relevance, however cross-class techniques, as the one proposed, eliminates the unspecific and useless terms.

$PRC$ seems, thus, suitable to select important domain features. Next section shows our summarization models based on $PRC$ as well as preliminary experiments to test their effectiveness for the users.

#### Evaluation of different summaries

The aim of these experiments is to measure the effectiveness of our explanation methods. This objective can be achieved in several ways. As our purpose is to design a document filtering system based on users' information needs we have implemented a specific experimental procedure to test the user satisfaction.

A randomly generated set of about 200 documents ($UTS$) has been selected from the classified *test-set.* The user has to evaluate if an incoming document $d$ is correctly labeled in the category $C$, i.e. if $d$ belongs to $C$ according to his own perception of the classification scheme. Documents are presented to the users together to a category $C$ that may or may not be the true category of the document according to the classification scheme (50% are correct). The user is asked to state its *acceptance*, or its *rejection* with respect to the shown class $C$. For each document $d \in UTS$, the user goes through 3 steps that make available different kinds of information:

1. The *Indicative summary*, made of the document title and the $S_{bf}$ (set of best features) or $S_{ff}$ (set of frequent features) defined in Section 4.3.2.

| *PRC* | *Frequency* |
|---|---|
| myocardial infarction | myocardial infarction |
| coronary angioplasty | coronary artery |
| coronary artery | risk factor |
| essential hypertension | coronary angioplasty |
| acute myocardial infarction | congestive heart failure |
| congestive heart failure | acute myocardial infarction |
| myocardial ischemia | pulmonary hypertension |
| hypertensive patients | essential hypertension |
| ventricular function | myocardial ischemia |
| arterial pressure | ventricular tachycardia |
| ventricular tachycardia | arterial pressure |
| pulmonary hypertension | hypertensive rat |
| hypertensive rat | ventricular function |
| cardiovascular disease | hypertensive patients |
| coronary angiography | vascular resistance |
| cardiac catheterization | cardiac arrest |
| atrial fibrillation | atrial fibrillation |
| cardiac arrest | appo patients |
| cardiac output | cardiac output |
| thrombolytic therapy | control subject |
| mitral regurgitation | significant difference |
| hypertrophic cardiomyopathy | consecutive patients |
| vascular resistance | chest pain |
| angina pectoris | cardiac catheterization |
| antihypertensive agent | hypertrophic cardiomyopathy |
| doppler echocardiography | side effect |
| unstable angina | pulmonary artery |
| enzyme inhibitors | cardiovascular disease |
| atrial pressure | cardiac death |
| coronary disease | thrombolytic therapy |
| mitral stenosis | normal subject |

Table 4.12: Complex term Ohsumed Cardiovascular disease class descriptor: *PRC* vs. simple frequency

Table 4.13 shows that the keywords are ranked by relevance and that a weight is also provided for the user decision.

2. The *Informative summary*, including the $S_{bp}$ (set of the *best* paragraphs) or $S_{fp}$ (set of the *frequent* paragraphs) is shown as described by the Table 4.14.

3. The *Full document* where the entire document is shown for the final decision (see Table 4.15).

The example of Table 4.13 shows that the features chosen by $S_{bf}$ model appears to be very related to the *Nervous System Diseases* category. The complex nominals make more meaningful the indicative summary, e.g., *cerebral palsy* is more understandable than single term *palsy*. It is worth noting that our com-

```
                             Title

Periventricular hyperintensity detected by magnetic resonance
imaging in infancy.
                            Keywords

        brain                 (weight: 0.024370685722)
        meningitis            (weight: 0.011201831273)
        intracranial          (weight: 0.010946837855)
        cerebral_palsy        (weight: 0.010880907534)
        magnetic              (weight: 0.010098027662)
        frontal               (weight: 0.009724019459)
        resonance             (weight: 0.008938488026)
        bacterial_meningitis  (weight: 0.006495032071)
        periventricular       (weight: 0.005859534725)
        spastic               (weight: 0.004452016709)
        lateral               (weight: 0.003823484388)
        quadriplegia          (weight: 0.003561293779)
        diplegia              (weight: 0.002432972968)
        hyperintensity        (weight: 0.002430102542)
        lateral_ventricles    (weight: 0.002124774555)
        cerebral_injury       (weight: 0.002051837893)

   Do you agree that this document is related to the
        'Nervous System Diseases' Category?

0) yes, 1) weakly, 2) probably not, 3) not at all
```

Table 4.13: Phase 1 of user understandability testing. Only the title and the relevant keywords are provided to decide if the document is relevant or not for the target category.

```
                             Title

Periventricular hyperintensity detected by magnetic resonance
imaging in infancy.
                            Summary

No PVH pattern I infants had cerebral palsy; only 1 such infant
had mild developmental delay.
Our study suggests that the extent of PVH reflects the severity
of brain damage in neonates with cerebral injuries.
Ten infants had experienced neonatal asphyxia, 6 intracranial
hemorrhage, 2 bacterial meningitis, and 3 apnea.

   Do you agree that this document is related to the
        'Nervous System Diseases' Category?

0) yes, 1) weakly, 2) probably not, 3) not at all
```

Table 4.14: Phase 2 of user understandability testing. The title and the summary is shown to the user to decide if the document is relevant or not for the target category.

```
                           Title

Periventricular hyperintensity detected by magnetic resonance
imaging in infancy.
                          Abstract

Twenty-one infants younger than 12 months of age were diagnosed
as having periventricular hyperintensity (PVH) on T2-weighted
magnetic resonance imaging.  Ten infants had experienced neonatal
asphyxia, 6 intracranial hemorrhage, 2 bacterial meningitis, and
3 apnea.
PVH was classified according to its extent.  Round foci of
PVH surrounding the frontal and occipital horns of the lateral
ventricles were observed in 4 infants (PVH pattern I). Continuous
PVH was observed in 17 infants (PVH patterns II and III).
Fourteen infants with continuous PVH had spastic diplegia or
quadriplegia.
Developmental delay was demonstrated in 15 infants with
continuous PVH. No PVH pattern I infants had cerebral palsy; only
1 such infant had mild developmental delay.  Our study suggests
that the extent of PVH reflects the severity of brain damage in
neonates with cerebral injuries.

   Do you agree that this document is related to the
        'Nervous System Diseases' Category?

0) yes, 1) weakly, 2) probably not, 3) not at all
```

Table 4.15: Phase 3 of user understandability testing. The entire document is shown to the user that can finally give his perception of the document category

plex nominal extractor does not cover every phenomena yet. For instance the term *spastic diplegia or quadriplegia* is not recognized. The lack of *diplegia* and *quadriplegia* words in our lexicon increased the error probability of the POS-tagger and consequently of the terminology extractor. In any case as the single words alone were judged important for the domain and displayed to the user.

We note that the example do not show any verbal phrases as indicative keywords. These were also rare in the indicative summaries of other documents/categories. The explanations could be: (a) the linguistic content of Ohsumed documents, i.e., there are few meaningful verbal phrases, and (b) to the higher complexity of clustering verbal phrases, especially when they are not frequents.

It is worth noticing that the $S_{bp}$ is displayed after the user has been exposed to the $S_{bf}$ while $S_{fp}$ is shown after $S_{ff}$. This means that it was not possible to measure the $S_{bp}$ and $S_{fp}$ independently from the related indicative summaries.

When a wrong category is proposed (with respect to the test information in Ohsumed), the system always provides its best explanation. The user has thus no information about the correctness of the proposed class, so that he relies only on explanations.

The first user group (1,2,3, and 4) tested the Rocchio-based explanation models (i.e. $S_{bf}$ and $S_{bp}$); the other users tested the $k$-frequent explanation models (i.e. $S_{ff}$ and $S_{fp}$). We define the explanation *score* as the user coherence with its own final decision. This can be measured for both phase 1 (indicative summaries) and phase 2 (informative summaries). We define the user coherence as:

> *the number of matches between the decisions taken for the target phase and the last phase, when the entire document is displayed.*

It is worth noticing that that we collect for each document 4 types of answers: *yes*, *weakly*, *probably not*, *not at all.* In these preliminary experiments we group together the first two as affirmative and the last two as negative answers. Other more refined way of evaluating the user perception can be further implemented by using grading matches rather than binary ones.

In Table 4.16 the performances of 7 users are reported. Users have been divided in two groups. In columns *Ind. Summary* and *Inf. Summary*, the scores of the explanation models based on indicative and informative summaries are respectively reported. In *Classifier* column is reported the user satisfaction with respect to the category assigned by the classifier. In the *avg.* rows, the average of the corresponding user group is shown.

Two main trends can be observed. First, the category assigned by the classifier seems to be the least satisfying, i.e. its agreement score (with the final user opinion) is the lowest. If the $S_{bf}$ are added for explaining the category label the average score increases of about 13% (79.13% vs. 66.08%). As the explanation model becomes richer, i.e. the $S_{bp}$ are also provided, the users better appreciate the final document content. This reflects in a further increase of about 8% with respect to the feature based model. The overall improvement of

Table 4.16: Evaluation of the class explanation model

| User | Classifier | Ind. Summary | Inf. Summary |
|---|---|---|---|
| 1 | 0.7450 | 0.8431 | 0.9019 |
| 2 | 0.5890 | 0.7945 | 0.8493 |
| 3 | 0.6557 | 0.7950 | 0.8852 |
| 4 | 0.6534 | 0.7326 | 0.8712 |
| *avg.* | 0.6608 | 0.7913 | 0.8769 |
| 5 | 0.7647 | 0.8921 | 0.9705 |
| 6 | 0.5791 | 0.6582 | 0.7861 |
| 7 | 0.7523 | 0.8012 | 0.8802 |
| *avg.* | 0.6987 | 0.7838 | 0.8789 |

the user satisfaction of the combined explanation model is around 21% (87.69% vs. 66.08%).

The second aspect is that even the explanation models based on the simple frequency are helpful. In this case, the $S_{ff}$ and $S_{fp}$ improve the baseline of about, respectively, 9% and 18%. As expected, adding explanatory information about the document category is always effective. However, the $PRC$ approach to feature selection seems more promising as it better improve (+21%) the baseline explanation (i.e. category and title only) than the document frequency heuristic (+18%).

A further advantage of the $S_{bf}$ over $S_{ff}$ is that the first actually selects and presents to the user only 4 features, on average, with respect to 8 shown by the second. In the $PRC$ based summary approach the reader is exposed to less than half number of features when he has to take his decision. The compression of relevant information is mainly due to the selection technique of $PRC$.

It is worth noting that the proposed test does not compare directly the two explanation systems ($PRC$ and frequency based). Thus the results could be affect by the high variability of users own behavior in the revision process. A feasible solution to limit this problem could be testing the target users with both two explanation systems.

## 4.4 Conclusions

In this chapter we have presented the use of TC for three most important NLP systems: Information Extraction, *Question/Answering* and Document Summarization.

First, we have developed a new learning method for automatically acquiring information extraction rules for new domains. In our experiments, the rules obtained performed extraction with high precision, thus enabling the coverage of any new extraction domain when they are further bootstrapped with additional relevant textual information. This two-pass semantic labeling technique we have developed performs with both human-like precision and recall for a large number of semantic frames. In our experiments we have employed the first release of FrameNet.

Second, we have presented five methods of categorizing questions and two methods of categorizing the answers produced by a Q/A system. Evaluation indicate that even with a question categorization method that does not perform as well as the answer categorization, the accuracy of Q/A can be improved in two ways: (1) by re-ranking the answers and by eliminating incorrect answers.

Finally, an explanation/summarization system based on TC has been presented. This includes two types of summaries that aim to improve the user satisfaction with respect to the delivered documents. The user, by simply reading the proposed summaries, can decide if the document meets his own interests. Both indicative and informative summaries are obtained by using a TC approach ($PRC$) together with a robust parser to select the concepts and paragraphs related to the target category. A preliminary evaluation of our explanation model has been carried out by testing the users' satisfaction. The summary-based explanation seems to be a promising solution for giving an explanation of the automatic categorization.

Chapter 3 has shown the useless of NLP for TC. In contrast, in this chapter preliminary studies on three main NLP applications have shown that TC can help to improve NLP effectiveness.

# Chapter 5

# Conclusions and Future Work

In this thesis the complex interaction between Natural Language Processing and Text Categorization has been studied. A specific attention has been devoted to the use of efficient NLP algorithms and efficient TC models, as their usefulness depends on their applicability in operational scenarios.

First, a study on improving the performance of very efficient profile-based classifiers (e.g., Rocchio) has been carried out. Original weighting schemes, score adjustment techniques and parameterization techniques have been proposed. In particular, the parameterization method designed for the Rocchio classifier, $PRC$, allows the Rocchio model to improve (at least 5% points) with respect to the best literature parameterization on every corpus. The results on *Reuters-21578* have shown that $PRC$ is the second[1] best figure classifier after $SVM$ in term of $f_1$ measure among the simple models (those not considered are classifier committees, boosting techniques and combined classifiers [Lam and Ho, 1998]). Moreover, the time complexity of $PRC$ is equal to the Rocchio's, i.e., the lowest among the not trivial classifiers [Sebastiani, 2002].

Second, the impact of syntactic and semantic document representations on TC accuracy has been studied. Syntactic features such as POS-tags as well as syntactic relations among words have been used to engineer complex linguistic features (i.e., phrases). The phrases experimented were proper nouns and complex nominals detected by NLP techniques. The results have shown that TC is not very much affected by this type of information. The main reasons that we have found are: (a) the words with ambiguous POS-tag are a small percentage, especially if features like numbers and special strings are included in the target feature set, and (b) in common natural language documents the sequences of words have the same indexing power of the single words. Phrases, in our as well

[1] $KNN$ measured on *Reuters-21578* in our studies as well as in other researches, e.g., [Joachims, 1998; Lam and Lai, 2001; Raskutti *et al.*, 2001; Toutanova *et al.*, 2001] has performance ranging between 80% and 82%.

as other researches seem slightly improve *weak* TC models. Our feeling is that the most of the improvements derive from a better suited parameterization when phrases are used (perhaps caused by these latter). In fact, $SVM$, that does not need estimation of parameters, is not improved by syntactic information. Semantic representation, when an accurate WSD is used, slightly improves the SVM accuracies. However, its performances are not clearly related to the accuracy of the WSD algorithm. Futher investigation is thus needed to determine the impact of WSD algorithms in TC.

Finally, preliminary experiments on the use of TC for three important tasks of NLP, Information Extraction, *Question/Answering* and Text Summarization have been carried out. The original idea of classifying sentences in FrameNet frames enables the possibility to model Open Domain Information Extraction systems whereas the question categorization seems viable to reduce the number of incorrect answers output by the Q/A systems. The powerful learning algorithms of TC allow to effectively model indicative and informative summaries related to a particular category.

Future research could be addressed to find a more effective algorithm that better exploits the feature selection interpretation of the Rocchio formula, given in this thesis. On the contrary, in our opinion there is small room for using complex representations for TC, derived by the current NLP techniques. Some literature work report positive results on the use of NLP for TC. We have shown that the quality of the outcomes are not statistically sufficient to assess the superiority of the NLP-driven models. Our feeling is that such results come more from the desire to govern the *cold* statistical models by means of (more *understandable*) symbolic approaches than from an evaluation sustained by empirical experimental data. The last chapter of this thesis, instead, has illustrated again that the statistical learning can be positively used to drive natural language processes as the non-so-recent NLP history has repeatedly shown. Thus, the use of Text Categorization for Natural Language Processing applications as either proposed in this thesis or in other original ways is a promising and exciting future research.

# Appendix A

# Notation

| | |
|---|---|
| $C$ | a category |
| $C_i$ | the category $i$ |
| $\mathcal{C}$ | collection of categories |
| $\vert\mathcal{C}\vert$ | number of categories |
| $f$ | a feature |
| $f_i$ | the $i$_th feature of the *corpus* |
| $W^i_f$ | the weight of $f$ in $C$ |
| $\vec{C}$ | vector representation of $C$, $\vec{C} =< W_{f_1},..,W_{f_N} >$ |
| $\vec{a}$ | $= \vec{C}$ |
| $d$ | a document |
| $P$ | set of positive documents for $C$ |
| $P_i$ | set of positive documents for $C_i$ |
| $\bar{P}$ | set of negative documents for $C$ |
| $\bar{P}_i$ | set of negative documents for $C_i$ |
| $w^d_f$ | the weight of $f$ in $d$ |
| $\vec{d}$ | vector representation of $d$, $\vec{d} =< w^d_{f_1},..,w^d_{f_N} >$ |
| $\phi$ | the classification binary function $\phi : D \rightarrow 2^C$ |
| $s_{di}$ | scalar product between document $d$ and category $i$ |
| $\sigma$ | threshold over $s_{di}$ (similarity) |
| $b$ | threshold over $s_{di}$ (in the hyperplane equation) |
| $M$ | total number of corpus documents |
| $M_f$ | total number of corpus documents that contains $f$ |
| $N$ | total number of corpus features |
| $m$ | maximum number of features in a document |
| $O$ | total occurrences of features |
| $O_f$ | total occurrences of feature $f$ |
| $o^d_f$ | occurrences of feature $f$ in $d$ |

| | |
|---|---|
| $IDF$ | Inverse Document Frequency |
| $IWF$ | Inverse Word Frequency |
| $Precision$ | Precision |
| $Recall$ | Recall |
| BEP | Breakeven point |
| $f_1$ | $f_1$ measure |
| $\mu Precision$ | Microaverage Precision |
| $\mu Recall$ | Microaverage Recall |
| $\mu BEP$ | Microaverage Breakeven point |
| $\mu f_1$ | Microaverage $f_1$ measure |

# Appendix B

# A sample of Reuters-21578 Terminology

abal_khail
abdel_jabbar
abdel_rahim
abdel_shakour
abdul_aziz
abdul_hadi
abdul_karim
abdul_rahim
abitibi_price
about_face
above_average
above_mentioned
above_normal
above_target
abu_dhabi
academy_of_sciences
accord_dealers
accord_miyazawa
account_deficit
accounting_method
accu_weather
acme_cleveland
acreage_reduction
acreage_reductions
across_the
ad_hoc
adams_russell
added_value
addis_ababa
adm_.
administration_officials
advanced_micro_devices
advo_system
aegean_sea
afl_cio
african_countries
after_effect
after_tax
after_write
afternoon_session
ag_brown
again_montagu
agfa_gevaert
agip_petroli
ago_usda
agreed_upon
agricultural_products
agricultural_stabilization
agriculture_committee
agriculture_department
agriculture_minister
agriculture_ministry
agriculture_secretary
agriculture_secretary_richard_lyng
agro_economist
agro_food
agro_industrial
aids_related
air_atlanta
air_canada
air_force
air_moving
airbus_industrie
akzo_dupont
al_aam
al_abdulla
al_ahmed
al_anba
al_anbaa
al_asadi
al_awsat
al_azzawi
al_bader
al_bukhoosh

al_chalabi
al_chalaby
al_ittihad
al_juaimah
al_khalifa
al_khatib
al_nahayan
al_otaibi
al_oteiba
al_qabas
al_qassem
al_rai
al_rashid
al_riyadh
al_sabah
al_salim
al_shaheen
al_sharq
al_tayer
al_thani
al_wattari
al_zubedei
al_zubeidi
ala_.
alcan_aluminum
alcan_australia
alex_._brown
all_cash
all_destination
all_embracing
all_new
all_of
all_out
all_party
all_saudi
all_star
all_suite
all_time
all_year
allan_hawkins
allan_leslie
allan_saunderson
allegheny_ludlum
allen_bradley
allen_wallis
allied_lyons
allied_signal
allied_stores
allis_chalmers
already_fragile
amerada_hess
american_brands
american_can
american_caught
american_cyanamid
american_express
american_flag
american_flagged
american_home_products
american_led
american_made
american_medical
american_motors
american_owned
american_petroleum_institute
american_pork_congress
american_realty
american_registered
american_soybean
american_soybean_association
american_stock_exchange
american_telephone_and_telegraph
american_transport
amsterdam_rotterdam
an_investor
analyst_richard
anchor_glass_container
andres_soriano
angeles_based
anglo_dutch
anheuser_busch
animal_borne
annual_capacity
annual_div
annual_meeting
annual_report
annual_revenues
antar_belzberg
anti_aircraft
anti_alchohol
anti_alcohol
anti_apartheid
anti_communist
anti_competitive
anti_crisis
anti_dumping
anti_ec
anti_government
anti_infective
anti_inflammatory
anti_inflation
anti_japanese
anti_peptic
anti_protectionism
anti_ship
anti_shipping
anti_takeover
anti_trust
anti_u
anti_viral
antimicrobial_resistant
antwerp_hamburg
api_says_distillate
apple_computer

appropriate_sized
approve_merger
apt_sat
ara_ghent
arab_states
arabian_sea
archer_daniels
argentina_brazil
argentine_grain
ariz_.
ark_.
arms_for
arms_length
army_corps_of_engineers
arthurs_jones
as_of
as_ofs
as_well_as
asa_backed
asa_sponsored
ashland_oil
ashton_tate
asia_pacific
asian_development_bank
asian_pacific
assistant_secretary_david
association_of_flight_attendants
at_and_t
athens_limestone
atlanta_based
atlantic_city
atlantic_coast
atlantic_research
atlantic_richfield
att_philips
attorney_general
attractive_boschwitz
aulnay_sous
australia_based
australia_new
australian_based
australian_prime
australian_wheat
australian_wheat_board
average_grade
average_price
averaged_out
avgs_mlns
avon_products
bache_securities
back_pay
bad_debt
bahamas_based
bahia_blanca
bahrain_based
bail_out
baker_chung
baker_hughes
balance_date
balance_of
balance_sheet
baltimore_based
banco_santander
band_four
band_three
bandar_abbas
bangkok_bank
bank_funded
bank_houston
bank_of_china
bank_wilmington
banking_group
banking_group_ltd
banking_sources
banking_system
banks_raise
banque_indosuez
barge_customers
barge_freight
barrel_per
base_1980
base_rate_cut
basis_points
bass_family
bass_led
bass_strait
baton_rouge
bay_area
bay_resources_ltd
be_acquired
be_privatised
bear_stearns
beef_producing
beggar_my
beghin_say
belgian_owned
belgo_factors
belgo_luxembourg
bell_atlantic
bell_telephone
below_cost
below_normal
below_six
bergen_richards
berliner_bank
bermuda_based
berth_sized
bertram_trojan
best_interests
best_known
beta_format
beteiligungs_ag
bethlehem_steel
better_than

beverly_hills
bidding_war
big_ticket
billion_bushels
billion_cubic_feet
billion_deposits
billion_dinars
billion_dlr
billion_dlr_customer_repurchase
billion_dlrs
billion_francs
billion_guilders
billion_lire
billion_marks
billion_pesos
billion_rand
billion_riyals
billion_yen
bio_chem
bio_synthetic
bio_vascular
bird_by
bisphenol_a
black_ruled
blue_print
bluebell_altamont
board_chairman
board_member
boart_msa
body_gatt
boise_cascade
boliden_ab
bon_yong
bond_equivalent
bond_futures
bond_market
bonn_based
bonus_issue
bonus_wheat
book_squaring
book_value
borg_warner
borrowing_facilities
borrowing_facility
borrowing_target
boston_based
boston_globe
brand_name
brazilian_coffee_institute
brazilian_loans
bread_making
break_even
break_free
brent_grade
brierley_investments
bristol_meyers
bristol_myers
british_aerospace
british_based
british_broadcasting_corporation
british_chancellor
british_columbia
british_designed
british_listed
british_made
british_operated
british_petroleum
british_steel
british_sugar
british_telecom
british_virgin_islands
broad_based
broad_scale
broadly_based
broadly_defined
broker_dealer
brokerage_firm
brown_afg
brown_forman
browning_ferris
brucellosis_free
brussels_based
btr_nylex
bu_sorghum
buchanan_smith
budget_cutting
budget_deficit
budget_deficits
budget_saving
budget_savings
buenos_aires
buffer_stock
build_up
building_materials
building_products
building_societies
building_society
built_in
bulk_carrier
bullion_coin
buoy_loading
burger_king
burns_fry
burr_brown
business_backed
business_combination
business_editor
business_loan
business_loans_fall
bust_up
buy_backs
buy_out
buy_outs
buying_tender

buys_dollars
buys_stake
by_means_of
by_product
by_products
c_itoh
cabinet_level
cable_and_wireless
cable_news_network
cable_systems
cable_television
caesars_world
cajamarquilla_spokesman
cal_mankowski
calendar_1987
calgary_based
california_based
calorie_conscious
canada_dome
canada_u
canadian_banks
canadian_dlr
canadian_led
canadian_money_supply
canadian_rapeseed
canadian_tire
canadian_u
canary_islands
cane_growing
capital_account
capital_expenditure
capital_expenditures
capital_flows
capital_goods
caracas_based
carbon_chloride
cargo_handling
cargo_preference
carl_icahn
carry_in
carrying_value
carsey_werner
carter_hawley
carter_wallace
case_by
cash_balance
cash_certificate
cash_distribution
cash_flow
cash_portion
cash_settled
casualty_property
cathay_pacific
cathay_pacific_airways
cathay_pacific_airways_ltd
cattle_ranching
cattle_slaughter
cattle_slaughter_guesstimates
cayman_islands
cc_bank
ccc_stocks
cd_roms
cdu_led
cebeco_handelsraad
cedar_rapids
cell_research
central_bank
central_bank_sets_lira
central_banks
centrale_credit
centrally_planned
centrally_run
centre_right
centre_west
cereals_management_committee
certain_circumstances
certain_conditions
certain_liabilities
certificate_case
chairman_david
chairman_designate
chairman_elect
chairman_paul
chamber_of_commerce
chamber_of_commerce_and_industry
champlin_petroleum
chancellor_of_the_exchequer
chao_ming
chapter_11
chapter_11_bankruptcy
charge_offs
charter_crellin
chase_amp
chase_manhattan
checking_account
chemical_business
chemical_industry
chesebrough_pond
chesebrough_ponds
chi_cheng
chicago_based
chicago_board_of_trade
chicago_mercantile_exchange
chicago_milwaukee
chief_economist
chief_executive
chief_executive_officer
chien_hsien
chien_kuo
chien_ming
chien_shien
china_based
china_daily
china_national

chinese_built
chinese_made
chip_makers
chloramphenicol_resistant
chou_wiest
chris_craft
chrysler_amc
chung_jung
cia_dia
ciba_geigy
cie_generale
cincinnati_based
circuit_court_of_appeals
citgo_petroleum
citicorp_capital_investors
city_resources
cjmf_fm
clark_equipment
class_action
clayton_yeutter
clean_up
cleveland_cliffs
close_cooperation
closely_held
closely_knit
closely_watched
cms_energy
cnt_per
co_backed
co_chairman
co_development
co_financing
co_inc
co_international
co_led
co_ltd
co_op
co_operate
co_operation
co_operative
co_ops
co_ordinate
co_ordinating
co_ordination
co_owned
co_partners
co_responsibility
co_sponsor
co_sponsored
co_steel
co_subsidiary
co_underwriters
coal_fired
coarse_grain
coast_guard
coca_cola
cocoa_exchange
coconut_planters
code_named
coffee_growing
coffee_producing
coin_operated
cold_rolled
cold_weather
colgate_palmolive
colo_.
colombian_pipeline
colorado_springs
columbus_based
comdata_network
come_back
comment_on
commerce_chemical
commerce_clearing_house
commerce_commission
commerce_department
commerce_secretary
commerce_secretary_malcolm
commercial_bank
commercial_banks
commercial_workers
commerzbank_ag
commission_house_representatives
commission_president_jacques
commodity_chemical
commodity_credit_corp
commodity_credit_corporation
commodity_exchange
commodity_pact
commodity_pacts
commodity_prices
common_equivalent
common_stock
community_wide
compact_disc
compagnie_francaise_des_petroles
company_controlled
company_owned
company_petrobras
company_petroleos
compaq_computer
comparative_figures
competitively_priced
completes_acquisition
completes_merger
completes_purchase
compounding_ratio
computer_aided
computer_associates
computer_based
computer_chip
computer_memories
computer_software
computer_systems

comsat_contel
confidence_building
confidential_information
congressional_sources
conoco_inc
conoco_statoil
consent_decree
conservation_program
conservation_service
conservative_party
consolidated_papers
consulting_firm
consumer_goods
consumer_oriented
consumer_price
consumer_prices
consumer_prices_rise
consumer_products
consuming_countries
contained_copper
continental_grain
continental_grain_co
control_data
convertible_debentures
cooper_basin
cooper_development
cooper_eromanga
copper_lead
copper_plated
core_businesses
cormier_navon
corn_growers
corn_sweetener
corn_u
corning_glass
corning_glass_works
corporate_purposes
corporate_raiders
corporation_tax
corpus_christi
corrected_elder
corrected_hecla
corrected_insituform
corrected_lilly
corrected_network
coruna_based
cost_control
cost_cutting
cost_price
cost_reduction
costa_mesa
costa_rica
cotton_y
council_meeting
council_session
counter_balanced
counter_bid
counter_bids
counter_offer
counter_productive
counter_proposal
counter_purchases
counter_reaction
country_by
courier_division
courier_operation
court_approved
court_of_appeals
cpi_u
cpi_w
craig_sloane
credit_card
credit_conditions
credit_guarantees
credit_rose
credit_starved
credit_suisse
credit_suisse_first_boston
creditanstalt_bankverein
creditor_banks
crisis_laden
cross_border
cross_channel
cross_compliance
cross_currency
cross_default
cross_rate
cross_rates
cross_shareholdings
cross_trades
crown_central_petroleum
crown_prince
crude_oil
crude_oil_prices
crude_palm
csr_esso
cts_vs
cubic_feet
cubic_meters
cummins_engine
cumulative_effect
cure_all
currency_based
currency_denominated
currency_fluctuations
currency_stability
current_account
current_account_deficit
current_account_surplus

## B.1 Acquisition Terminology Ranked by $PRC$

share
said_it
acquisition
stake
company
offer
merger
acquire
common
corp
group
unit
sell
stock
shareholder
acquired
buy
outstanding
transaction
subsidiary
mln_dlrs
cash
common_stock
complete
bid
tender_offer
exchange_commission
agree
undisclosed
purchase
takeover
investor
sale
subject
asset
disclose
securities
term
management
control
agreement
firm
approval
investment
completes
tender
hold
board
filing
cyclops
division
plc
commission
systems
letter
receive
seek
merge
definitive
gencorp
companies
intent
usair
buyout
propose
approve
principle
holdings
sells
signed
industries
business
definitive_agreement
deal
buys
financial
twa
proposal
dixons
affiliate
director
terminate
chrysler
partnership
holding
make
announce
financing
merger_agreement
purolator
acquires
court
borg-warner
plans
international
rights
says_it
allegheny
air
previously

## B.2 Acquisition complex terms Ranked by $PRC$

said_it
mln_dlrs
common_stock
tender_offer
exchange_commission
definitive_agreement
merger_agreement
says_it
new_york
be_acquired
make_acquisition
takeover_bid
seek_control
completes_acquisition
investment_purposes
loan_association
first_boston
merger_with
general_partners
talks_with
usair_group
merge_with
dixons_group
it_has
los_angeles
purolator_courier
merger_talks
june_29
chief_executive_officer
waste_management
june_1
caesars_world
american_motors
industrial_equity
co_inc
an_investor
has_no
regulatory_approvals
mark_iv
comment_on
american_express
wall_street
crazy_eddie
talks_on
joint_venture
june_19
says_it_is
rights_plan
trans_world_airlines
taft_broadcasting
cable_television
further_details
supermarkets_general
limited_partnership
dome_petroleum
first_federal
makes_acquisition
co_ltd
department_of_transportation
takes_over
harcourt_brace_jovanovich
life_insurance
first_union
nippon_life
book_value
newspaper_advertisement
may_seek
dlrs_per
piedmont_aviation
shareholder_approval
san_diego
waiting_period
due_diligence
hanson_trust
withdrawal_rights
annual_meeting
buys_stake
chief_executive
senior_management
first_national
dayton_hudson
dart_group
allied_stores
brokerage_firm
mts_acquisition
san_miguel
real_estate
bond_corp
co_subsidiary
hong_kong
open_market
an_investor_group
justice_department
becor_western
working_capital
annual_revenues
computer_memories
merrill_lynch
federal_court
great_western
dominion_textile
voting_power
new_york_stock_exchange
comdata_network
standard_oil
boliden_ab
june_30

british_printing
voting_trust
sells_unit
pc_acquisition
venture_capital
cable_and_wireless
hughes_tool
federal_savings
completes_purchase
earlier_today
eastman_kodak
financial_services
irwin_jacobs
brierley_investments
donald_trump
shareholders_approve
row_publishers
shearson_lehman_brothers
business_combination
investor_group
revlon_group
nova_corp
risk_arbitrage
baker_international
entertainment_marketing
general_acquisition
transportation_department
minimum_number
hostile_tender
best_interests
national_bank
lucky_stores
ic_gas
majority_interest
new_jersey
standstill_agreement
federal_home_loan_bank_board
consent_decree
certain_conditions
unit_sells
industrial_products
same_price
merger_with_baker
electrospace_systems
preference_shares
news_corp
williams_holdings
national_amusements
santa_fe
independent_directors
gates_learjet
general_electric
private_placement
martin_sosnoff
june_2
financial_advisers
department_of_justice
fort_lauderdale
financial_security
computer_associates
harcourt_brace
centrale_credit
federal_trade_commission
video_affiliates
national_distillers
san_francisco
transcanada_pipelines
security_pacific
consumer_products
expiration_date
renouf_corp
gabelli_group
firm_ups
edelman_group
systems_division
approve_merger
salt_lake_city
reed_international
ic_industries
investor_asher_edelman
registration_statement
mario_gabelli
financial_details
proxy_materials
communication_corp
minority_stake
american_security
hanson_industries
financial_group
union_pacific
year_ended
british_petroleum
texas_air
drexel_burnham_lambert
poison_pill
corporate_purposes
new_hampshire
patti_domm
certain_circumstances
march_30
fort_worth
shopping_centers
exercise_price
60_days
july_31
product_line
emery_air_freight
corp_offers
carl_icahn
scandinavia_fund
fairchild_semiconductor

## Appendix C

# A sample of Ohsumed Terminology

```
abnormal_blood
abnormal_blood_pressure
abnormal_findings
abnormal_gag_reflex
abnormal_groups
abnormal_heart
abnormal_heart_rate
abnormal_outcomes
abnormal_pulmonary_function
abnormal_regulation
absolute_incidence
absorption_process
acceptable_alternative
access_port
accurate_diagnosis
acid_administration
acid_antagonist
acid_aspiration
acid_composition
acid_concentrations
acid_load
acid_output
acid_sequence
acid_stone
acid_stones
action_potential_duration
acute_abdomen
acute_asthma
acute_chest
acute_chest_pain
acute_effect
acute_ethanol
acute_ethanol_administration
acute_ethanol_exposure
acute_ethanol_intoxication
acute_gastroenteritis
acute_graft
acute_hepatitis
acute_illness
acute_illnesses
acute_intervention
acute_lung_injury
acute_phase
acute_rejection
acute_stage
acute_stroke
acute_water_intoxication
ad_hoc
addictive_disorder
additional_cases
additional_group
additional_information
additional_therapy
adequate_therapy
admission_test
adult_height
adverse_consequences
adverse_effect
adverse_effects
adverse_event
adverse_events
adverse_outcome
adverse_outcomes
adverse_reactions
adverse_side_effects
after_adjustment
age_35_years
age_55
age_60_years
age_65
age_65_years
age_children
age_group
age_groups
```

age_range
aged_12
aged_35
aged_35_years
aged_40
aged_40_years
aged_50
aged_50_years
aged_65_years
aged_70_years
aged_75
ages_ranged
ages_ranging
aggressive_therapy
agricultural_trauma
air_leaks
air_space_volume
air_spaces
alcohol_abuse
alcohol_consumption
alcohol_dependence
alcohol_intake
alcohol_use
alcohol_withdrawal
alcoholic_beverage
alcoholic_hepatitis
alien_hand
alone_group
alpha_chain
alternative_methods
ambulance_staff
animal_model
animal_models
annual_incidence
annual_mortality_rate
anticoagulant_group
appropriate_methods
appropriate_therapy
appropriate_treatment
appropriate_use
arch_obstruction
artery_flow
artery_obstruction
artificial_heart
as_part
as_well_as
asbestos_bodies
aspirin_325
assist_device
asthma_attacks
asthma_severity
asthma_symptoms
attack_rates
attending_physicians
atypical_transformation_zone
average_age
average_annual_incidence
average_duration
average_time
back_pain
balloon_catheter
balloon_inflation
barrel_field
base_pairs
base_station
base_station_physician
basic_drive
basic_forms
basic_protein
bearing_mice
behavior_problems
beige_mice
beneficial_effect
beneficial_effects
benign_breast
benign_condition
benign_course
benign_diseases
benign_strictures
beta_cell
beta_cells
beta_chain
beta_degrees
beta_gene_expression
beta_production
better_control
better_delivery
better_predictor
better_prognosis
better_understanding
bicycle_exercise
bilateral_disease
bilateral_total_knee
bilateral_vocal_cord_paralysis
binding_domain
binding_properties
binding_protein
binding_proteins
binding_sites
birth_weight
black_women
bladder_cancer
bladder_capacity
bladder_compliance
bladder_function
bladder_neck
bladder_outflow
bladder_outlet
bladder_tumor
bladder_wall
bleeding_complications
bleeding_episodes
bleeding_risk
bleeding_tendency

bleeding_time
blind_design
blind_fashion
blind_trial
blocking_agents
blocking_effect
blocking_factor
blood_alcohol
blood_alcohol_concentration
blood_cell
blood_cell_count
blood_cells
blood_conservation
blood_culture
blood_culture_bottle
blood_culture_bottles
blood_cultures
blood_donation
blood_donors
blood_flow
blood_gas
blood_gas_analysis
blood_gases
blood_loss
blood_pressure
blood_pressure_readings
blood_products
blood_requirement
blood_sample
blood_samples
blood_sampling
blood_stages
blood_supply
blood_tests
blood_transfusion
blood_urea
blood_urea_nitrogen
blood_vessel
blood_vessels
blood_volume
blot_analysis
blunt_chest_trauma
blunt_trauma
body_fat_distribution
body_mass
body_mass_index
body_size
body_surface
body_surface_areas
body_surfaces
body_temperature
body_weight
bone_abnormalities
bone_defects
bone_density
bone_formation
bone_gap
bone_growth
bone_loss
bone_mass
bone_mineral
bone_mineral_content
bone_mineral_density
bone_mineral_status
bone_screws
border_zone
brain_areas
brain_barrier
brain_damage
brain_excitability
brain_stem
brain_tissue
brain_water
brain_water_content
breast_cancer
breast_conservation_surgery
breast_infection
breast_milk
breast_preservation
breast_tissue
breath_hydrogen
breath_test
breathing_controls
breathing_pattern
broad_spectrum
brown_product
burn_wound
burst_suppression
by_means_of
bypass_procedures
bypass_surgery
calcium_absorption
calcium_antagonist
calcium_antagonists
calcium_channel
calcium_channel_blockade
calcium_deposition
calcium_entry
calcium_flux_responses
calcium_handling
calcium_intake
calcium_release
calcium_salt
calcium_transport
cancer_cell_lines
cancer_fear
cancer_mortality
cancer_pain
carbon_dioxide
carbon_dioxide_laser
carbon_dioxide_pressure
carbon_dioxide_tension
cardiac_chamber
cardiac_complications

## C.1 Cardiovascular disease complex terms ranked by $PRC$

blood_pressure
coronary_artery_disease
heart_failure
coronary_artery
heart_rate
coronary_heart_disease
coronary_arteries
converting_enzyme
wall_motion
chronic_heart_failure
cardiac_index
heart_disease
calcium_antagonists
cardiac_output
cycle_length
segment_depression
wall_motion_abnormalities
calcium_channel
outflow_tract
total_cholesterol
coronary_artery_bypass
blood_flow
sudden_death
cardiac_cycle
pulmonary_artery
stroke_work
cardiac_events
peak_exercise
resistance_vessels
cardiac_performance
assist_device
balloon_inflation
peripheral_resistance
low_sodium
rate_pressure_product
density_lipoprotein_chole
regional_wall_motion
severe_coronary_artery
smooth_muscle_cells
defect_size
sudden_cardiac_death
bicycle_exercise
continuity_equation
chronic_coronary_artery_d
balloon_catheter
cardiac_function
sympathetic_activity
standard_balloon
coronary_circulation
exercise_capacity
three_vessel
201_imaging
chest_dogs
pulmonary_congestion
switch_operation
salt_diet
oxygen_demand
bypass_surgery
border_zone
collateral_circulation
pulmonary_artery_wedge
pulmonary_wedge_pressure
exercise_tolerance
end_points
sympathetic_nerve_activity
sodium_diet
calcium_channel_blockade
potential_importance
blood_pressure_readings
regional_wall_motion_abnormalities
energy_phosphate
human_arteries
work_load
deep_vein
severe_heart
late_death
life_support
standard_error
great_vessels
coronary_flow_reserve
flow_properties
primary_prevention
conventional_balloon
cardiac_rehabilitation
positive_exercise_test
calcium_handling
calcium_entry
quantitative_analysis
heart_attack
elective_coronary_artery_bypass
primary_success
wall_shear
low_density_lipoprotein_cholesterol
coronary_segments
age_55
dynamic_exercise
lowering_effect
basic_drive
calcium_antagonist
adverse_side_effects
risk_profile
sudden_deaths
driving_pressure
temporal_artery
mechanical_properties

# Appendix D

# A sample of ANSA Terminology

abolizione_di_reato
accertamento_da_parte_di_carabiniere
accoglienza_la_badessa
accordio_di_pace
accordo_polo
acqua_di_pacifico
acquaviva_delle_fonti
acquisto_di_massimiliano_cappioli
ad_eccezione_di
ad_esempio
ad_uso
aeroporto_militare
affare_costituzionale
affare_crespo
affermazione_di_centro
affetto_da_malattia
affollamento_di_carcere
agente_di_questura
agenzia_dpa
agevolazione_concesso_supero
aggiunto_amato
agip_petroli
agricoltura_alfonso_pecoraro
ai_sensi_di
aiuto_ammissibile_in_zona
al_centro_di
al_dettaglio
al_minuto
al_momento_di
alba_adriatica
albano_laziale
albissola_marina
albissola_superiore
all_estero
all_ingrosso
all_portata_di
alla_stregua_di
allargamento_di_unione
allarme_terrorismo
alleanza_con_bossi
alleanza_nazionale
alleato_bossi
aloisi_de_larderel
altezza_di_localita\′
altilia_di_santa_severina
altipiani_di_arcinazzo
amarezza_di_papa
ambasciatore_marco
america_del_nord
america_del_sud
american_cyanamid
american_express
american_home_products
ammesso_che
amministratore_delegato
amministratore_locale
amministratore_regionale
amministrazione_clinton
amministrazione_comunale
ammonito_clinton
ammortamento_di_titolo_di_stato
ampliamento_di_impianto
analista_finanziaria
anche_quando
anche_se
ancor_piu\′
andamento_di_economia
anna_maria
anno_cinquant
anno_consecutivo
annullamento_di_visita_di_khatami
annuncio_da_zurigo
apertura_di_inchiesta
apparecchiatura_elettronico

appello_di_papa
appiano_gentile
applicazione_di_riforma
appuntamento_sportivo
archivio_di_tradizione_orale
arcinazzo_romano
arco_alpino
area_christian
area_continuo
area_generale
arena_made_in_bo
argentino_ral_gimnez
arma_automatico
arma_da_fuoco
arrivo_di_nona_prova
articolo_pubblicato
artista_contemporaneo
ascoli_piceno
asian_development_bank
aspetto_umano
asse_francia
assegnazione_di_mondiali
assemblea_di_socio
assessorato_regionale
assessore_andrea
assicurazione_di_responsabilita’_civile
assistenza_sanitario
associazione_marco
assunzione_di_nuovo_personale
astensione_di_neozelandese
at_&_t
atollo_kwajalein
attaccante_brasiliano
attacco_sinistro
atterraggio_morbido_di_crescita
attesa_di_analista
attesa_di_giudizio
attivita’_culturale
atto_a
audizione_ministro
audizione_su_dpef
aula_consiliare
aumento_di_prezzo
aumento_di_tasso
auto_contenuto_in_pacchetto
autorita’_antitrust
autoveicolo_volkswagen
avente_per_oggetto
avventura_di_superman
avventura_europeo
avversario_politico
avvio_di_procedura
avvocato_difensore
azienda_americana
azienda_di_scarpa_sportivo
azione_ordinario
azione_usa
azionista_stabile
azzano_decimo
baco_di_millennio
bagni_di_tivoli
banca_agricola
banca_antoniana
banca_centrale
banca_commerciale
banca_d’_affare
banca_nazionale
banca_popolare
banco_santander
barile_di_greggio
basco_di_eta
base_aereo_di_vandenberg
baselga_di_pine’
bastia_umbra
battuto_in_finale
belga_frank
bella_cosa_di_mondo
bella_jordan
bene_culturale
bene_immobile
bene_mobile
beverly_hills
bianca_d’_epoca
bilancio_agricolo_europeo
bilancio_di_vittima
bill_clinton
bisogno_di_sicurezza
blocco_di_tariffa_rc
bolognese_alfeo_gigli
bordo_di_auto
borgo_valsugana
borsa_di_hong_kong
bosco_bruciato
bottiglia_di_birra
bozza_di_protocollo
braccio_destro
brasiliano_alex
brindisi_di_montagna
british_airways
british_petroleum
brokeraggio_assiprogetti
buenos_aires
buseto_palizzolo
busto_arsizio
c_._r_.
cable_and_wireless
cagnano_varano
calcio_marco
calo_di_prezzo_di_greggio
cambio_a_favore_di_destra
cambio_di_autorizzazione
camera_alto
camera_di_commercio

camerata_nuova
camp_david
campagna_d’_istruzione
campagna_di_scavo
campionato_europeo
campione_d’_europa
campione_di_manchester
campo_assicurativo
canale_televisivo
cancellazione_di_debito
cancelleria_friedrich
cancelliere_gerhard_schroeder
candidato_premier
candidato_vaccino
candidatura_africano
cantiere_edile
canto_suo
canzone_originale
capello_rosso
capitale_britannico
capitale_europeo
capitolo_in_studio_di_eta’
capo_di_diplomazia_europeo
capo_di_opposizione
capo_di_ufficio
capogruppo_ds
capoluogo_emiliano
cappelle_sul_tavo
carabiniere_di_ros
carattere_democratico_di_votazione
carcere_di_poggioreale
carcere_italiano
carica_di_commissario_tecnico
carlo_maria
carne_da_macello
carne_secca
carnevale_diverso
carriera_in_classe
carta_d’_identita’
cartello_di_prezzo
cartone_animato
casa_automobilistico
casa_bianca
casa_marco
casalecchio_di_reno
caso_di_depressione
caso_emerson
cassa_di_risparmio
cassa_di_stato
cassa_rurale
castagneto_carducci
castel_di_sangro
castelfranco_veneto
castellana_grotte
castelnuovo_rangone
castrocaro_terme
causa_di_fondo
causa_di_incendio
cava_dei_tirreni
cavasso_nuovo
cavo_in_regno_unito
celebrazione_di_quarto_centenario
cena_in_tema
cenate_sotto
centinaia_di_ettaro
cento_anno
centro_abitato
centro_civico
cerimonia_commemorativo
cerreto_guidi
cerreto_sannita
cervara_di_roma
cervignano_del_friuli
chiaramonte_gulfi
chiave_di_jesolo
chiesa_cattolico
chilo_di_cocaina
chilometro_da_parigi
chiusura_precedente
cielo_sereno
cifra_giusto
cinema_italiano
cinese_zhu_rongji
cinisello_balsamo
cinque_anno
circolazione_di_capitale
circolo_mario_mieli
circostanza_sospetto
circuito_differenziato
circuito_toscano
citta’_di_castello
cittadina_di_unita’
cittadino_britannico
cittadino_cubano
cividale_del_friuli
civita_castellana
classe_optimist
classico_napoletano
classifica_di_vendita
classifica_generale
clausola_in_contratto
cliente_straniero
clima_di_tensione
club_emiliano
co_.
coalizione_di_premier_ehud
coca_cola
coda_di_corteo
codice_di_procedura_penale
collaboratore_di_giustizia
collegamento_diretto
collezione_cittadina
collina_bolognese
colonna_ininterrotto

colore_di_medioevo
colosso_francese
colpo_d’_arma_da_fuoco
colpo_di_arma_da_fuoco
comandante_di_forza
comando_provinciale
come_se
comitato_irvin
commercializzazione_di_prodotto_assicurativo
commercio_estero_pascal
commesso_reato_in_italia
commissario_schreyer
commissione_affari_costituzionali
compagnia_aereo
compagnia_assicurativo
compagno_di_squadra
compaq_computer
compensazione_di_imposta
competenza_politica
compito_di_indirizzo_politico
compleanno_di_giancarlo_menotti
complesso_aziendale
complesso_di_intervento
complesso_di_materiale
componente_di_esecutivo_di_fifa
comunicato_di_ufficio_stampa
comunicazione_verbale
comunita’_albanese
con_esclusione_di
con_riferimento_a
concerto_di_gruppo
concessionario_d’_auto
concessione_di_credito
concessione_umts
conclusione_di_giornata_di_contrattazione
concorrenza_mario_monti
concorso_di_bellezza
condanna_di_volkswagen_da_parte_di_corte
condizione_atmosferico
condizione_generale
conferenza_james
conferimento_di_laurea
conflitto_di_interesse
conforme_a
confronto_fra_roma
congelamento_di_tariffa_di_contratto
congresso_nazionale
coniuge_lo_monaco
connesso_con
conquista_roma
conseguenza_di_premessa
conserve_italia
considerato_che
considerazione_di_interesse_generale
consigliere_comunale
consigliere_regionale
consiglio_amato
consiglio_amministrazione
consiglio_dei_ministri
consiglio_superiore_della_magistratura
consumatore_david
consumo_di_famiglia
contemporaneo_rene’_aubry
contenuto_di_visita_in_tunisia
continente_africano
contingente_tariffario
conto_amato
conto_corrente
conto_di
conto_proprio
conto_terzi
contratto_biennale
contratto_nazionale
controllo_congiunto
convegno_di_democratici_su_sicurezza
convenzione_quinquennale_fra_amministrazione
cooperazione_internazionale
coordinamento_lucano
coordinatore_regionale
copertura_finanziaria
coppa_davis
coppia_di_fatto
coppia_di_gay
cornice_di_parco_di_palazzo
corpo_forestale
corsa_in_alto_quota
corsia_d’_emergenza
corsia_preferenziale
corso_accertamento
corte_dei_conti
corte_di_assise
corte_franca
corte_italiano
corte_suprema
corteo_di_gay_pride
cosa_certo
cosa_concreto_molto
coscienza_in_attivita’
cosimo_damiano
costa_crociere
costa_di_inghilterra_meridionale
costa_smeralda
costituzione_federale
costo_di_denaro
costruzione_di_nuovo
creazione_di_nuovo_posto

# Appendix E

# Questions on the Reuters-21578 corpus

*What did the Champion Products Inc approve related to the shares?*
*What produces the swing in operating results?*
*What a large trade deficit with the U.S. can determine?*
*What did the SEC decide on future charges?*
*Which strategy aimed activities on core businesses?*
*What was a weakening Dollar responsible for?*
*How is the benefit of using the bank's international operations?*
*How does the continued growth in consumed lending affect the market?*
*What would impact the cost sharing for the research and development on the market?*
*What revenues does the records conversion work produce?*
*How does the higher earnings from the bank's own account contribute to record profits?*
*What do the analysts think about the repurchase program?*
*How do oil prices and the weak dollar affect the stock prices?*
*How could the transpacific telephone cable between the U.S. and Japan contribute to forming a join venture?*
*What solutions are available for the institutional debt?*
*What is the impact of West German competitors on the car market?*
*What has the Federal Communications Commission ordered about phone tariff?*
*What will be the Commerce Bank behavior with respect to market uncertainties?*
*How do the severe weather conditions impact on costs?*
*Which recapitalization plan included asset sales and equity offering parts?*
*What does the lowering of refining and sales profit margin determine?*
*What are the claims generated from personal auto insurance and the volatile commercial liability coverages?*
*Who required changes on the long-distance market laws?*
*What was the most significant factor for the lack of the distribution of the asset?*

*How do analysts think about public companies?*
*Which are the European companies interested in buying in the U.S.?*
*Why American Express remained silent?*
*What would happen if American Express reduced its exposure to the brokerage?*
*What is Shearson studying to access capital?*
*What is Ropak considering in order to acquire Buckhorn?*
*What do Japan Fund Inc and Sterling Grace Capital Management want to buy?*
*What economical advantages do the gold mining companies in Brazil provide?*
*Which companies have extended the deadline for accepting shares?*
*Who wanted to avoid any attack on the heart of its business empire?*
*What rumors circulated on Wall Street?*
*Why is the acquisition of a pharmaceutical too expensive?*
*How does an independent public company reflect on shareholders?*
*How is south Africa situation impacting on local companies?*
*Who was causing harm to the companies?*
*Who is planning to testify at the Senate hearing against raiders?*
*What industry is an attractive investment opportunity for Japanese corporations?*
*Why all risks have to be registered in a commission?*
*Where does American Nutrition operate?*
*Which company is subject to the boards of First Southern and Victor and regulatory agencies?*
*What is Kuwait known for?*
*What are European companies interested in buying in the U.S.?*
*Where do European companies acquire energy?*
*What is Washington considering for energy export?*
*What did the Director General say about the energy floating production plants?*
*What February production did energy commission indicate?*
*Why did the Reagan administration consider the export of oil to the Soviet Union?*
*How much oil was produced during the test in SOUTH AFRICA?*
*Why did Turkish Prime Minister intimate the stop of Greece drilling activities?*
*Where is Petro-Canada proposing a development for a drilling plant?*
*What is the opinion of Pickens about domestic energy?*
*Why are the major drilling companies exploring overseas?*
*Where is Ecuador Deputy Minister looking for energy help?*
*How much do buyers of U.S. pay for oil acquisition?*
*What do buyers say about oil location?*
*What did Grisanti approve in the assembly?*
*What are the recorded expected earning of USX?*
*What supply does Venezuela give to another oil producer?*
*How many Japanese companies will acquire Iranian oil ?*
*What did Silas say about the development of oil and gas company of Phillips?*
*What did Brazil's seafarers want?*
*Why was the port of Philadelphia closed?*
*What ship will be built for the Canadian coast guard?*

*How many vessels will the United States Lines provide?*
*Why do certain exporters fear that China may renounce its contract?*
*How is Kenya establishing a shipping line?*
*Why is Taiwan planning a joint production agreement with Japan?*
*What is needed for reaching a Soviet Baltic port?*
*What conditions halted shipping?*
*When did the strikes start in the ship sector?*
*What causes the limited shipping restrictions for the rivers?*
*Why is the shipping moving in the narrow Bosphorus?*
*Who attacked the Saudi Arabian supertanker in the United Arab Emirates sea?*
*Why is the entire Seaway already free of ice?*
*What was reported about movements in Harbour port?*
*Why did men in port's grain sector stop work?*
*How much will the Port of Singapore Authority spend?*
*What will the Asia Port project offer?*
*What was the position of the Reagan administration related to the Soviet Union?*
*Which anti-inflation plan made worsened the economical situation after one year?*
*What is the government planning to prevent the current account surplus from rising quickly?*
*How did the trade surplus and the reserves weaken Taiwan's position?*
*Why did Canadian negotiators open talks last summer?*
*What did Canadians learn about their domestic market?*
*What is the economic status of French Market?*
*Why is the French economy not well-adapted to demand?*
*What will happen to the Trading houses without a MITI export license?*
*Why is it important that the U.S. Market reduce the Chinese restrictions?*
*What is the Reagan Administration doing to obtain Japanese cooperation?*
*Which trade measure is the U.S. Senate Agriculture Committee considering?*
*What are Spain's plans for reaching European Community export level?*
*Why could China impact potential export markets and generate potential competition for U.S. industries?*
*What the Paris agreements called?*
*What is the cause for the Japanese surplus reduction?*
*What is necessary for resolving the subsidy problem?*
*Why could the benefits of the U.S. be smaller than those of Canada?*
*What did the South Korea's do to reduce its debit with the United States?*
*What were the reasons for the U.S. deficit?*

# Bibliography

[Abney, 1996] Steven Abney. Part-of-speech tagging and partial parsing. In G.Bloothooft K.Church, S.Young, editor, *Corpus-based methods in language and speech*. Kluwer academic publishers, Dordrecht, 1996.

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

[Arampatzis *et al.*, 2000] Avi Arampatzis, Jean Beney, C.H.A. Koster, and T.P. van der Weide. Incrementality, half-life, and threshold optimization for adaptive document filtering. In *the Nineth Text REtrieval Conference (TREC-9),Gaithersburg, Maryland*, 2000.

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

[Baayen *et al.*, 1995] R. H. Baayen, R. Piepenbrock, and L. Gulikers, editors. *The CELEX Lexical Database (Release 2) [CD-ROM]*. Philadelphia, PA: Linguistic Data Consortium, University of Pennsylvania, 1995.

[Barzilay and Elhadad, 1997] R. Barzilay and M. Elhadad. Using lexical chains for text summarization. In *In Proceedings of the Intelligent Scalable Text Summarization Workshop (ISTS'97), ACL, Madrid, 1997*, 1997.

[Basili and Moschitti, 2001] R. Basili and A. Moschitti. A robust model for intelligent text classification. In *Proceedings of the thirteenth IEEE International Conference on Tools with Artificial Intelligence, November 7-9, 2001 Dallas, Texas*, 2001.

[Basili and Moschitti, 2002] Roberto Basili and Alessandro Moschitti. Intelligent NLP-driven text classification. *International Journal on Artificial Intelligence Tools*, Vol. 11, No. 3, 2002.

[Basili and Zanzotto, 2002] Roberto Basili and Fabio Massimo Zanzotto. Parsing engineering and empirical robustness. *Natural Language Engineering*, to appear, 2002.

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

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

[Basili *et al.*, 1998b] R. Basili, M. Di Nanni, L. Mazzucchelli, M.V. Marabello, and M.T. Pazienza. NLP for text classification: the trevi experience. In *Proceedings of the Second International Conference on Natural Language Processing and Industrial Applications, Universite' de Moncton, New Brunswick (Canada)*, 1998.

[Basili *et al.*, 1998c] Roberto Basili, Maria Teresa Pazienza, and Fabio Massimo Zanzotto. Efficient parsing for information extraction. In *Proc. of the ECAI98*, Brighton, UK, 1998.

[Basili *et al.*, 1999] R. Basili, A. Moschitti, and M.T. Pazienza. A text classifier based on linguistic processing. In *Proceedings of IJCAI 99, Machine Learning for Information Filtering, http://www-ai.cs.uni-dortmund.de/EVENTS/IJCAI99-MLIF/papers.html*, 1999.

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

[Basili *et al.*, 2000b] R. Basili, A. Moschitti, and M.T. Pazienza. Robust inference method for profile-based text classification. In *Proceedings of JADT 2000, 5th International Conference on Statistical Analysis of Textual Data, Lausanne, Switzerland*, 2000.

[Basili *et al.*, 2000c] Roberto Basili, Maria Teresa Pazienza, and Michele Vindigni. Corpus-driven learning of event recognition rules. In *Proceedings of the Workshop on Machine Learning for Information Extraction, held jointly with ECAI 2000*, Berlin, Germany, 2000.

[Basili *et al.*, 2000d] Roberto Basili, Maria Teresa Pazienza, and Fabio Massimo Zanzotto. Customizable modular lexicalized parsing. In *Proc. of the 6th International Workshop on Parsing Technology, IWPT2000*, Trento, Italy, 2000.

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

[Basili *et al.*, 2002] R. Basili, A. Moschitti, and M.T. Pazienza. Empirical investigation of fast text classification over linguistic features. In *Proceedings* *of the 15th European Conference on Artificial Intelligence (ECAI2002), Lyon , France*, 2002.

[Basili *et al.*, 2003] Roberto Basili, Alessandro Moschitti, Maria Teresa Pazienza, and Fabio Massimo Zanzotto. Personalizing web publishing via information extraction. *Special Issue on Advances in Natural Language Processing, IEEE Intelligent System*, to appear, 2003.

[Bekkerman *et al.*, 2001] Ron Bekkerman, Ran El-Yaniv, Naftali Tishby, and Yoad Winter. On feature distributional clustering for text categorization. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 146–153. ACM Press, 2001.

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

[Buckley and Salton, 1995] Christopher Buckley and Gerald Salton. Optimization of relevance feedback weights. In *Proceedings of SIGIR-95*, pages 351–357, Seattle, US, 1995.

[Caropreso *et al.*, 2001] Maria Fernanda Caropreso, Stan Matwin, and Fabrizio Sebastiani. A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization. In *Idea Group Publishing, Hershey, US*, 2001.

[Charniak, 2000] Eugene Charniak. A maximum-entropy-inspired parser. In *In Proceedings of the 1st Meeting of the North American Chapter of the ACL*, pages 132–139, 2000.

[Chinchor *et al.*, 1998] N. Chinchor, E. Brown, and P. Robinson. The hub-4 ie-ne task definition version 4.8. Available online at `http://www.nist.gov/speech/tests/bnr/hub4_98/hub4_98.htm`, 1998.

[Chuang *et al.*, 2000] Wesley T. Chuang, Asok Tiyyagura, Jihoon Yang, and Giovanni Giuffrida. A fast algorithm for hierarchical text classification. In *Proceedings of DaWaK-00*, pages 409–418, London, UK, 2000. Springer Verlag, Heidelberg, DE.

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

[Church, 1988] K. A. Church. A stochastic parts program and noun phrase parser for unrestricted text. In *Proc. of Second Conference on Applied Natural Language Processing*, 1988.

[Clark *et al.*, 1999] P. Clark, J. Thompson, and B. Porter. A knowledge-based approach to question-answering. In *proceeding of AAAI'99 Fall Symposium on Question-Answering Systems. AAAI*, 1999.

[Cohen and Singer, 1999] William W. Cohen and Yoram Singer. Context-sensitive learning methods for text categorization. *ACM Transactions on Information Systems*, 17(2):141–173, 1999.

[Collins, 1997] Michael Collins. Three generative, lexicalized models for statistical parsing. In *Proceedings of the ACL and EACLinguistics*, pages 16–23, Somerset, New Jersey, 1997.

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

[Dagan *et al.*, 1997] Ido Dagan, Yael Karov, and Dan Roth. Mistake-driven learning in text categorization. In Claire Cardie and Ralph Weischedel, editors, *Proceedings of EMNLP-97, 2nd Conference on Empirical Methods in Natural Language Processing*, pages 55–63, Providence, US, 1997. Association for Computational Linguistics, Morristown, US.

[Daille, 1994] 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.

[Drucker *et al.*, 1999] Harris Drucker, Vladimir Vapnik, and Dongui Wu. Automatic text categorization and its applications to text retrieval. *IEEE Transactions on Neural Networks*, 10(5):1048–1054, 1999.

[Dumais *et al.*, 1998] Susan T. Dumais, John Platt, David Heckerman, and Mehran Sahami. Inductive learning algorithms and representations for text categorization. In Georges Gardarin, James C. French, Niki Pissinou, Kia Makki, and Luc Bouganim, editors, *Proceedings of CIKM-98, 7th ACM International Conference on Information and Knowledge Management*, pages 148–155, Bethesda, US, 1998. ACM Press, New York, US.

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

[Fellbaum, 1998] Christiane Fellbaum. *WordNet: An Electronic Lexical Database.* MIT Press., 1998.

[Fillmore, 1982] Charles J. Fillmore. Frame semantics. In *Linguistics in the Morning Calm*, pages 111–137, 1982.

[Furnkranz *et al.*, 1998] J. Furnkranz, T. Mitchell, and E. Rilof. A case study in using linguistic phrases for text categorization on the www. In *Working Notes of the AAAI/ICML, Workshop on Learning for Text Categorization*, 1998.

[Furnkranz, 1998] Johannes Furnkranz. A study using n-gram features for text categorization. Technical report oefai-tr-9830, Austrian Institute for Artificial Intelligence., 1998.

[Gale and Church, 1990] William Gale and Kenneth W. Church. Poor estimates of context are worse than none. *In Proceedings of the June 1990 DARPA Speech and Natural Language Workshop*, pages 283–287, 1990.

[Gildea and Jurasky, 2002] Daniel Gildea and Daniel Jurasky. Automatic labeling of semantic roles. *Computational Linguistic*, 28(3):496–530, 2002.

[Gövert *et al.*, 1999] Norbert Gövert, Mounia Lalmas, and Norbert Fuhr. A probabilistic description-oriented approach for categorising Web documents. In *Proceedings of CIKM-99*, pages 475–482, Kansas City, US, 1999. ACM Press, New York, US.

[Harabagiu and Maiorano, 2000] S. Harabagiu and S. Maiorano. Acquisition of linguistic patterns for knowledge-based information extraction. In *in Proceedings of LREC-2000, June 2000, Athens Greece*, 2000.

[Harabagiu *et al.*, 2000] S. Harabagiu, M. Pasca, and S. Maiorano. Experiments with open-domain textual question answering. In *Proceedings of the COLING-2000*, 2000.

[Harabagiu *et al.*, 2001] Sanda M. Harabagiu, Dan I. Moldovan, Marius Pasca, Rada Mihalcea, Mihai Surdeanu, Razvan C. Bunescu, Roxana Girju, Vasile Rus, and Paul Morarescu. The role of lexico-semantic feedback in open-domain textual question-answering. In *Meeting of the ACL*, pages 274–281, 2001.

[Hardy *et al.*, 2001] Hilda Hardy, Nobuyuki Shimizu, Tomek Strzalkowski, and Xinyang Zhang Liu Ting. Cross-document summarization by concept classifcation. In *Proceedings of the Document Understanding Conference*, New Orleans, U.S.A., 2001.

[Hirschman *et al.*, 1999] L. Hirschman, P. Robinson, L. Ferro, N. Chinchor, E. Brown, R. Grishman, and B. Sundheim. *Hub-4 Event99 General Guidelines and Templettes*. Springer, 1999.

[Hull, 1994] David Hull. Improving text retrieval for the routing problem using latent semantic indexing. In *Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval*, pages 282–291, Dublin, IE, 1994.

[I. Moulinier and Ganascia, 1996] G. Raskinis I. Moulinier and J. Ganascia. Text categorization: a symbolic approach. In *Proceedings of the Fifth 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*, pages 301–315, Las Vegas, US, 1995.

[Jacquemin, 2001] Christian Jacquemin, editor. *Spotting and Discovering Terms through Natural Language Processing.* The MIT Press, Cambridge, Massachussets, USA, 2001.

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

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

[Joachims, 1999] T. Joachims. T. joachims, making large-scale svm learning practical. In B. Schlkopf, C. Burges, and MIT-Press. A. Smola (ed.), editors, *Advances in Kernel Methods - Support Vector Learning*, 1999.

[Johnson and Fillmore, 2000] Christopher R. Johnson and Charles J. Fillmore. The framenet tagset for frame-semantic and syntactic coding of predicate-argument structure. In *In the Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics (ANLP-NAACL 2000), April 29-May 4, 2000, Seattle WA*, pages 56–62, 2000.

[Kan *et al.*, 2001] Min-Yen Kan, Kathleen R. McKeown, and Judith L. Klavans. Domain-specific informative and indicative summarization for information retrieval. In *Proceedings of the Document Understanding Conference*, New Orleans, U.S.A., 2001.

[Kilgarriff and Rosenzweig, 2000] A. Kilgarriff and J. Rosenzweig. English senseval: Report and results. In *English SENSEVAL: Report and Results. In Proceedings of the 2nd International Conference on Language Resources and Evaluation, LREC, Athens, Greece.*, 2000.

[Kohavi and John, 1997] Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[Kolcz *et al.*, 2001] Aleksander Kolcz, Vidya Prabakarmurthi, and Jugal Kalita. Summarization as feature selection for text categorization. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 365–370. ACM Press, 2001.

[Lam and Ho, 1998] Wai Lam and Chao Y. Ho. Using a generalized instance set for automatic text categorization. In *Proceedings of SIGIR-98*, 1998.

[Lam and Lai, 2001] Wai Lam and Kwok-Yin Lai. A meta-learning approach for text categorization. In *Proceedings of SIGIR-01, 24th ACM International Conference on Research and Development in Information Retrieval*, New Orleans, US, 2001. ACM Press, New York, US.

[Lewis and Gale, 1994] David D. Lewis and W. Gale. A sequential algorithm for training text classifiers. In *Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval*, pages 3–12, Dublin, Ireland, 1994.

[Lewis and Sebastiani, 2001] David D. Lewis and Fabrizio Sebastiani. Report on the workshop on operational text classification systems (otc-01). *ACM SIGIR Forum*, 35(2):8–11, 2001.

[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 SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval*, pages 298–306, Zürich, CH, 1996.

[Lewis, 1992] David D. Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In *Proceedings of SIGIR-92, 15th ACM International Conference on Research and Development in Information Retrieval*, pages 37–50, Kobenhavn, DK, 1992.

[McCallum, 1996] Andrew Kachites McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. http://www.cs.cmu.edu/ mccallum/bow, 1996.

[Mitchell, 1997] Tom Mitchell, editor. *Machine Learning.* McCraw Hill, 1997.

[Mladenić and Grobelnik, 1998] Dunja Mladenić and Marko Grobelnik. Word sequences as features in text-learning. In *Proceedings of ERK-98, the Seventh Electrotechnical and Computer Science Conference*, pages 145–148, Ljubljana, SL, 1998.

[Moschitti and Zanzotto, 2002] Alessandro Moschitti and Fabio Massimo Zanzotto. A robust summarization system to explain document categorization. In *Proceedings of RObust Methods in Analysis of Natural language Data (ROMAND02)*, Frascati, Italy, 2002.

[Moschitti *et al.*, 2003] Alessandro Moschitti, Paul Morarescu, and Sanda Harabagiu. Open domain information extraction via automatic semantic labeling. In *Proceedings of the 2003 Special Track on Recent Advances in Natural Language at the 16th International FLAIRS Conference*, St. Augustine, Florida, 2003.

[Moschitti, 2003a] Alessandro Moschitti. Is text categorization useful for word sense disambiguation or question answering? In *Proceedings of the 2nd Annual Research Symposium of the Human Language Technology Research Institute*, Dallas, Texas, 2003.

[Moschitti, 2003b] Alessandro Moschitti. A study on optimal parameter tuning for Rocchio text classifier. In Fabrizio Sebastiani, editor, *Proceedings of ECIR-03, 25th European Conference on Information Retrieval*, Pisa, IT, 2003. Springer Verlag.

[Ng *et al.*, 1997] H.T. Ng, W.B. Goh, and K.L. Low. Feature selection, preceptron learning and a usability case study for text categorization. In *Proceedings of SIGIR-97, 20th ACM International Conference on Research and Development in Information Retrieval*, pages 67–73, 1997.

[Nigam *et al.*, 1999] K. Nigam, J. Lafferty, and A. McCallum. Using maximum entropy for text classification. In *IJCAI-99 Workshop on Machine Learning for Information Filtering*, pages 61–67, 1999.

[Nigam *et al.*, 2000] Kamal Nigam, Andrew K. McCallum, Sebastian Thrun, and Tom M. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134, 2000.

[Pasca and Harabagiu, 2001] Marius A. Pasca and Sandra M. Harabagiu. High performance question/answering. In *Proceedings ACM SIGIR 2001*, pages 366–374. ACM Press, 2001.

[Pazienza, 1997] M.T. Pazienza, editor. *Information Extraction: a Multidisciplinary Approach to an Emerging Information Technology.* Springer_Verlag, Heidelberg, Germany, 1997.

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

[Raskutti *et al.*, 2001] Bhavani Raskutti, Herman Ferrá, and Adam Kowalczyk. Second order features for maximising text classification performance. In *Proceedings of ECML-01, 12th European Conference on Machine Learning.* Springer Verlag, Heidelberg, DE, 2001.

[Riloff and Jones, 1999] E. Riloff and R. Jones. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*, pages 474–479, 1999.

[Riloff, 1996] Ellen Riloff. Automatically generating extraction patterns from untagged text. In *AAAI/IAAI, Vol. 2*, pages 1044–1049, 1996.

[Robertson and Walker, 1994] Stephen E. Robertson and Steve Walker. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In *Proceedings of SIGIR-94*, pages 232–241, Dublin, IE, 1994.

[Rocchio, 1971] J.J. Rocchio. *Relevance feedback in information retrieval.* In G. Salton, editor, The SMART Retrieval System–Experiments in Automatic Document Processing, pages 313-323 Englewood Cliffs, NJ, Prentice Hall, Inc., 1971.

[Sable and Church, 2001] Carl Sable and Ken Church. Using bins to empirically estimate term weights for text categorization. In Lillian Lee and Donna Harman, editors, *Proceedings of EMNLP-01, 6th Conference on Empirical*

*Methods in Natural Language Processing*, pages 58–66, Pittsburgh, US, 2001. Association for Computational Linguistics, Morristown, US.

[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, 1989] G. Salton. *Automatic text processing: the transformation, analysis and retrieval of information by computer.* Addison-Wesley, 1989.

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

[Schapire *et al.*, 1998] Robert E. Schapire, Yoram Singer, and Amit Singhal. Boosting and Rocchio applied to text filtering. In W. Bruce Croft, A. Moffat, C. J. van Rijsbergen, R. Wilkinson, and J. Zobel, editors, *Proceedings of SIGIR-98*, pages 215–223, Melbourne, AU, 1998. ACM Press, New York, US.

[Schütze *et al.*, 1995] Hinrich Schütze, David A. Hull, and Jan O. Pedersen. A comparison of classifiers and document representations for the routing problem. In Edward A. Fox, Peter Ingwersen, and Raya Fidel, editors, *Proceedings of SIGIR-95, 18th ACM International Conference on Research and Development in Information Retrieval*, pages 229–237, Seattle, US, 1995. ACM Press, New York, US.

[Scott and Matwin, 1999] Sam Scott and Stan Matwin. Feature engineering for text classification. In Ivan Bratko and Saso Dzeroski, editors, *Proceedings of ICML-99, 16th International Conference on Machine Learning*, pages 379–388, Bled, SL, 1999. Morgan Kaufmann Publishers, San Francisco, US.

[Sebastiani, 2002] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[Singhal *et al.*, 1995] Amit Singhal, Chris Buckley, Mandar Mitra, and Gerard Salton. Pivoted document length normalization. Technical Report TR95-1560, Cornell University, Computer Science, November 29, 1995.

[Singhal *et al.*, 1997a] Amit Singhal, John Choi, Donald Hindle, and Fernando C. N. Pereira. ATT at TREC-6: SDR track. In *Text REtrieval Conference*, pages 227–232, 1997.

[Singhal *et al.*, 1997b] Amit Singhal, Mandar Mitra, and Christopher Buckley. Learning routing queries in a query zone. In *Proceedings of SIGIR-97*, pages 25–32, Philadelphia, US, 1997.

[Smeaton, 1999] Alan F. Smeaton. Using NLP or NLP resources for information retrieval tasks. In Tomek Strzalkowski, editor, *Natural language information retrieval*, pages 99–111. Kluwer Academic Publishers, Dordrecht, NL, 1999.

[Strzalkowski and Carballo, 1997] Tomek Strzalkowski and Jose Perez Carballo. Natural language information retrieval: TREC-6 report. In *Text REtrieval Conference*, 1997.

[Strzalkowski and Jones, 1996] Tomek Strzalkowski and Sparck Jones. NLP track at trec-5. In *Text REtrieval Conference*, 1996.

[Strzalkowski *et al.*, 1998] Tomek Strzalkowski, Gees C. Stein, G. Bowden Wise, Jose Perez Carballo, Pasi Tapanainen, Timo Jarvinen, Atro Voutilainen, and Jussi Karlgren. Natural language information retrieval: TREC-7 report. In *Text REtrieval Conference*, pages 164–173, 1998.

[Strzalkowski *et al.*, 1999] Tomek Strzalkowski, Jose Perez Carballo, Jussi Karlgren, Anette Hulth Pasi Tapanainen, and Timo Jarvinen. Natural language information retrieval: TREC-8 report. In *Text REtrieval Conference*, 1999.

[Sussua, 1993] M. Sussua. Word sense disambiguation for free-text indexing using a massive semantic network. In ACM Press New York, editor, *The Second International Conference on Information and Knowledge Management (CKIM 93)*, pages 67–74, 1993.

[Tan *et al.*, 2002] C.-M. Tan, Y.-F. Wang, and C.-D. Lee. The use of bigrams to enhance text categorization. *accepted for publication in Information Processing and Management*, 2002.

[Toutanova *et al.*, 2001] Kristina Toutanova, Francine Chen, Kris Popat, and Thomas Hofmann. Text classification in a hierarchical mixture model for small training sets. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 105–113. ACM Press, 2001.

[Tzeras and Artman, 1993] K. Tzeras and S. Artman. Automatic indexing based on bayesian inference networks. In *SIGIR 93*, pages 22–34, 1993.

[Van Rijsbergen, 1979] C. J. Van Rijsbergen, editor. *Information retrieval.* London: Butterworths, 1979.

[Vapnik, 1995] V. Vapnik. *The Nature of Statistical Learning Theory.* Springer, 1995.

[Voorhees, 1993] Ellen M. Voorhees. Using wordnet to disambiguate word senses for text retrieval. In Robert Korfhage, Edie M. Rasmussen, and Peter Willett, editors, *Proceedings of the 16th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Pittsburgh, PA, USA, June 27 - July 1, 1993*, pages 171–180. ACM, 1993.

[Voorhees, 1994] Ellen M. Voorhees. Query expansion using lexical-semantic relations. In W. Bruce Croft and C. J. van Rijsbergen, editors, *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Dublin, Ireland, 3-6 July 1994 (Special Issue of the SIGIR Forum)*, pages 61–69. ACM/Springer, 1994.

[Voorhees, 1998] Ellen M. Voorhees. Using wordnet for text retrieval. In C. Fellbaum, editor, *WordNet: An Electronic Lexical Database*, pages 285–303. The MIT Press, 1998.

[Wiener *et al.*, 1995] Erik D. Wiener, Jan O. Pedersen, and Andreas S. Weigend. A neural network approach to topic spotting. In *Proceedings of SDAIR-95, 4th Annual Symposium on Document Analysis and Information Retrieval*, pages 317–332, Las Vegas, US, 1995.

[Yang and Liu, 1999] Yiming Yang and Xin Liu. A re-examination of text categorization methods. 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 ICML-97*, pages 412–420, Nashville, US, 1997.

[Yang *et al.*, 2000] Yiming Yang, Thomas Ault, and Thomas Pierce. Combining multiple learning strategies for effective cross-validation. In Pat Langley, editor, *Proceedings of ICML-00, 17th International Conference on Machine Learning*, pages 1167–1182, Stanford, US, 2000. Morgan Kaufmann Publishers, San Francisco, US.

[Yang, 1994] Yiming Yang. Expert network: effective and efficient learning from human decisions in text categorisation and retrieval. In *Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval*, pages 13–22, Dublin, IE, 1994.

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

[Yangarber *et al.*, 2000] R. Yangarber, R. Grishman, P. Tapanainen, and S. Huttunen. Unsupervised discovery of scenario-level patterns for information extraction. In *Proceedings of the Sixth Conference on Applied Natural Language Processing, (ANLP-NAACL 2000)*, pages 282–289, 2000.

[Yarowsky, 2000] D. Yarowsky. Hierarchical decision lists for word sense disambiguation. In *Computers and the Humanities, 34(1-2).*, 2000.