Complex Linguistic Features for Text Classification: a comprehensive study

Alessandro Moschitti[1] and Roberto Basili[2]

[1] University of Texas at Dallas, Human Language Technology Research Institute
Richardson, TX 75083-0688, USA
alessandro.moschitti@utdallas.edu
[2] University of Rome Tor Vergata, Computer Science Department
00133 Roma (Italy),
basili@info.uniroma2.it

Abstract. Previous researches on advanced representations for document retrieval have shown that statistical *state-of-the-art* models are not improved by a variety of different linguistic representations. Phrases, word senses and syntactic relations derived by Natural Language Processing (NLP) techniques were observed ineffective to increase retrieval accuracy. For Text Categorization (TC) are available fewer and less definitive studies on the use of advanced document representations as it is a relatively new research area (compared to document retrieval).
In this paper, advanced document representations have been investigated. Extensive experimentation on representative classifiers, Rocchio and SVM, as well as a careful analysis of the literature have been carried out to study how some NLP techniques used for indexing impact TC. Cross validation over 4 different corpora in two languages allowed us to gather an *overwhelming evidence* that complex nominals, proper nouns and *word senses* are not adequate to improve TC accuracy.

1 Introduction

In the past, several attempts to design complex and effective features for document retrieval and filtering were carried out. Traditional richer representations included: document *Lemmas*, i.e. base forms of morphological categories, like nouns (e.g. *bank* from *banks*) or verbs (e.g. *work* from *worked,working*); *Phrases*, i.e. sentence fragments as word sequences; *word senses*, i.e. different meanings of content words, as defined in dictionaries.

Phrases can be divided in: (a) *simple n-grams*[3], i.e., sequences of words (e.g., *officials said*) selected by applying statistical techniques, e.g. *mutual information* or χ^2; (b) *Noun Phrases* such as Named Entities (e.g., *George Bush* or *Washington D.C.*) and other complex nominals (e.g., *satellite cable television system*); and (c) $<head, modifier_1, .., modifier_n>$ tuples in which the relations between the *head* word and modifier words are detected using syntactic parsers, e.g. [1]. Typical relations (used in [2]) are *subject-verb* or *verb-object*, e.g. in *Minister announces* and *announces plans.*

The aim of phrases is to improve the precision on concept matching. For example, incorrect documents that contain the word sequence *company acquisition*

[3] The term n-grams is traditionally referred to as the sequences of n characters from text but in this context they will be referred to as words sequences.

are retrieved by the query *language* + *acquisition*. Instead, if the word sequences are replaced by the complex nominals *company acquisition* and *language acquisition*, the incorrect documents will not be retrieved since partial matches are not triggered.

Word senses can be defined in two ways: (a) by means of an explanation, like in a dictionary entry or (b) by using other words that share the same sense, like in a thesaurus, e.g. WordNet [3]. 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*.

Phrases were experimented for the document retrieval track in TREC conferences [2, 4–6]. The main conclusion was that the higher computational cost of the employed Natural Language Processing (NLP) algorithms prevents their application in operative IR scenario. Another important conclusion was that the experimented NLP representations can increase basic retrieval models (which use only the basic indexing model e.g., SMART) that adopt simple stems for their indexing. Instead, if advanced statistical retrieval models are used such representations do not produce any improvement [5]. In [7] was explained that pure retrieval aspects of IR, such as the statistical measures of word overlapping between queries and documents is not affected by the NLP recently developed for document indexing.

Given the above considerations, in [7] were experimented NLP resources like WordNet instead of NLP techniques. WordNet was used to define a semantic similarity function between noun pairs. As many words are polysemous, a Word Sense Disambiguation (WSD) algorithm was developed to detect the right word senses. However, positive results were obtained only after the senses were manually validated since the WSD performance, ranging between 60-70%, was not adequate to improve document retrieval. Other studies [8–10] report the use of word semantic information for text indexing and query expansion. The poor results obtained in [10] show that semantic information taken directly from WordNet without performing any kind of WSD is not helping IR at all. In contrast, in [11] promising results on the same task were obtained after the word senses were manually disambiguated.

In summary the high computational cost of the adopted NLP algorithms, the small improvement produced[4] and the lack of accurate WSD tools are the reasons for the failure of NLP in document retrieval. Given these outcomes, why should we try to use the same NLP techniques for TC? TC is a subtask of IR, thus, the results should be the same. 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 theoretically motivated machine learning techniques that better select the document representations.

[4] Due to both the NLP errors in detecting the complex structures and the use of NLP derived features as informative as the *bag-of-words*.

- Categories differ from queries as they are static, i.e., a predefined set of training documents stably define the target category. Feature selection techniques can, thus, be applied to select the relevant features and filtering out those produced by NLP errors. Moreover, documents contain more words than queries and this enables the adoption of statistic methods to derive their endogenous information.
- Effective WSD algorithms can be applied to documents whereas this was not the case for queries (especially for the short queries). Additionally, recent evaluation carried out in SENSEVAL [12], has shown accuracies of 70% for verbs, 75 % for adjectives and 80% for nouns. These last results, higher than those obtained in [7], make viable the adoption of semantic representation as a recent paper on the use of senses for document retrieval [13] has pointed out.
- For TC are available fewer studies that employ NLP techniques for TC as it is a relatively new research area (compared to document retrieval) and several researches, e.g. [14–19] report noticeable improvements over the *bag-of-words*.

In this paper, the impact of richer document representations on TC has been deeply investigated on four corpora in two languages by using cross validation analysis. Phrase and sense representations have been experimented on three classification systems: Rocchio [20] and the Parameterized Rocchio Classifier (PRC) described in [21, 22], and SVM-light available at `http://svmlight.joachims.org/` [23, 24]. Rocchio and PRC are very efficient classifiers whereas SVM is one *state-of-the-art* TC model.

We chose the above three classification systems as richer representations can be really useful only if: (a) accuracy increases with respect to the *bag-of-words* baseline for the different systems, or (b) they improve computationally efficient classifiers so that they approach the accuracy of (more complex) state-of-art models. In both cases, NLP would enhance the TC *state-of-the-art*.

Unfortunately results, in analogy with document retrieval, demonstrate that the adopted linguistic features are not able to improve TC accuracy. In the paper, Section 2 describes the NLP techniques and the features adopted in this research. In Section 3 the cross corpora/language evaluation of our document representations is reported. Explanations of why the more sophisticated features do not work as expected is here also outlined. The related work with comparative discussion is reported in Section 4, whereas final conclusions are summarized in Section 5.

2 Natural Language Feature Engineering

The linguistic features that we used to train our classifiers are POS-tag information, i.e. syntactic category of a word (nouns, verbs or adjectives), phrases and word senses.

First, we used the Brill tagger [25][5] to identify the syntactic category (POS-tag) of each word in its corresponding context. The POS information performs

[5] Although newer and more complex POS-taggers have been built, its performance is quite good, i.e. $\sim 95\%$.

a first level of word disambiguation: for example for the word *book*, it decides which is the most suitable choice between categories like *Book Sales* and *Travel Agency*.

Then, we extracted two types of phrases from texts:

– Proper Nouns (PN), which identify entities participating to events described in a text. Most named entities are locations, e.g. *Rome*, persons, e.g. *George Bush* or artifacts, e.g. *Audi 80* and are tightly related to the topics.
– Complex nominals expressing domain concepts. Domain concepts are usually identified by multiwords (e.g., *bond issues* or *beach wagon*). Their detection produce a more precise set of features that can be included in the target vector space.

The above phrases increase the precision in categorization as they provide core information that the single words may not capture. Their availability is usually ensured by external resources, i.e. thesauri or glossaries. As extensive repositories are costly to be manually developed or simply missing in most domains, we used automated methods to extract both proper nouns and complex nominals from texts. The detection of proper nouns is achieved by applying a grammar that takes into a account capital letters of nouns, e.g., *International Bureau of Law*. The complex nominal extraction has been carried out using the model presented in [26]. This is based on an integration of symbolic and statistical modeling along three major steps: the detection of atomic terms *ht* (i.e. singleton words, e.g., *issue*) using IR techniques [27], the identification of admissible candidates, i.e. linguistic structures headed by *ht* (satisfying linguistically principled grammars), and the selection of the final complex nominals via a statistical filter such as the mutual information.

The phrases were extracted per category in order to exploit the specific word statistics of each domain. Two different steps were thus required: (a) a complex nominal dictionary, namely D_i, is obtained by applying the above method to training data for each single category C_i and (2) the global complex nominal set D is obtained by merging the different D_i, i.e. $D = \cup_i D_i$.

Finally, we used word senses in place of simple words as they should 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*.

As nouns can be disambiguated with higher accuracy than the other content words we decided to use sense representation only for them. We assigned the noun senses using WordNet [3]. In this dictionary words that share the same meaning (*synonyms*) are grouped in sets called *synsets*. 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 several 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[6]. Since it is not known how much WSD accuracy impacts on TC accuracy, we have implemented additionally to the baseline, a WSD algorithms based on the glosses information and we used an accurate WSD algorithm, developed by the LCC, *Language Computer Corporation* (`www.languagecomputer.com`). This algorithm is an enhancement of the one that won the SENSEVAL competition [12].

The gloss-based algorithm exploits the glosses that define the meaning of each synset. 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").
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"*. We process only the definition part by considering it as a *local context*, whereas the document where the target noun appears is considered as a *global context*. Our semantic disambiguation function selects the sense whose local context (or gloss) *best* matches the global context. The matching is performed by counting the number of nouns that are in both the gloss and the document.

3 Experiments on linguistic features

We subdivided our experiments in two steps: (1) the evaluation of phrases and POS information, carried out via Rocchio PRC and SVM over *Reuters3*, Ohsumed and ANSA collections and (2) the evaluation of semantic information carried out using SVM[7] on *Reuters-21578* and 20NewsGroups corpora.

3.1 Experimental set-up

We adopted the following collections:

- The *Reuters-21578* corpus, Apté split, (`http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html`). It includes 12,902 documents for 90 classes with a fixed split between testing and training (3,299 vs. 9,603).
- The *Reuters3* corpus [28] prepared by Y. Yang and colleagues (`http://moscow.mt.cs.cmu.edu:8081/reuters 21450/apte`). It includes 11,099 documents for 93 classes, with a split of 3,309 vs. 7,789 between testing and training.
- The ANSA collection [22], which includes 16,000 news items in Italian from the ANSA news agency. It makes reference to 8 target categories (2,000 documents each).
- The Ohsumed collection (`ftp://medir.ohsu.edu/pub/ohsumed`), including 50,216 medical abstracts. The first 20,000 documents, categorized under the 23 *MeSH diseases* categories, have been used in our experiments.
- The 20NewsGroups corpus (20NG) available at `http://www.ai.mit.edu/people/jrennie/20Newsgroups/` . It contains 19997 articles for 20 categories taken from the Usenet newsgroups collection. We used only the subject and the

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

[7] Preliminary experiments using Rocchio and PRC on word senses showed a clear lowering of performances.

body of each message. This corpus is different from Reuters and Ohsumed because it includes a larger vocabulary and words typically have more meanings.

To better study the impact of linguistic processing on TC, we have considered as baselines two set of tokens:

- *Tokens* set which contains a larger number of features, e.g., numbers or string with special characters. This should provide the most general *bag-of-words* results as it includes all simple features.
- *Linguistic-Tokens*, i.e. only the nouns, verbs or adjectives. These tokens are selected using the POS-information. This set is useful to measure more accurately the influence of linguistic information.

Together with the token sets we have experimented the feature sets described in Section 2, according to the following distinctions:

- Proper Nouns and Complex Nominals: +CN[8] indicates that the proper nouns and other complex nominals are used as features for the classifiers.
- Token augmented with their POS tags in context (+POS), e.g., *check*/N vs. *check*/V.

Table 1. Characteristics of Corpora used in the experiments.

Corpus Name	Docs	Cat.	*Tokens*	Tokens +POS+CN	*Ling.-Tokens*	noun senses with BL-WSD	Lang.	*test-set*
*Reuters*3	11,077	93	30,424	39,840	19,000	-	Eng.	30%
Ohsumed	20,000	23	42,481	46,054	-	-	Eng.	40%
ANSA	16,000	8	56,273	69,625	-	-	Ita.	30%
Reuters-21578	12,902	90	29,103	-	-	6,794	Eng.	30%
20NGs	19,997	20	97,823	-	-	13,114	Eng.	30%

+CN denotes a set obtained by adding to the target token set, the proper nouns and complex nominals extracted from the target corpus. This results in atomic features that are simple tokens or chunked multiwords sequences (PN or CN), for which POS tag is neglected. Notice that due to their unambiguous nature, the POS tag is not critical for PN and CN. +POS+CN denotes the set obtained by taking into account POS tags for lemmas, proper nouns and complex nominals.

It is worth noting that the NLP-derived features are added to the standard token sets (instead of replacing some of them), e.g. complex nominals and proper nouns are added together with their compounding words. This choice has been made as our previous experiments showed a decrease of classifier accuracies when the compounding words were replaced with one single *phrase-feature*. This has also been noted in other researches, e.g. [29]. The resulting corpus/feature set can be observed in Table 3.1 (the reported number of senses refers to the senses generated by the baseline WSD algorithm).

The classifiers use the *ltc* weighting scheme [27] and the following parameterization: (a) Rocchio and PRC thresholds are derived from validation sets, (b)

[8] Proper nouns are indeed a special case of complex nominals, thus we used a single label, i.e. +CN.

parameters, $\beta = 16$ and $\gamma = 4$, are used for Rocchio whereas PRC estimates them on validation sets (as described in [22]) and (c) the default parameters of *SVM-light* package are used for SVM.

The performances are evaluated using the Breakeven Point (BEP) and the f_1 measure for the single categories whereas the microaverage BEP (μBEP) and the microaverage f_1 measure (μf_1) are used in case of global performances of category sets [28].

3.2 Cross-corpora/classifier validations of Phrases and POS-information

In the following we show that cross validation and the adoption of the most general token set as baseline is advisable. For example if we had used the *Linguistic-Tokens* set (nouns, verbs and adjectives) for a single experiments on the standard *Reuters3 test-set*, we would have obtained the PRC results shown in Table 2.

We note that both POS-tags and complex nominals produce improvements when included as features. The best model is the one using all the linguistic features. It improves the *Linguistic-Tokens* model of $\sim$ 1.5 absolute points.

Table 2. Breakeven points of PRC over *Reuters3* corpus. The linguistic features are added to the *Linguistic-Tokens* set.

	Linguistic-Tokens	+CN	+CN+POS
μBEP (93 cat.)	82.15%	83.15%	83.60%

However, the baseline has been evaluated on a subset of the $Tokens$ set, i.e. the *Linguistic-Tokens* set; it may produce lower performance than a more general *bag-of-words*. To investigate this aspect, in the next experiments we have added the $Tokens$ set to the linguistic feature sets. We expect a reduction of the positive impact provided by NLP since the rate of tokens sensible to linguistic processing is lowered (e.g. the POS-tags of numbers are not ambiguous).

Moreover an alternative feature set could perform higher than the *bag-of-words* in a single experiment. The classifier parameters could be better suited for a particular *training/test-set* split. Note that redundant features affect the weighting scheme by changing the norma of documents and consequently the weights of other features. Thus, to obtain more general outcomes we have cross-validated our experiments on three corpora: *Reuters3*, Ohsumed and ANSA on three classifiers Rocchio, PRC and SVM using 20 random generated splits between *test-set* (30%) and *training-set* (70%). For each split we have trained the classifiers and evaluated them on the test data. The reported performances are the average and the Std. Dev. (preceded by the ± symbol) over all 20 splits.

Tables 3 shows the uselessness of POS information for *Reuters3* corpus as the measures in column 5 (+CN) and 6 (+POS+CN) assume similar values. SVM was ran on simple tokens (column 7) and on complex nominals (column 8) as they have been shown to bring more selective information in PRC. Similar type of evaluations are reported in tables 4 and 5.

The global performances (i.e. the microaverages) in all the tables show small improvements over the *bag-of-words* approach ($Tokens$ column). For example, PRC improves of 84.97% - 84.42% = 0.55 that is lower than 1.45 observed in Table 2. An explanation is that the cardinality of complex nominals in these

Table 3. Rocchio, PRC and SVM performances on different feature sets of the Reuters3 corpus

	Rocchio	PRC				SVM	
	Tokens	Tokens		$+CN$	+POS+CN	Tokens	+CN
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
μf_1 (93 cat.)	80.07±0.5	84.90±0.5	84.42±0.5	84.97±0.5	84.82±0.5	88.58±0.5	88.14±0.5

Table 4. Rocchio, PRC and SVM performances on different feature sets of the Ohsumed corpus

	Rocchio	PRC				SVM	
	Tokens	Tokens		$+CN$		Tokens	$+CN$
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
Cardiovas.	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
Digest.Sys.	59.24	71.49	71.50	71.28	71.46	74.11	73.23
Neonatal	41.84	49.98	50.05	52.83	52.71	48.55	51.81
μf_1 (23 cat.)	54.36 ±0.5	66.06 ±0.4	65.81±0.4	65.90±0.4	66.32±0.4	68.43±0.5	68.36±0.5

Table 5. Rocchio and PRC performances on different feature sets of the ANSA corpus

	Rocchio	PRC		
	Tokens	Tokens	+CN	+POS+CN
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
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
μf_1 (8 cat.)	61.76±0.5	71.00±0.4	71.80±0.4	72.37±0.4

experiments is rather lower than the cardinality of $Tokens$[9] resulting in a small impact on the microaverages. The SVM global performances are slightly penalized by the use of *NLP-derived* features. We also note that some classes are improved by the extended features, e.g. *Neonatal Disease & Abnormalities* in Ohsumed and *Politics* or *Economic Politics* in the ANSA corpus, but this should be consider as the normal *record of cases.*

[9] There is a ratio of about 15:1 between simple tokens and complex nominals.

3.3 Cross validation on word senses

In these experiments, we compared the SVM performances over *Tokens* against the performances over the semantic feature sets. These latter were obtained by merging the *Tokens* set with the set of disambiguated senses of the training document nouns. We used 3 different methods to disambiguate senses: the baseline, i.e. by picking-up the first sense, Alg1 that uses the gloss words and the Alg2 one of the most accurate commercial algorithm.

Additionally, we performed an indicative evaluation of these WSD algorithms on 250 manually disambiguated nouns extracted from some random *Reuters-21578* documents. Our evaluation was 78.43 %, 77.12 % and 80.55 % respectively for the baseline and the algorithms 1 and 2. As expected, the baseline has an accuracy quite high since (a) in Reuters the sense of a noun is usually the first and (b) it is easier to disambiguate nouns than verb or adjective. We note that using only the glosses, for an unsupervised disambiguation, we do not obtain systems more accurate than the baseline.

Table 6. Performance of SVM text classifier on the *Reuters-21578* corpus.

Category	*Tokens*	BL	Alg1	Alg2
earn	97.70±0.31	97.82±0.28	97.86±0.29	97.68±0.29
acq	94.14±0.57	94.28±0.51	94.17±0.55	94.21±0.51
money-fx	84.68±2.42	84.56±2.25	84.46±2.18	84.57±1.25
grain	93.43±1.38	93.74±1.24	93.71±1.44	93.34±1.21
crude	86.77±1.65	87.49±1.50	87.06±1.52	87.91±1.95
trade	80.57±1.90	81.26±1.79	80.22±1.56	80.71±2.07
interest	75.74±2.27	76.73±2.33	76.28±2.16	78.60±2.34
ship	85.97±2.83	87.04±2.19	86.43±2.05	86.08±3.04
wheat	87.61±2.39	88.19±2.03	87.61±2.62	87.84±2.29
corn	85.73±3.79	86.36±2.86	85.24±3.06	85.88±2.99
μf_1 (90 cat.)	87.64±0.55	88.09±0.48	87.80±0.53	87.98±0.38

Reuters-21578 and 20NewsGroups have been used in these experiments. The latter was chosen as it is richer, in term of senses, than the journalistic corpora. The performances are the average and the Std. Dev. (preceded by the ± symbol) of f_1 over 20 different splits (30% test-set and 70% training) for the single categories and the μf_1 for all category corpus.

Table 6 shows the SVM performances for 4 document representations: *Tokens* is the usual most general *bag-of-words*, BL stands for the baseline algorithm and Alg i stands for Algorithm i. We can notice that the presence of semantic information has globally enhanced the classifier. Surprisingly, the microaverage f-score (μf_1) of the baseline WSD method is higher than those of the more complex WSD algorithms. Instead, the ranking among Alg1 and Alg2 is the expected one. In fact, Alg2, i.e. the complex model of LCC, obtains an accuracy better than Alg1, which is a simpler algorithm based on glosses. 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 7. SVM μf_1 performances on 20NewsGroups.

Category	*Tokens*	BL	Alg1	Alg2
μf_1 (20 cat.)	83.38±0.33	82.91±0.38	82.86±0.40	82.95±0.36

Similar results have been obtained for 20NewGroups, i.e. adding semantic information does not improve TC. Table 7 shows that when the words are richer in term of possible senses the baseline performs lower than Alg2.

To complete the study on the word senses, instead to add them to the *Token* set, we replaced all the nouns with their (disambiguated) senses. We obtained lower performances (from 1 to 3 absolute points) than the *bag-of-words*.

3.4 Why do phrases and senses not help?

The NLP derived phrases seems to be bring more information than *bag-of-words*, nevertheless, experiments show small improvements for weak TC algorithms, i.e. Rocchio and PRC, and no improvement for theoretically motivated machine learning algorithm, e.g., SVM. We see at least two possible properties of phrases as explanations.

(*Loss of coverage*). Word information cannot be easily subsumed by the phrase information. As an example, suppose that (a) in our representation, *proper nouns* are used in place of their compounding words and (b) we are designing a classifier for the *Politics* category. If the representation for the proper noun *George Bush* is only the single feature `George_Bush` then every political test document containing only the word `Bush`, will not trigger the feature `George_Bush` typical of a political texts.

(*Poor effectiveness*). The information added by word sequences is poorer than word set. It is worth noticing that for a word sequence to index better than its word set counterpart, two conditions are necessary: (a) words in the sequence should appear not sequentially in some incorrect documents, e.g. *George* and *Bush* appear non sequentially in a sport document and (b) all the correct documents that contain one of the compounding words (e.g. *George* or *Bush*) should at the same time contain the whole sequence (*George Bush*). Only in this case, the proper noun increases precision while preserving recall. However, this scenario also implies that *George Bush* is a strong indication of "*Politics*" while words *Bush* and *George*, in isolation, are not indicators of such (political) category. Although possible, this situation is just so unlikely in text documents: many co-references usually are triggered by specifying a more common subsequence (e.g. *Bush* for *George Bush*). The same situation occurs frequently for the complex nominals, in which the head is usually used as a short referential.

The experiments on word senses show that there is not much difference between senses and words. The more plausible explanation is that the senses of a noun in documents of a category tend to be always the same. Moreover, different categories are characterized by different words rather than different senses. The consequence is that words are sufficient surrogates of exact senses (as also pointed out in [13]). This hypothesis is also supported by the accuracy of the WSD baseline algorithm, i.e. by selecting only the most frequent sense, it achieves a performance of 78.43% on *Reuters-21578*. It seems that almost 80%

of the times one sense (i.e. the first) characterizes accurately the word meaning in Reuters documents.

A general view of these phenomena is that *textual representations* (i.e. tokens/words) are always very good at capturing the overall semantics of documents, at least as good as linguistically justified representations. This is shown over all the types of linguistic information experimented, i.e. POS tags, phrases and senses. If this can be seen partially as a negative outcome of these investigations, it must said that it instead pushes for a specific research line. IR methods oriented to textual representations of document semantics should be firstly investigated and they should stress the role of words as vehicles of natural language semantics (as opposed to logic systems of semantic types, like ontologies). It suggests that a word centric approach should be adopted in IR scenarios by trying also to approach more complex linguistic phenomena, (e.g. structural properties of texts or anaphorical references) in terms of word-based representations, e.g. word clusters or generalizations in lexical hierarchies[10].

4 Related Work

The previous section has shown that the adopted NLP techniques slightly improve weak TC classifier, e.g. Rocchio. When more accurate learning algorithms are used, e.g. SVM, such improvements are not confirmed. *Do other advanced representations help TC?* To answer the question we examined some literature work[11] that claim to have enhanced TC using features different from simple words. Hereafter, we will discuss the reasons for such successful outcomes. In [14] advanced NLP has been applied to categorize the HTML documents. The main purpose was to recognize 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 [31] 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 departments 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 [32] and RIPPER [33]. The authors reported higher precisions when the NLP-representation is used in place of the *bag-of-words.* These improvements were only obtained for recall levels lower than 20%. It is thus to be noticed that the low coverage of linguistic patterns explains why they are so useful only in low recall *measures.* Just because of this, no evidence is provided about a general and effective implication on TC accuracy.

In [15] n-grams with $1 \leq n \leq 5$, selected by using an incremental algorithm, were used. The Web pages in two Yahoo categories, *Education* and *References*, were used as target corpora. Both categories contain a sub-hierarchy of many

[10] These latter, obviously, in a fully extensional interpretation.

[11] We purposely neglected the literature that did not find representation useful for TC e.g. [30].

other classes. An individual classifier was designed for each sub-category. The set of classifiers was trained with the n-grams observed in the few training documents available. Results showed that n-grams can produce an improvement of about 1% (in terms of $Precision$ and $Recall$) in the *References* and about 4 % for *Educational.* This latter outcome may represent a good improvement over the *bag-of-words.* However, the experiments are reported only on 300 documents, although cross validation was carried out. Moreover, the adopted classifier (i.e. the *Bayesian* model) is not very accurate in general. Finally, the target measures relate to a non standard TC task: many sub-categories (e.g., 349 for *Educational*) and few features.

In [34], results on the use of n-grams over the *Reuters-21578* and 20NewsGroups corpora are reported. n-grams were, as usual, added to the compounding words to extend the *bag-of-words.* The selection of features was done using simple document frequency. Ripper was trained with both n-grams and simple words. The improvement over the *bag-of-words* representation, for *Reuters-21578* was less than 1%, and this is very similar to our experimental outcomes referred to complex nominals. For 20NewsGroups no enhancement was obtained.

Other experiments of n-grams using Reuters corpus are reported in [18], where only bigrams were considered. Their selection is slightly different from the previous work since Information Gain was used in combination with the document frequency. The experimented TC models were Naive Bayes and Maximum Entropy [35] and both were fed with bigrams and words. On *Reuters-21578*, the authors present an improvement of ∼2 % for both classifiers. The accuracies were 67.07% and 68.90%[12] respectively for Naive Bayes and Maximum Entropy. The above performances (obtained with the extended features) are far lower than the *state-of-the-art.* As a consequence we can say that bigrams affect the complexity of learning (more complex feature make poor methods more performant), but they stil not impact on absolute accuracy figures. The higher improvement reported for another corpus, i.e. some *Yahoo* sub-categories, cannot be assessed, as results cannot be replicated. Note in fact comparison with experiments reported in [15] are not possible, as the set of documents and *Yahoo* categories used there are quite different.

On the contrary, [16] reports bigram-based SVM categorization over *Reuters-21578.* This enables the comparison with (a) a state-of-art TC algorithm and (b) other literature results over the same datasets. The feature selection algorithm 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 employed 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. These weights are applied 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

[12] They used only the top 12 populated categories. Dumais reported for the top 10 categories a μf_1 of 92 % for SVM [36].

test-set. When bigrams were added, *SVM* performed 86.2% by improving about 0.6% the adopted token set. This may be important because to our knowledge it is the first improvement on SVM using phrases. However, it is worth considering that:

- Cross validation was not applied: the fact that *SVM* is improved on the Reuters fixed *test-set* only does not prove that *SVM* is generally enhanced. In fact, using cross validation we obtained (over *Tokens*) 87.64% (similar to the results found in [36] that is higher than the bigram outcome of Raskutti et al. [16]
- If we consider that the Std. Dev., in our and other experiments [17], are in the range $[0.4, 0.6]$, the improvement is not sufficient to statistically assess the superiority of the bigrams.
- Only, the words were used, special character strings and numbers were removed. As it has been proven in Section 3.2 they strongly affect the results by improving the unigram model. Thus we hypothesize that the baseline could be even higher than the reported one (i.e. 85.6%).

On the contrary, another corpus experimented in [16], i.e., *ComputerSelect* shows higher *SVM* μBEP when bigrams are used, i.e. 6 absolute percent points. But again the *ComputerSelect* collection is not standard. This makes difficult to replicate the results.

The above literature shows that in general the extracted phrases do not affect accuracy on the Reuters corpus. This could be related to the structure and content of its documents, as it has been also pointed out in [16]. Reuters news are written by journalists to disseminate information and hence contain few and precise words that are useful for classification, e.g., *grain* and *acquisition*. On the other hand, other corpora, e.g. *Yahoo* or *ComputerSelect*, include more technical categories with words, like *software* and *system*, which are effective only in context, e.g., *network software* and *array system*.

It is worth noticing that textual representations can here be also seen as a promising direction. In [17], the Information Bottleneck (IB), i.e. a feature selection technique that cluster similar features/words, was applied. *SVM* fed with IB derived clusters was experimented on three different corpora: *Reuters-21578*, WebKB and 20NewsGroups. Only 20NewsGroups corpus showed an improvement of performances when IB method was used. This was explained as a consequence of the corpus "complexity". Reuters and WebKB corpora seem to require fewer features to reach optimal performance. IB can thus be adopted either to reduce the problem complexity as well as to increase accuracy by using a simpler representation space. The improvement on 20NewsGroups, using the cluster representation, was ~ 3 percent points.

5 Conclusions

This paper reports the study of advanced document representation for TC. First, the tradition related to NLP techniques for extracting linguistically motivated features from document has been followed. The most widely used features for

IR, i.e. POS-tag, complex nominals, proper nouns and word senses, have been extracted.

Second, several combination of the above feature sets have been extensively experimented with three classifiers Rocchio, PRC and SVM over 4 corpora in two languages. The purpose was either to improve significantly efficient, but less accurate, classifiers, such as Rocchio and PRC, or to enhance a *state-of-the-art* classifier, i.e. SVM. The results have shown that both semantic (word senses) and syntactic information (phrases and POS-tags) cannot achieve any of our purposes. The main reasons are their poor coverage and weak effectiveness. Phrases or word senses are well substituted by simple words as a word in a category assumes always the same sense, whereas categories differ on words rather than on word senses.

However, the outcome of this careful analysis is not a negative statement on the role of complex linguistic features in TC but suggests that the elementary textual representation based on words is very effective. We emphasize the role of words, rather than some other logical system of semantic types (e.g. ontologies), as a vehicle to capture phenomena like event extraction and anaphora resolution. Expansion (i.e. the enlargement of the word set connected to a document or query) and clustering are another dimension of the same line of thought.

References

1. Collins, M.: Three generative, lexicalized models for statistical parsing. In: Proceedings of the ACL and EACL, Somerset, New Jersey (1997) 16–23
2. Strzalkowski, T., Jones, S.: NLP track at TREC-5. In: Text REtrieval Conference. (1996)
3. Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press. (1998)
4. Strzalkowski, T., Carballo, J.P.: Natural language information retrieval: TREC-6 report. In: TREC. (1997)
5. Strzalkowski, T., Stein, G.C., Wise, G.B., Carballo, J.P., Tapanainen, P., Jarvinen, T., Voutilainen, A., Karlgren, J.: Natural language information retrieval: TREC-7 report. In: TREC. (1998)
6. Strzalkowski, T., Carballo, J.P., Karlgren, J., Hulth, A., Tapanainen, P., Jarvinen, T.: Natural language information retrieval: TREC-8 report. In: TREC. (1999)
7. Smeaton, A.F.: Using NLP or NLP resources for information retrieval tasks. In Strzalkowski, T., ed.: Natural language information retrieval. Kluwer Academic Publishers, Dordrecht, NL (1999) 99–111
8. Sussua, M.: Word sense disambiguation for free-text indexing using a massive semantic network. In New York, A.P., ed.: Proceeding of CKIM 93. (1993)
9. Voorhees, E.M.: Using wordnet to disambiguate word senses for text retrieval. In: Proceedings of SIGIR 1993, PA, USA. (1993)
10. Voorhees, E.M.: Query expansion using lexical-semantic relations. In: Proceedings of SIGIR 1994. (1994)
11. Voorhees, E.M.: Using wordnet for text retrieval. In Fellbaum, C., ed.: WordNet: An Electronic Lexical Database, The MIT Press (1998) 285–303
12. Kilgarriff, A., Rosenzweig, J.: 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)

13. Stokoe, C., Oakes, M.P., Tait, J.: Word sense disambiguation in information retrieval revisited. In: Proceedings of SIGIR03, Canada. (2003)
14. Furnkranz, J., Mitchell, T., Rilof, E.: A case study in using linguistic phrases for text categorization on the www. In: AAAI/ICML Workshop. (1998)
15. Mladenić, D., Grobelnik, M.: Word sequences as features in text-learning. In: Proceedings of ERK98, Ljubljana, SL (1998)
16. Raskutti, B., Ferrá, H., Kowalczyk, A.: Second order features for maximising text classification performance. In: Proceedings of ECML-01, 12th European Conference on Machine Learning, Springer Verlag, Heidelberg, DE (2001)
17. Bekkerman, R., El-Yaniv, R., Tishby, N., Winter, Y.: On feature distributional clustering for text categorization. In: Proceedings of the ACM SIGIR 2001, ACM Press (2001) 146–153
18. Tan, C.M., Wang, Y.F., Lee, C.D.: The use of bigrams to enhance text categorization. Information Processing & Management (2002)
19. Scott, S., Matwin, S.: Feature engineering for text classification. In: Proceedings of ICML-99, Bled, SL (1999)
20. Rocchio, J.: 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)
21. Basili, R., Moschitti, A., Pazienza, M.: NLP-driven IR: Evaluating performances over text classification task. In: Proceedings of IJCAI01, USA. (2001)
22. Moschitti, A.: A study on optimal parameter tuning for Rocchio text classifier. In Sebastiani, F., ed.: Proceedings of ECIR-03, 25th European Conference on Information Retrieval, Pisa, IT, Springer Verlag (2003)
23. Vapnik, V.: The Nature of Statistical Learning Theory. Springer (1995)
24. Joachims, T.: T. joachims, making large-scale svm learning practical. In: Advances in Kernel Methods - Support Vector Learning. (1999)
25. Brill, E.: A simple rule-based part of speech tagger. In: Proc. of the Third Applied Natural Language Processing, Povo, Trento, Italy. (1992)
26. Basili, R., De Rossi, G., Pazienza, M.: Inducing terminology for lexical acquisition. In: Preoceeding of EMNLP 97 Conference, Providence, USA. (1997)
27. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information Processing and Management **24(5)** (1988) 513–523
28. Yang, Y.: An evaluation of statistical approaches to text categorization. Information Retrieval Journal (1999)
29. Caropreso, M.F., Matwin, S., Sebastiani, F.: A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization. In: Idea Group Publishing, Hershey, US. (2001)
30. Lewis, D.D.: An evaluation of phrasal and clustered representations on a text categorization task. In: Proceedings of SIGIR-92, Kobenhavn, DK (1992)
31. Riloff, E.: Automatically generating extraction patterns from untagged text. In: AAAI/IAAI, Vol. 2. (1996) 1044–1049
32. Mitchell, T.: Machine Learning. McGraw Hill (1997)
33. Cohen, W.W., Singer, Y.: Context-sensitive learning methods for text categorization. ACM Transactions on Information Systems **17** (1999) 141–173
34. Furnkranz, J.: A study using n-gram features for text categorization. Technical report oefai-tr-9830, Austrian Institute for Artificial Intelligence. (1998)
35. Nigam, K., Lafferty, J., McCallum, A.: Using maximum entropy for text classification. In: IJCAI-99 Workshop. (1999)
36. Dumais, S.T., Platt, J., Heckerman, D., Sahami, M.: Inductive learning algorithms and representations for text categorization. In: Proceedings of CIKM-98, Bethesda, US, ACM Press, New York, US (1998) 148–155