Combining Qualitative and Quantitative Keyword Extraction Methods with Document Layout Analysis

Stefano Ferilli[1], Marenglen Biba[1], Teresa M.A. Basile[1], Floriana Esposito[1]
[1] Università di Bari, Dipartimento di Informatica,
via E. Orabona 4, 70126 Bari, Italy
{ferilli, biba, basile, esposito}@di.uniba.it

Abstract. The large availability of documents in digital format posed the problem of efficient and effective retrieval mechanisms. This involves the ability to process natural language, which is a significantly complex task. Traditional algorithms based on term matching between the document and the query, although efficient, are not able to catch the intended meaning of both, and hence cannot ensure effectiveness. To step on toward semantics, problems such as polysemy and synonimy must be tackled automatically by text processing systems. This work aims at introducing in the document processing framework of DOMINUS qualitative techniques based on the lexical taxonomy WordNet and its extension WordNet Domains for text categorization and keyword extraction, that can support the currently embedded techniques based on quantitative approaches. In particular, a density function is exploited to assign the proper importance to the involved concepts and domains. Preliminary results on texts of different subjects confirm its effectiveness.

Keywords: Lexical taxonomies, Text Categorization, Keyword Extraction.

1 Introduction

In the last years the amount of available documents in digital format has grown exponentially, which affects the retrieval of interesting and significant information at need (a problem known as "information overload") and requires the development of proper techniques that improve the performance as regards amount and quality of the returned documents. Much of the document content is in the form of text (and hence unstructured)[1], which has been a significant motivation to the development of Natural Language Processing (NLP) techniques. The outcome of such techniques is the input to further processing aimed at indexing the documents and extracting information from them, in order to support information retrieval. In NLP two important application fields can be found: Information Retrieval (IR) and Information Extraction (IE). IR aims at selecting a relevant set of documents from a larger dataset, as an answer to a user query which expresses with a set of terms his information need. IE aims at identifying useful and relevant information that describes the content of a set of unstructured texts, and to report them in a (semi-)structured format suitable for

[1] In the following, a "document" will be intended as the set of text blocks contained in it, possibly labeled according to the role they play in the document.

filling a database or for exploitation by computers. Thus, the latter can be a significant aid to the former, in that basing document indexing (and hence IR) only on the relevant information preliminarily obtained by IE can improve the query result quality. For instance, it would be helpful to be able to search documents according to their content category, and later exploit keywords to better specify their content.

The objective of developing information extraction techniques that are able to catch the intended meaning of the text is very hard, due to peculiar ambiguities of natural language: synonymy, polysemy, phraseology (also known as *n*-grams), specific and technical terms [1]. For instance, the Italian word "Calcio" may stand for the name of a city, a chemical element, a part of a gun, or the action of kicking. In order to tackle the complexity of natural language, NLP tasks are usually divided in progressively higher-level and complex steps:

- *lexical analysis* breaks a text into tokens (usually corresponding to words);
- *syntactic analysis* organizes the tokens in a hierarchical structure according to their grammatical role;
- *semantic analysis* associates a meaning to the syntactic structure and thus, indirectly, to the text itself.

This work aims at extending the functionality of DOMINUS, a document processing framework, with text categorization and keyword extraction based on qualitative approaches applied to the lexical level only.

Text Categorization (TC) is the task of classifying words in natural language into specific categories belonging to a pre-defined set [2], or of assigning automatically a category to a corpus of documents. More formally, given a set of classes of interest and a set of documents already categorized in those classes (training set), the aim is building a decision function (classifier) that can map new documents (test set) to one or more classes according to their content. Hence, two main steps can be identified: learning and categorization. In the former, the system operates on the training set to learn information about the categories and the way to distinguish them, this way building a classifier. In the latter, using such a classifier new documents can be classified according to the given categories.

Keyword Extraction (KE) [3] is an information extraction task that aims at representing the essence of the intended message carried by the document according to the terms exploited. Two main approaches are present in the literature to tackle this problem. The quantitative approach assumes that a simple list of words included in the text can represent its subject. However, this approach, although providing a rough classification for the text, does not allow to organize it in sub-categories. To do this, semantics must be taken into account, that is the focus of the qualitative approach. This involves a semantic description of lexical objects in the text, that takes into account possible semantic domains and relationships, and can result in more specific and reliable outcomes than the quantitative approach.

In the following, after presenting the work that is at the basis of our experiment, the peculiarities of DOMINUS that make it suitable to such an integration will be introduced along with the experimental outcomes confirming the proposal viability.

2. Preliminaries

The increasing availability of linguistic resources as a support to NLP tasks, has allowed the proposed techniques approaching more closely the semantic level. One of the most famous and widely used such resources is WordNet, a lexical knowledge base designed to associate terms with a semantics based on groups of synonyms.

2.1 Ontologies

Born as a philosophical discipline, ontologies have developed in Information technology as exhaustive and rigorous conceptual schemata to formally describe a given domain. They have gained a fundamental importance with the spread of the Internet, since they represent a tool by which computer can exchange information based on its semantics rather than simple syntax. According to Tom Gruber, "*an ontology is an explicit specification of a conceptualization*" [4].

Lexical ontologies aim at characterizing (part of) a language independently of the domain, by expressing a lexical knowledge, made up of a set of words (intended as character strings), and a semantic knowledge, that encompasses word meanings and relations between words. WordNet [5, 6] is a famous lexical knowledge base aimed at overcoming the limitations of one-dimensional dictionaries. It groups terms in synsets (synonymous sets), partitioned into nouns, verbs, adjectives and adverbs. A polysemic term will belong to different synsets, and some relationships among synsets (such as hyperonimy, meronimy, etc.) are specified. Thus, WordNet is an outstanding candidate for supporting all tasks that are related to synonymy, in contrast to classical term-based representations of documents.

However, relationships expressed in WordNet are not exhaustive, so that some semantically related terms are not linked in it (e.g., 'doctor' and 'hospital').Considering such connections in terms co-occurring in the same text would be of great help in stepping on from the syntactic/quantitative level to a semantic/qualitative one. In fact, domain labels (e.g., medicine, sports, etc.) are a powerful way to establish semantic relations between terms, and represent a fundamental semantic property on which text consistency is based, in the sense that terms in the same text tend to belong to the same domain(s). Hence, a limited number of terms (typically nouns) can determine the main domain of a text, and be of great help in text disambiguation; most terms, conversely, are not relevant because highly polysemic (typically verbs).

The *One Sense per Discourse* (OSD) hypothesis refers to the trend of terms in one discourse of having always the same sense. Conversely, the *One Domain per Discourse* (ODD) hypothesis assumes that term occurrences in a consistent portion of text tend to show the same domain. To defeat such an assumption, Krovetz [7] states that is sufficient that even one term in the same text does not fulfill it. While OSD seems not to hold, ODD does: in a text only a limited number of domains exists.

According to these principles, an extension of WordNet has been set up, called *WordNet Domains (WND)* [8, 9], that associates each synset to the corresponding domain(s), where a domain is intended as a set of words among which strong semantic relations exist. The domains are taken from a domain hierarchy made up of nearly 200 elements and inspired to the *Dewey Decimal Classification (DDE)* system

[10] used by librarians to categorize books. After manually setting some high-level synsets, the relationships already present in WordNet were exploited to automatically complete such an assignment for all synsets. The ODD hypothesis supports the use of WordNet Domain in text disambiguation, to identify the main domain in a text.

2.2 Density Function and Its Exploitation

In the following we recall the WordNet-based density computation according to the *density function* presented in [11]. Terms not included in WordNet (frequent words such as articles, pronouns, conjunctions and prepositions) are not evaluated for classification, this way implicitly performing a stop-word removal.

Given the set $W = \{t_1,...,t_n\}$ of terms in a sentence, each having a set of associated synsets $S(t_i)$, a generic synset s will have weights

- $p(S(t_i), s) = 1/ |S(t_i)|$ if $s_k \in S(t_i)$, 0 otherwise, in $S(t_i)$, and
- $p(W,s) = \Sigma_{i=1,..,n}\, p(S(t_i) ,s) / |W|$ in sentence W.

If a term t is not present in WordNet, $S(t)$ is empty and t will not contribute to computation of $|W|$. The weight of a synset associated to a single term t_i is $1 / (|W| \bullet | S(t_i)|)$. The normalized weight for a sentence is equal to 1.

Given a document D made up of m sentences W_i , each with associated weight $w_i >$ 0, is $p(D, W_i) = w_i / (\Sigma_{k=1,...,h}\, w_k)$. The total weight for a document, given by the sum of weights of all its sentences, is equal to 1. Thus, the weight of a synset s in a document can be defined as:

$$p(D, s) = \Sigma_{j=1,...,m}\, p(W_j, s) \bullet p(D, W_j)$$

In order to assign a document to a category, the weights of the synsets in the document that refer to the same WordNet Domains category are summed, and the category with highest score is chosen. This Text Categorization technique, differently from traditional ones, represents a static classifier that does not need a training phase, and takes the categories from WordNet Domains.

For a successful exploitation of this technique, internal cohesion of the document is very important. Indeed, each sentence in the document conveys a portion of the information it refers to, and authors tend to limit redundancy in the whole text by means of cross-references among sentences. Thus, documents in which each sentence concerns a different topic will probably yield a uniform distribution of scores for the different categories, and prevent the identification of a unique dominant category to be assigned. However, this is not to be considered as a fault of the technique, rather as a case of problematic input in itself.

The technique proposed in [11] exploits information about the PoS of the terms to filter the relevant synsets only and thus improve weight computation. Since wrong assignment of the PoS tag to a term could negatively affect the weight computation (e.g., *marine* as a noun would denote the *military* domain, while as an adjective would indicate *biology*), we want to check the effectiveness of the technique without such knowledge.

3 Document and Text Processing in DOMINUS

DOMINUS (DOcument Management INtelligent Universal System) [12] is a framework designed to intensively exploit advanced Artificial Intelligence and Machine Learning techniques in automatic document processing. It covers all aspects and functionality involved in a digital library, from document acquisition to information retrieval, and particularly focus on the semantic aspects of the information it handles. A document submitted to the system goes through a number of steps that progressively acquire higher-level information, and specifically:

1. *acquisition*: documents in different formats are acquired and translated into a unique representation
2. *layout analysis*: the various components of the document pages are extracted and organized in a structure called layout hierarchy
3. *document image understanding*: the document is assigned to a layout class, and each component in it is associated to a label expressing its role
4. *text analysis*: text from components playing a relevant role is extracted, along with its grammatical and logical structure, and stored for future retrieval
5. *text categorization*: the document is assigned to a category expressing its domain of interest
6. *information extraction*: further information of interest for the specific domain is extracted.

Various steps include intelligent techniques that can be trained and later exploited automatically on new documents. A quality threshold is specified for each step, so that when the accuracy falls below such a threshold the system requires user confirmation before proceeding to the next step; in turn, the user intervention is exploited to improve performance on future cases and take the system above the threshold again.

Here, we focus on the last two steps, that are based on the outcome of step 4 that turns the text in a form that can be processed in a more easy and efficient way.

Specifically, after a *tokenization step* that aims at splitting the text into homogeneous components such as words, values, dates, nouns and a *language identification step,* DOMINUS also carries out additional steps that are language-dependent: PoS-tagging (each word is assigned to the grammatical role it plays in the text by means of a rule-based approach), Stopword removal (less frequent or uniformly frequent items in the text, such as articles, prepositions, conjunctions, pronouns, etc, are ignored to improve effectiveness and efficiency), Stemming (all forms of the same term are reported to a standardized form, this way reducing the amount of elements and highlighting word correspondences), Syntactic Analysis (yielding the grammatical structure of the sentences in the text) and Logical Analysis (providing the role of the various grammatical component in the sentences). For Italian, a rule-based procedure performs in a single step PoS tagging, stopword removal and stemming. For English, different modules are planned to perform these steps, of which only stemming is implemented, according to Porter's algorithm [13]. Hence, we wanted to check whether the qualitative technique described in Section 2 can work effectively even based on stemmed words only.

DOMINUS is very suitable to include a technique based on the density function presented above for various reasons. First of all, Step 3 of document processing can provide the kinds of components to be weighted differently by the density function. Second, the density function deals separately with each document, and hence fits the incremental behavior of DOMINUS more than statistical techniques that require statistics computed on the whole set of documents. Moreover, logical analysis of the sentence can provide phraseology to be considered as a whole instead of the single terms that make it up (although this is left to future work). Lastly, the application of the density function to text categorization and keyword extraction would cover two services that are currently needed in DOMINUS: indeed, the former is still missing, while the latter is currently carried out with a naïve Bayes technique [14] that could be complementary to the semantically-based approach.

The naïve Bayes technique is a quantitative method based on the concepts of frequency and position of a term and on the independence of such concepts. Indeed, a term is a possible keyword candidate if the frequency of the term is high both in the document and in the collection. Furthermore, the position of a term (both in the whole document and in a specific sentence or section) is an interesting feature to consider, since a keyword is usually positioned at the beginning/end of the text. Such features are combined according to the Bayesian Theorem in a formula to calculate the probability of a term to be a keyword in the following way:

$$P(key|T,D,PT,PS) = \frac{P(T|key) * \sum_{i=1}^{|insD|} P\left(D_i|key\right) * \sum_{j=1}^{|insT|} P\left(PT_j|key\right) * \sum_{k=1}^{|insS|} P\left(PS_k|key\right)}{P\left(\sum_{i=1}^{|insD|} D_i + \sum_{j=1}^{|insT|} PT_j + \sum_{k=1}^{|insS|} PS_k\right)}$$

where P(key) represents the probability *a priori* that a term is a keyword (the same for each term), P(T | key) is the standard *tf-idf* value of the term, P(D | key), respectively P(PT | key) and P(PS | key), are computed by dividing the distance of the first occurrence of the term from the beginning of the section (D), document (PT), sentence (PS) with the number of the terms in the section, respectively document and sentence. Finally, P(D, PT, PS) is computed by adding the distances of the first occurrence of the term from the beginning of the section, document and sentence. Since a term could occur in more than one document, section or sentence, the sum of the values are considered. In this way, the probability for the candidate keyword are calculated and the first k (k=10) with the highest probability are considered as the final keywords for the document.

On the other hand, in the qualitative method based on the density function computation, for each text block identified by DOMINUS in the document, the weight associated to the corresponding label and the terms extracted from it are exploited to obtain the *domain categories* to which the document content belongs and the set of *keywords* for the document. The extracted keywords can be exploited to support IR tasks, by computing query results according to keyword matching rather than complete text matching. For instance, given a document with keywords:

{jellyfish, invertebrate, marine, tentacle, cnidaria, ocean}

and that a user query contains the following terms:

{jellyfish, marine, invertebrate, medusa}

the overlapping between the query terms and the document keywords is 3 out of 4, which probably indicates the relevance of the latter to the former.

Concerning the exploitation of knowledge about the document logical structure, as provided by DOMINUS, in the density function computation, different weights will be assigned to the kinds of text blocks as a whole, depending on the role they play in the document, rather than to the single sentences. For instance, in the case of scientific papers the following labels could be considered of interest, along with sensible weights that express their relative importance: TITLE importance 4 - ABSTRACT importance 3 - BIBLIOGRAPHY importance 2 - BODY importance 1

The document weighting algorithm, after computing the density function, proceeds as follows:

1. sort the list of synsets in the document by decreasing weight;
2. assign to the document the first *k* (*k* = 10) terms in the document referred to the synsets with highest weight in the list, whose domain category is not “factotum”;
3. for each pair synset-weight create the pair label-weight where label is the one that WordNet Domains assignes to that synset
4. sort by decreasing weight the pairs label-weight;
5. select the first *n* domain labels that are above a given quality threshold.

Terms with category different from “factotum” are considered, since these have not a peculiar meaning or are frequently used (this improves the stopword removal process).

After assigning weights to all synsets expressed by the document under processing, the synsets with highest ranking can be selected, and the corresponding terms can be extracted from the document as best representatives of its content.

By exploiting WordNet to retrieve the synsets associated to words in the text, the density function identifies the most significant words that express the document subject. Each synset is associated to a weight from the density function, and the weight of the involved synsets is exploited to assign a weight to the overall categories and keywords for the document. Based on WordNet, the new functionality will extract information from documents based not only on the frequency of occurrence of the words they contain, but also on the possible concepts underlying those words.

4 Experiments

The proposed technique was implemented in Java 6.0, embedded in the document processing system DOMINUS and evaluated, for the task of Keyword Extraction and Text Categorization, according to the behavior, effectiveness and efficiency. All experiments were run on a PC endowed with an Intel Core 2 Duo T7200 2.0 Ghz processor and 1 GB RAM, working under Windows XP Professional.

For evaluation purposes, we built a small dataset made up of 5 documents, concerning very different domains and subjects, as specified below:

1. electronic computers, their birth and evolution;

2. child education from birth to adolescence;
3. jellyfish, their habitat and their main features;
4. rubber and its chemical feature;
5. rugby, with references to its rules and history.

For each document, the following figures report the extracted keywords and the corresponding weight computed through the density function. The best ranking keywords are reported, and currently only those having weight greater or equal to 0.03 are selected.

Document		1	2	3	4	5
Length		1317	1167	607	176	834
Runtime KE		13.2"	12.1"	8.4"	5.5"	10"
Runtime TC		2'17"	2'05"	1'35"	37"	1'57"
	1	computer 0.04834439	teaching 0.021461325	jellyfish 0,03463425	synthesized 0.05172414	rugby 0.046938974
	2	electromechanical 0.011299435	education 0.021461325	tentacle 0,03429106	rubber 0.024820872	team 0.013085229
	3	Neumann 0.010788882	instruction 0.021461325	creatures 0,008853416	Hevea 0.023824453	goal 0.011058885
	4	instructions 0.006961259	pedagogy 0.021461325	animal 0,008853416	Kuhn 0.020689657	minutes 0.008610792
	5	transistor 0.004842615	training 0.016265217	zooplankton 0,008333334	latex 0.011050157	football 0.006467662
	6	store 0.00454148	instructor 0.009772334	digestive 0,0074955914	colloidal 0.0103448285	score 0.0063810167
	7	wartime 0.004237288	teachers 0.009772334	invertebrates 0,0069444445	entropy 0.0103448285	hit 0.0063810167

By comparing the keywords extracted to the documents content, a significant overlapping is evident, suggesting high effectiveness of the technique. Indeed, it can identify important terms even when they are specific to the domain discussed in the document (e.g., Cnidaria in Document 3, indicating a class of jellyfish). The first keywords in the ranking are sufficient to identify the domain, while the others are useful to complete and refine the idea about the document content. Runtime, as expected, increases along with the document size, and is quite fast.

As to Text Categorization, the following table reports the identified categories and the corresponding weight computed through the density function.

Doc	1	2	3	4	5
	computerscience 0,074179690	pedagogy 0,13590841	animals 0,145654480	chemistry 0,076343600	play 0,125770640

	mathematics 0,041016333	school 0,056096878	biology 0,134478210	pure_science 0,051724140	sports 0,077846320
	mechanics 0,026428163	politics 0,033727642	anatomy 0,075752420	animals 0,044569080	rugby 0,046938974
	time_period 0,026385480	sociology 0,029268516	gastronomy 0,024342252	biology 0,044454810	person 0,030388908
	person 0,025812928	administration 0,025442114	person 0,023609525	plants 0,036958255	music 0,029128496
	industry 0,024380295	number 0,024071350	chemistry 0,022065999	physics 0,034578360	music 0,021006696
	geography 0,015836516	person 0,022174576	geography 0,017274980	industry 0,033254784	animals 0,018921590
	publishing 0,015047456	university 0,020350550	food 0,014234459	fashion 0,029777769	biology 0,016300263
	buildings 0,014745613	geography 0,017418027	military 0,011330307	sexuality 0,024820872	telecommunic 0,013834923
	art 0,012498402	biology 0,015105671	economy 0,008133663	music 0,017505657	football 0,012883367

Looking at the results, it is evident that the system always succeeds in catching the proper domain category the document deals with. However, differently from the keyword extraction task, the ranking of short documents is shorter, and thus errors in assigning categories are possible due to closely weighted domains (e.g., in document 4). The threshold for assigning a domain to a document was empirically set to 0.03. A lower threshold would tend to provide a single-label classification, whereas a higher threshold would include wrong categories as well. With such a bias, Document 3 would be assigned to categories *animals* and *biology*, while Document 4 would be assigned to categories *chemistry* and *pure_science*.

Runtime is again proportional to the document length, but, in this case, as expected, is much slower. However, this task is carried out only once for each document separately, and the prototype has not been optimized. The main cause of such a behavior is the choice to scan the whole synset map in order to extract the domain categories and rank them by importance. Indeed, such a map is quite large. If the system is to be exploited for web applications/purposes, however, such runtimes are not acceptable, and some alternative way must be found to compute the ranking. A solution can be scanning only a portion of the whole map, in which case such a proportion must be properly defined in order to obtain results similar to the complete case, with a minimal quality decay. Empirical tests showed that using only 1/20th of the whole map yields imprecise and misleading results, particularly for document 5 that is very short. Thus, various experiments led us to empirically set the threshold to 1/10th, so that the results are similar to the original ones also for short documents. Indeed, the results for a short and a long document are reported in the following table, to highlight the different behavior in the two cases.

Doc	1		4	
	whole synset map	***1/10th synset map***	***whole synset map***	***1/10th synset map***
	computer_science 0,074179690	computer_science 0,067159660	chemistry 0,076343600	chemistry 0,068965520
	mathematics 0,041016333	mathematics 0,028364072	pure_science 0,051724140	pure_science 0,051724140
	mechanics 0,026428163	mechanics 0,024066502	animals 0,044569080	animals 0,041827142
	time_period 0,026385480	industry 0,019222680	biology 0,044454810	biology 0,029852908
	person 0,025812928	time_period 0,015588739	plants 0,036958255	plants 0,029780567
	industry 0,024380295	person 0,011930388	physics 0,034578360	fashion 0,024820872
	geography 0,015836516	electricity 0,010278916	industry 0,033254784	industry 0,024820872
	publishing 0,015047456	geography 0,010015408	fashion 0,029777769	sexuality 0,024820872
	buildings 0,014745613	electronics 0,009804817	sexuality 0,024820872	physics 0,022642877
	art 0,012498402	buildings 0,009618105	music 0,017505657	music 0,012056910

Document 1 took 13 sec to accomplish the task and yield the correct category *computer_science*. Document 4 took 8 sec to yield the results, assigning the document to *chemistry* and *pure_science* as in the previous experiment (additionally, category *animals* was introduced, due to references to the animal and vegetal reigns that get a higher importance in a short document).

In large documents, the dominant category becomes more neatly separate from the others, since only more important synset are considered for defining the ranking. Runtimes are neatly reduced, as desired.

A further experiment was carried out that aimed at evaluating both the qualitative and quantitative approaches embedded in DOMINUS, and in order to better test and understand the way the two approaches can be integrate. Thus, we built a dataset made up of 100 documents equally distributed on the following 10 categories: Architecture, Astronomy, Biology, Chemistry, Computers, Economics, Geography, Law, Oceanography-Meteorology, Religion. All the documents were downloaded from the web and in particular from some university libraries.

On each document (only the first page of the documents was used) both techniques were applied requiring the extraction of 10 keywords for each of the methods. Among the 1000 keywords that were extracted from the 100 documents, 518 are the same for both the techniques on 99 documents, i.e. an average of 5.23 keywords in common were extracted (in one case no common keywords were extracted). However, we noted that in many cases, the keywords that are not in common can be used to complete the set of keywords identified by one of the two techniques. Specifically, the quantitative method, that is based on the naïve bayes technique, completes the document description with keywords that describe the topic of the document at a quite general level. On the other hand, a more detailed and specialized description of the document content can be obtained exploiting the quantitative method, that is based on the density function computation. For example, for a document on Oceanography-Meteorology topic, the quantitative method found spring, pacific, eastern, season that are more general than the keywords extracted form the quantitative method, i.e. ship, measurement, compare, maintenance.

Thus, an integration of the two methods is able to better define and identify the document content by mixing the generalized and specialized topic description that each technique is able to grasp. As regards scalability, for the qualitative method this is not an issue since in that case the keywords are computed for each single document with respect to WordNet alone, independently of all the others. For the quantitative method, based on *tf*idf* and hence on the entire collection of documents, since DOMINUS stores in a relational database statistics concerning all *tf* and *idf* values of the terms, it is sufficient to update them incrementally, when a new document arrives, for the terms appearing in that document alone. We plan to perform extensive experiments regarding this problem by using larger collections of documents.

5 Conclusions

This work has presented an extension of the functionality of DOMINUS, a prototypical system for intelligent document processing based on Artificial Intelligence techniques, with qualitative approaches to text processing, based on the semantics of terms rather than on their number of occurrences only. Specifically, two lexical resources (WordNet and WordNet Domains) and a particular density function defined on them have been exploited to transform a document into a weighted map of synsets that describe it conceptually, thus supporting qualitative techniques for text categorization and keyword extraction. DOMINUS is particularly suited to such a technique since it can provide the role each piece of text plays in a document, can process each document separately in an incremental way and already provides quantitative techniques for keyword extraction that can be complemented by the new approach. Experimental results on both tasks are satisfactory, both for accuracy and for effectiveness.

Future works will concern defining a strategy for the selection of keywords and categories when more than one are required (also taking into account generalization relationships among them), and exploiting the extracted information to improve information retrieval and to build or refine specific domain taxonomies.

References

[1] T. De Mauro. Il dizionario della lingua italiana. www.demauroparavia.it

[2] F. Sebastiani (2002) "Machine Learning in Automated Text Categorization" ACM Computing Surveys, Vol.34 N.1, pp. 1-4. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.17.6513

[3] L. Hunyadi (2001) "Keyword extraction: aims and ways today and tomorrow". Lajos Kossuth University. Hungary, pp.1-6. www.keyword.kcl.ac.uk/redist/pdf/hunyadi.pdf

[4] T. Gruber (1995) "Toward Principles for the Design of Ontologies Used for Knowledge Sharing". International Journal of Human-Computer Studies,Vol. 43, No. 5-6. pp. 907-928.

[5] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, K. Miller (1990) "Introduction to WordNet: An On-line Lexical Database". International Journal of Lexicography, Vol. 3, No. 4, pp. 235-244. wordnet.princeton.edu/5papers.pdf

[6] C. Fellbaum (1998) "WordNet an Electronic Database", Cambridge: MIT Press. pp. 1-23.

[7] R.Krovetz. (1998) "More than one sense per discourse". Technical report, Princeton, NEC Research Institute. Proceedings of SENSEVAL Workshop, Herstmonceux Castle, UK, pp.1-10. citeseer.ist.psu.edu/185217.html

[8] B.Magnini, C.Strapparava, G. Pezzulo, A. Gliozzo (2002) "The role of domain Information in Word Sense Disambiguation". Natural Language Engineering, Vol. 8, No. 4 , pp. 359-373. www.istc.cnr.it/doc/1a_16p_Magnini-NLE-2002.pdf

[9] B. Magnini, G. Cavaglià (2000) "Integrating Subject Field Codes into WordNet". ITC-irst, Proc. Second International Conf. Language Resources and Evaluation, LREC2000, pp.1-6.

[10] (2006) "Dewey Decimal Classification" http://www.library.und.edu/research/handouts/027_Dewey_Decimal_Classification.pdf

[11] M. Angioni, R. Demontis, F. Tuveri (2008) "A Semantic Approach for Resource Cataloguing and Query Resolution", Communications of SIWN, ISSN 1757-4439, Vol.5, pp. 62-66.

[12] F. Esposito, S. Ferilli, T.M.A. Basile, N. Di Mauro (2008) "Machine Learning for Digital Document Processing: From Layout Analysis To Metadata Extraction" - Machine Learning in Document Analysis and Recognition 2008, pp. 105-138.

[13] M.F. Porter (1980), "An algorithm for suffix stripping". Program, vol.14, N.3, pp. 130−137. http://tartarus.org/~martin/PorterStemmer/def.txt

[14] Y. Uzun, "Keyword Extraction Using Naïve Bayes", Bilkent
University, Department of Computer Science, Turkey www.cs.bilkent.edu.tr/~guvenir/courses/CS550/ Workshop/Yasin_Uzun.pdf