An intelligent search agent system for semantic information retrieval on the internet

Conference Paper · November 2003

DOI: 10.1145/956699.956725 · Source: DBLP

3 authors, including:

A. D'Acierno
Italian National Research Council
102 PUBLICATIONS **1,259** CITATIONS

SEE PROFILE

Antonio Picariello
University of Naples Federico II
237 PUBLICATIONS **3,985** CITATIONS

SEE PROFILE

An Intelligent Search Agent System for Semantic Information Retrieval on the Internet

Carmine Cesarano
DIS, via Claudio 21
80125 - Napoli, ITALY
carmine.cesarano@unina.it

Antonio d'Acierno
ISA - CNR, via Roma 52
83100 - Avellino, ITALY
dacierno.a@isa.cnr.it

Antonio Picariello
DIS, via Claudio 21
80125 - Napoli, ITALY
antonio.picariello@unina.it

ABSTRACT

In this paper we describe a prototype system for information retrieval on the Internet. Our idea is that the Web has to be searched both *semantically* and *syntactically*. In order to automatically categorize the web pages *on the fly* we propose a novel approach based on ontology and semantic networks and we describe a prototype system based on the Intelligent Agent Paradigm. Preliminary experiments are shown and discussed while describing open problems and on-going research.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search Retrieval; H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Algorithms

Keywords

Information Retrieval, Semantic Network, Web Agents, Ontology

1. INTRODUCTION

The term "search engine" is used to indicate both crawler-based search engines and manually maintained directories, although they gather their indexes in radically different ways.

Crawler-based search engines, such as *Google*, create their catalogues automatically: they *crawl* the web, then the users searches through what they have found.

On the contrary, a manually maintained directory, such as the *Open Directory*, depends on humans: people submits a short description to the system about a certain site, or appropriate editors write a review for their assigned sites; thus, a web search looks for matches only in the submitted descriptions.

WIDM'03, November 7–8, 2003, New Orleans, Louisiana, USA.

In the web's early days, a search engine usually presented either crawler-based results *or* human maintained catalogues. Today, it is extremely common that both types of results are presented. Usually, a *hybrid* search engine will favor one type of listings over another. For example, *MSNSearch* is more likely to present human maintained directory then *LookSmart*. However, it also presents crawler-based results (as provided by Inktomi), especially for more complex and obscure queries.

From a general web-surfer point of view, clearly, a human maintained index allows to quickly and accurately search the web; in addition, the result of the queries are semantically organized. On the other hand, i) the catalogue service sometimes is not free and, as a consequence of that, some important sites are not included and indexed; and ii) the directories are not always up to date both for the enormous growth of information sources available on the Internet and for the inherently time varying nature of the web pages.

These problems require the availability of tools able to *automatically* categorize the web data; such tools, usually called *web mining tools*, are broadly classified in

- *web usage tools*, usually implementing *the process of customizing the content and the structure of web sites* in order to satisfy the specific need of each and every user, without asking for it explicitly;
- *web content mining tools* used for automatic classification of document contents, including their multimedia objects as well as textual information.

In this paper we describe a system for *semantic* web content mining; our system works *on-the-fly*, i.e. web pages are categorized as the web is searched so deriving an inherently up to date listing.

The paper is organized as follows: several introductory examples are given in section 2, in order to describe the research problems and the several solutions proposed in the literature. Section 3 contains a detailed description of the proposed information retrieval system, showing the several component agents. The designed semantic knowledge base and the mining process are given in sections 4 and 5, respectively. We will also report on a prototype implementation and describe some experimental results together with a deep analysis of open problems and possible solutions to be realized (section 6 and 7).

2. RELATED WORKS

To best understand our work, let us consider the following example. Current search engines, (e.g. *Google, Altavista, Lycos* and so on), of course return a number of relevant pages and also a lot of not useful links. Let us consider the task of retrieving all the information about a common musical instrument from the internet: a *guitar*. If one uses the italian word *chitarra*, a lot of pages related to the special italian pasta dish *spaghetti alla chitarra* are also retrieved!

To avoid this boring problem, it could be the case for the system to know the semantic context related to a user query. In this framework we suggest to enhance the search engines by retrieving the semantic content of a page and by comparing it to the one specified by the user itself. By the end of this paper, we would have described a possible approach to the extraction and to the comparison of this semantic information.

Several projects dealing with content-based and semantic discovery have been developed. The SAFARI project [7], for example, focuses on developing and integrating techniques for content-based mining of image archives and semantic information source matching in an agent-based foundation.

Similarity-based search are discussed in various paper related to web and text mining, as in [10], [4] and [8].

In [3], an interesting approach and a prototype system called *AKIRA* are proposed, providing an access to the explicit and implicit structure of web documents as well as an organization of the retrieved information into *concepts* and *meta-concepts*.

The integration of agent technology and ontology is proposed in [9] as a significative impact on the effective use of the web services.

Machine learning techniques have been widely used for automatically extracting semantic information and, thus, for text categorization; for a review on such techniques see [6].

However, to the best of our knowledge, there has been only some significant attempt to merge information retrieval and ontological models namely that by Missikoff *et al.* [2], who proposed a text processing system for building ontological domain. Building upon their influential work, we make the following contribution:

- we have used an agent based approach for semantic web searching and
- we have enriched the ontological concepts with a probabilistic notions in order to manage the related uncertainty.

3. SYSTEM ARCHITECTURE

Figure 1 presents an high level view of the proposed architecture for a semantic information retrieval system based on intelligent agents.

The user, by means of a graphical interface, submits a query to the system; apart from keywords and sentences to be found as a whole, she/he specifies a numeric value, that indicates the *Depth* at which each site has to be inspected, together with the *Language* of pages to be found and the *Context* that indicates the search area.

The user query is translated and/or adapted to be submitted to many standard search engines by means of the first component of the Search Engine Wrapper (*SEW*).

Standard search engines are then queried and the results are collected and parsed to obtain a list of *links*.

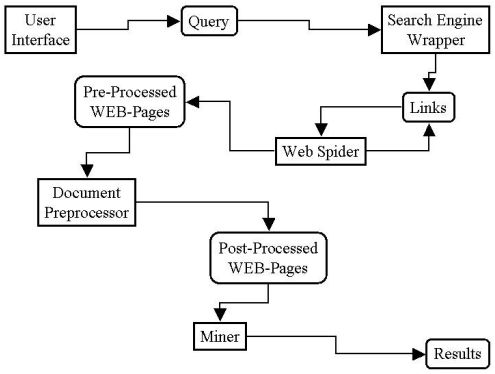

Figure 1: High level system architecture.

The Web Spider (*WS*) first of all downloads the found pages. Then, for each page, it inspects those links which refer to pages on the same site to find pages (if any) that, for some reason, are not found by standard engines. This process ends when the user-defined *depth* is reached.

Downloaded pages are processed by a *Document preprocessor* that extracts useful information within a single document. Eventually, the *Miner Agent* uses a Semantic Knowledge Base (*SKB*) to grade the retrieved pages according to their semantic similarity referred to the user defined context.

In the following subsection we will briefly describe the SEW, the WSA and the *Document Preprocessor*.

3.1 The Search Engine Wrapper

The SEW (figure 2) accepts the user query that is adapted to the syntax of the various search engines by means of the *Query Adapter* agent. The obtained queries (one for each used engine) are then submitted by the *Submitter* and the collected results are parsed in order to extract and merge the obtained collections of links (one for each engine).

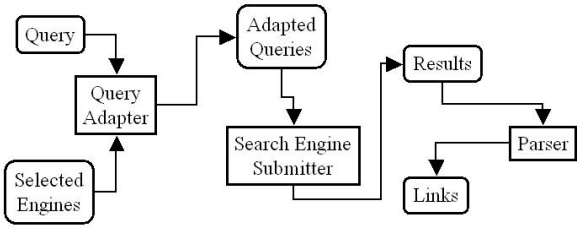

Figure 2: The Search Engine Wrapper.

It is worth noting that the user has the possibility of both selecting the engines that must be used and defining new ones.

3.2 The Web Spider Agent

The WSA has the responsibility of inspecting the found pages and of storing them in a local repository (figure 3).

It uses the *Web Catcher* agent that downloads the page; each page is stored in a local *Web Repository* and parsed by a *Web Page Parser*. This parser extracts - from the generic page - those links which refer to the pages relying to the same logical web site; these links, if they not already belong to the set of links under investigation, are stored in the links repository. This steps is clearly useful to find pages that, for

some reason, aren't found by standard engines. The iterative process ends when the user-defined depth is reached.

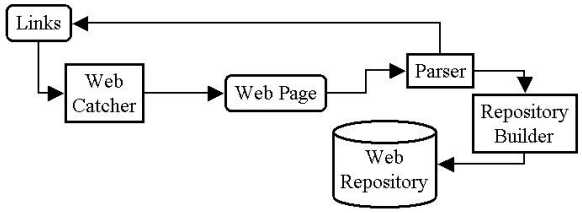

Figure 3: The Web Spider Agent.

The structure of the Web Repository (see figure 4) is roughly divided into two areas.

The *PREPROCESSED* entity contains the URL that identifies the page, the downloaded page (the whole HTML content) and its depth; the depth is assumed to be 0 for the pages which are directly found by at least one standard engine. We assume that the related engine for a certain page which is found only by the WSA, (the mandatory association *FOUND*), is the engine that has returned the link from which that page has been retrieved.

The *POSTPROCESSED* entity contains the HTML content of each page which has been stored by the Document Preprocessor (described in the following), while the *GRADE* entity contains the results of the Miner agent (see section 5).

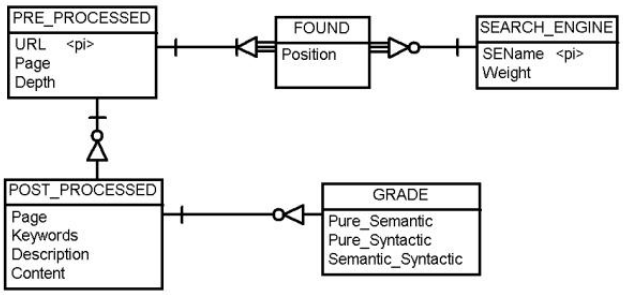

Figure 4: A simplified ER schema for the Web Repository.

3.3 The Document Preprocessor

Using HTML tags and meta-tags, the Document Preprocessor (figure 5) extracts, for each page, the title, the content, the description (if any) and the keywords (if any).

This service, moreover, uses a language dependent stopwords list to eliminate useless words; in our prototypal implementation of the system we use Italian.

Now, we have to concentrate our attention on the grading process performed by the miner; for this reason, we briefly describe how the semantic of a given domain can be quantitatively described by means of a semantic network.

4. SKB BASIC CONCEPTS

Ontological semantic is a theory which uses an *ontology* for extracting and representing the *meaning* of natural language texts and for reasoning about the knowledge derived

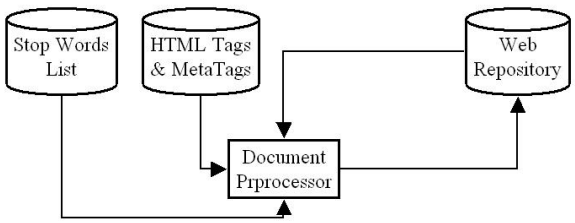

Figure 5: The Document preprocessor.

from such texts [5]. Following the suggestions of [1] and of [2], we use "*concepts*" for building a semantic network in order to provide a useful framework for discovering information from the net.

Despite the number of different and sometimes controversial definition of what an *ontology* is, for the purposes of this work we adopt the notion given in [11]: an ontology is a formalization that represents a *domain of objects* and their *relations*. An *ontological concept* is characterized by a *term*, denoting the concept itself, by a *description*, explaining the meaning of a concept, and by a set of *relationships* with the other concepts. In this context, a *Semantic Network* is a set of annotated *concepts* and *links*.

In other words, a Semantic Network for a knowledge domain D can be described as a directed weighted graph whose nodes represent concepts and whose edges represent relationships (generalization, similarity, etc.) between two concepts c_A and c_B. The associated weight to every edge represents the strength of the semantic relation between the concept c_A and c_B. For the sake of completeness, if is worth noting that each concept is related to itself with grade 1.

Figure 6 shows a semantic network representing a simple (and incomplete) ontology about the domain *music*[1]; self-relations are not reported.

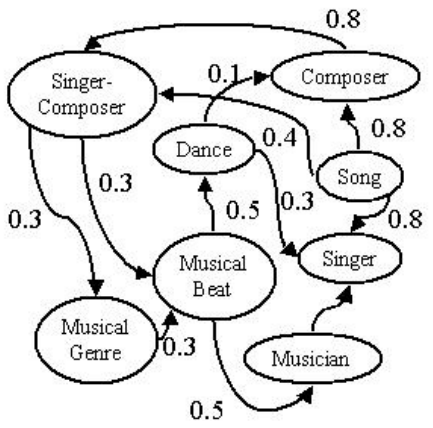

Figure 6: A simple (and incomplete) semantic network representing an ontology about the domain *music*.

Given n concepts, $(c_1, \ldots, c_n)$, the *semantic similarity*

[1]The semantic network actually used in our experiments is omitted both for the sake of simplicity and since it is in Italian.

between two concepts may be expressed in terms of a *probabilistic distance* function:

$$d(c_a, c_b) = \max_{i \in (1,..,n)} (\prod_{j=1}^{m_i} P_{ij}) \quad (1)$$

n being the number of existing paths from c_a to c_b, m_i the number of edges of the discovered path and P_{ij} the weight which is related to each edge.

Using 1 it's easy to prove that d satisfies the following relations:

1. $d(c_a, c_b) \neq d(c_b, c_a)$;
2. $d(c_a, c_b) = 0$ *iff* there not exists any path from c_a to c_b, i.e. the two concepts have no relations.
3. $d(c_a, c_c) \neq d(c_a, c_b) \times d(c_b, c_c)$.

Figure 7 shows the core part of the *ER* schema representing our SKB. The *WORD*, expresses a *CONCEPT* in a *DOMAIN* (that is, it has a *MEANING*).

Each relationship expressing a meaning is *weighted*, in order to quantify the relevance of that word and of that concept to the domain.

The Concepts are related to each other (see figure 6); each edge in the semantic network is represented by a relationship between two concepts. The relationship has an attribute measuring the distance among concepts; clearly each concept can be related to many other concepts, while two concepts can have at most one edge connecting them.

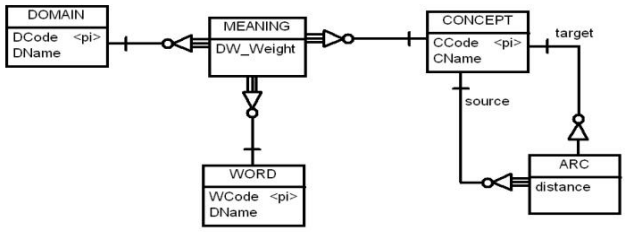

Figure 7: The ER model for the SKB.

5. THE MINER

The Miner Agent represents the main agent of the whole system having the responsibility of grading the retrieved pages according to their relevance to the semantically specified user query; i.e., the Miner has to *Categorize* a document.

Document (or text) categorization is the task of assigning a value to each pair $(d_j, c_i) \in D \times C$, where D is a domain of documents and C is a set of categories.

Document categorization can be *single-label* (exactly one category has to be assigned to a document) as well as *multil-abel* (in which case any number of categories can be assigned to a document). A special case of single-label categorization is the *binary* one, where there exist just two categories (say c and its complement $\bar{c}$).

Document categorization, moreover, can be *ranked*, when the system "simply" ranks a document according to its estimated appropriateness with each category, without taking any decision about best representing category (*hard* categorization).

Our miner is a ranking agent (since it expresses a grade for each page) and a binary classifier - there are just two categories: the user selected domain and its complement. The proposed ranking system uses both syntactic and semantic criteria (see figure 8).

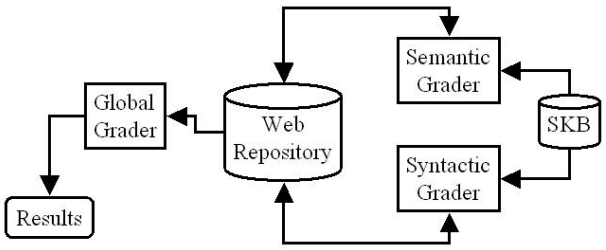

Figure 8: The architecture of the Miner.

For each occurrence of the POSTPROCESSED entity in the web repository, first we have a *Syntactic Grader* that (see figure 8) evaluates the syntactic relevance of the page (section 5.1). Concurrently, a *S*emantic Grader evaluates the semantic appropriateness according both to the presence in the page of words belonging to the user-selected domain (section 5.2) and to the used ontology (section 5.3).

5.1 Syntactic Grade

It is worth noting that search engines return several links according to some internal order (randomly in the extreme case). It is however a matter of facts that, when we search the Web, the links that are at the top of the first page are considered the most relevant. For such a reason we decided, first of all, to reward the pages according to their position in the returned lists of hyper links.

Let us consider n search engines; for each engine we use a weight (w_i) such that $\sum w_i = 1$ and $0 \leq w_i \leq 1 \ \forall i$. We moreover suppose, clearly without loss of generality, that:

$$w_1 \leq w_2 \ .... \leq w_n \quad (2)$$

Let $p_i(x)$ be the position of page x in the search engine i, let A be the set of search engines which have found x $(a = card(A))$ and let B be the set of search engines that did not find x.

We define the *Average Position AP* as:

$$AP(x) = \frac{\sum_{i \in A} p_i(x) \cdot (w_i + \overline{w})}{\sum_{i \in A} w_i} \quad (3)$$

where $\overline{w} = \frac{\sum_{j \in B} w_j}{a}$ simply penalizes pages found just by a subset of search engines.

In the best case (each search engine finds the page x in position 1) we have $AP(x) = 1$; $AP(x)$ of course increases (theoretically towards $+\infty$) as *(i)* increases the number and the weight of search engines that does not find x, as *(ii)* increases the number of links returned by each engine and *(iii)* as decreases the position of x in each standard search result.

In our system the number of links L that each engine has to return is user-defined; thus, in the worst case, because of 2, we have that:

$$AP_{min} = \frac{L * w_1 + \sum_{i=2}^{n} w_i}{w_1} \quad (4)$$

The *Syntactic Grade* of the page x ($SyG(x)$) is simply defined as:

$$SyG(x) = \frac{1}{AP(x)} \tag{5}$$

having of course that, $\forall x$, $\frac{1}{AP_{min}} \leq SyG(x) \leq 1$, 1 being the best case.

In the current implementation of the system we simply consider the list of returned links and we assume, as position, the position of the links in such list. The concept of position could be of course refined to consider, say, the engines that, for each link, return a matching score.

A problem to be addressed concerns the case in which a page is found just by WSA; in this paper we use as depth of search 0, that is WSA does not search sites at all. A new formulation of SyG that is well suited also for pages that do not appear in standard search engines result sets, is currently under study (see section 7).

5.2 Semantic-Syntactic Grade

We select for each page x the words that belong to the dictionary representing the user selected domain D. The presence in x of a word that is related to a desired domain, of course, semantically increases the confidence in the fact that x is a useful page, if we are looking in that domain; on the other hand, our system takes into account just words, so ignoring the relation among them; this is why we talk of *semantic-syntactic* confidence and we define a *Semantic-Syntactic Grade (SSG)* as:

$$SSG(x) = \frac{\sum_{i=1}^{N} w_i}{NT} \tag{6}$$

$w_i \in [0,1]$ being the weight of the word i in the dictionary D ($w_i = 0$ if the word is not present in the dictionary), NT being the number of tokens of x that belongs to D and N the number of words in the page.

Clearly we have that $SSG \in [0,1]$; 1 is the best case (each word in x belonging to the dictionary has the weight equal to 1), while 0 is the worse case (there not exists words in x that belongs to the dictionary).

5.3 Semantic Grade

In our ontological approach, the semantic relevance of a page is considered to be a function of concepts, each concept being expressed by a word.

First of all, the semantic grader translates the sequences of words representing content, title, keywords and description of the page x into sequences of concepts using the SKB.

To deal with semantically related concepts, we consider then, for each pair of concepts, the equation 1. To take into account the distance between the concepts, we define a *Normalized Probabilistic Distance* (NPD) as:

$$NPD(c_i, c_j) = \frac{d(c_i, c_j)}{DIST(c_i, c_j)} \tag{7}$$

$DIST(c_i, c_j)$ being the distance between the words representing the concept c_i and the concept c_j. Clearly, when the keywords are considered, $DIST(c_i, c_j)$ is always 1.

In order to semantically grade a page, we consider for each page all the possible pairs of concepts belonging to the ontology and, for each pair, we evaluate 7, so deriving the *Semantic Grade (SeG)*:

$$SeG(x) = \sum_{h=1}^{NC} \sum_{k=h+1}^{NC} NPD(C_{ik}, C_{ih}) \tag{8}$$

NC being the number of concepts that are found in page x; because many words can express the same concept, we have that $NC \leq NT$ (and possibly $NC \ll NT$).

In the worse case (the case in which there are no related words) $SeG = 0$; in the best case, on the other hand, all the NC concepts, in the content of the page are related with distance 1, so deriving:

$$SeG_{max} = \frac{NC^2 + NC}{2} \tag{9}$$

For this reason, we introduced the *Normalized* SeG so defined:

$$NSeG_i(x) = \sum_{h=1}^{NC} \sum_{k=h+1}^{NC} \frac{2 \cdot NPD(C_{ik}, C_{ih})}{NC^2 + NC} \tag{10}$$

Given a page x, equation 10 clearly has to be evaluated for the content ($i = c$), for the title ($i = t$), for the keywords ($i = k$) and for the description ($i = d$).

The *Normalized Semantic Grade* of the page x is evaluated as a linear combination of the $NSeG_i$'s:

$$NSeG(x) = \sum_{i \in \{t,c,k,d\}} \rho_i \cdot NSeG_t(x) \tag{11}$$

where $\rho_t + \rho_c + \rho_k + \rho_d = 1$.

5.4 Global Grade

The *Global Grade* of the page x is evaluated as linear combination of single grades that we have just defined:

$$GG(x) = \frac{k_y SyG(x) + k_{ss} SSG(x) + k_e NSeG(x)}{k_y + k_{ss} + k_e} \tag{12}$$

Here the k_i's represent those constants that allow to decide, for each query, the relevance of each grade in the evaluation of GG.

A performance problem arises considering $NSeG_c(x)$ (equation 10 with $i = c$), since, for the content of the page, we should consider each possible couple of words; when we deal with page with many concepts in its content, the time spent in evaluating such a term becomes prohibitive. As the distance between words increases, on the other hand, even if two words express heavily related concepts, NPD can be easily neglected.

We formalize such a simple concept by introducing the *Approximated* $NSeG_c$ ($\widetilde{NSeG_c}$):

$$\widetilde{NSeG_i}(x) = \sum_{h=1}^{NC} \sum_{k=h+1}^{NC} f\left(\frac{2 \cdot NPD(C_{ik}, C_{ih})}{NC^2 + NC}, T\right) \tag{13}$$

where T is experimentally fixed threshold and $f(x,T)$ is defined such that $f(x,T) = x$ if $x \leq T$, $f(x,T) = 0$ otherwise. Using such an approximation we have, for example, that with $T = 0.1$ we can consider, for each concept, just concepts having as maximum distance 4.

Table 1: Results obtained using some standard search engine with two difficult queries; see text for details.

Query	Google	Altavista	Lycos	T	R
zucchero	6/3	6/4	3/2	48	14
morandi	2/0	2/0	2/1	45	4

Table 2: Quantities used for grading a page.

Weight of *Google*	0.334
Weight of *Altavista*	0.333
Weight of *Lycos*	0.333
ρ_c	0.5
ρ_t	0.3
ρ_d	0.1
ρ_k	0.1
k_y	0.1
k_{ss}	5
k_e	70
T	0.1

We tested our system using equations 12 and 13, so deriving an *Approximated GG* ($\widetilde{GG}$):

$$\widetilde{NSeG}(x) = \rho_c \cdot \widetilde{NSeG_c}(x) + \sum_{i \in \{t,k,d\}} \rho_i \cdot NSeG_t(x) \quad (14)$$

$$\widetilde{GG(x)} = \frac{k_y SyG(x) + k_{ss} SSG(x) + k_e \widetilde{NSeG(x)}}{k_y + k_{ss} + k_e} \quad (15)$$

6. EXPERIMENTAL RESULTS

In this paper we show some results for two complex queries. The first one is the query *zucchero*, i.e. an Italian singer whose name, in English, sounds like "sugar". Because of this, each standard engine reported a lot links concerning *sugar* as a food together with many links concerning the singer *Zucchero* (the actual pages we was looking for).

The second query is the word *morandi*, i.e. the surname of a well known Italian painter and of (at least) two Italian singer; each standard engine so reported links about the painter Morandi, links related to one of the singer Morandi (the pages we was looking for), together with several links referring to other people whose surname is Morandi.

Although the syntactic search engines are not useful for a comparison with a semantic algorithm, we give some results obtained with some standard search engine just for furnishing an idea about the obtained improvements, as shown in Table 1; results are reported in the form x/y: x represents the number of relevant links (first 20 links are considered), y represents the number of relevant links found in the first results page (10 links per page). The column T reports the total number of retrieved links (consider that some links can be found by several engines), while the column R shows the total number of *relevant* retrieved links.

The proposed system has been tested using the parameters shown in Table 2, whose rationale is:

1. Standard search engines used (*Google*, *Altavista* and *Lycos*) are considered, roughly speaking, equivalent.
2. $\rho_k \ll \rho_c$, i.e. keywords are considered significantly less important than the content. This could seems surprising, but keywords are often considered not important, at least when pages are managed by inexperienced people.
3. $\rho_d \ll \rho_c$, i.e. the description is considered significantly less important than the content. Again, this could seems surprising, but description (when present) is often considered not important, at least when pages are managed by inexperienced people.
4. We give some importance to the title ($\rho_t = 0.3$) since it is typically significant, at least for home pages and for high level pages[2].
5. Theoretically, SSG and $NSeG$ are considered to have the same importance in the final grade of a page. However we have $k_e \gg k_{ss}$ since, because of 6 and 14, typically $\widetilde{NSeG}(x) \ll SSG(x)$
6. $k_y \ll k_{ss}$. This term is used just to reward pages with an high average position in the sets returned by standard search engines; our idea is that such a position should be considered not really important (and, in the extreme case, neglected).
7. $T = 0.1$ is just for performance considerations.

Remember that our system is a grading system, i.e. it grades pages according the relevance with the user defined context; in the best case, so, it should return, for each query, an ordered list of T links, the first R of which are *relevant* (see Table 1). Given such a best case, we can define a *False Positive Link (FPL)* as a link that is erroneously considered relevant by the system (i.e. it appears, in the ordered list, in one of the first R positions).

For the query *zucchero* we have just 1 FPL (position 11) while, for the query *morandi*, there are no FPL.

7. CONCLUSIONS AND ON GOING WORK

We are developing a system for semantically searching information on the Internet; the first implementation of the system shows an interesting performance in terms of searching precision. Our system categorizes web pages on the fly, since this is the only way to deal with highly time variant information sources like web pages.

Several problems have to be still addressed. From the implementation point of view, we are testing, for pages found just by WSA, a penalizing factor K_b, thus assuming as $SyG(x)$:

$$SyG(x) = \frac{1}{K_b * AP(x)} \quad (16)$$

A radically different approach that is being tested consists in ignoring $SyG(x)$, i.e. we can decide to use standard search engines just as starting point for our search and completely ignoring the position in which each link is returned.

Another problem is that the semantic network representing the ontology for each context has to be automatically built; machine learning strategies such as neural networks

[2]High level pages simply means pages near the root node of the tree representing the web site.

and graph matching techniques are being implemented and tested.

An interesting issue to be to analyzed concerns the domains that are, in some sense, related. Consider again the domain "music" and, for example, the domain *television*; these domains are clearly logically, and thus semantically, related. When we search, say, in the music domain, does it make sense to grade found pages also in related domains? A possibility could be to organize domains in a sort of lattice, having the system capable of grading pages in the user selected domain, as well as in related domains.

Last, the performance has to be considered. The prototype system runs on a single PC and has been realized using Java and Oracle as DBMS; since the agents are inherently concurrent, our idea is to use a cluster of PC's to speed-up the search process.

8. ACKNOWLEDGEMENTS

This work has been carried out partially under the financial support of the Ministero dell'Istruzione, dell'Universita' e della Ricerca (MIUR) in the framework of the FIRB Project "Middleware for advanced services over large-scale, wired-wireless distributed systems (WEB-MINDS)".

9. REFERENCES

[1] T. Eiter, T. Lukasiewicz, J. J. Lu, and V. S. Subrahmanian. Probabilistic object bases. Technical report, Technical University of Wien, 2001.

[2] P. Fabriani, M. Missikoff, and P. Velardi. Using text processing techniques to automatically enrich a domain ontology. In *FOIS'01*, pages 270–284. ACM, 2001.

[3] Z. Lacorix, A. Sahuguet, R. Chandrasekar. Information Extraction and Database Techniques: A User-Oriented Approach to Querying the Web. In *CAiSE 1998* , pages 289-304. ACM, 19998.

[4] C.-H. Lee and H.-C. Yang. Text mining of bilingual parallel corpora with a measure of semantic similarity. In *Proceedings on IEEE International Conference on Systems, Man, and Cybernetics 2001*, pages 470–475. IEEE, 2001.

[5] S. Niremburg and V. Raskin. Ontological semantics, formal ontology and ambiguity. In *FOIS'01*, pages 151–161. ACM, 2001.

[6] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys (CSUR)*, 34(1):1–47, 2002.

[7] E. Shek, A. Vellaikal, S. Dao, and B. Perry. Semantic agents for content-based discovery in distributed image libraries. In *Proceedings IEEE Workshop on Content-Based Access of Image and Video Libraries*, pages 19 –23. IEEE, 1998.

[8] R. Srihari, A. Rao, B. Han, S. Munirathnam, and X. Wu. A model for multimodal information retrieval. In *2000 IEEE International Conference on Multimedia and Expo (ICME 2000)*, volume 2, pages 701 –704. IEEE, 2000.

[9] L. Weihua. Ontology supported intelligent information agent. In *Proceedings on the First Int. IEEE Symp. on Intelligent Systems*, pages 383–387. IEEE, 2002.

[10] H. Xu, Y. Mita, and T. Shibata. Intelligent internet search applications based on vlsi associative processors. In *Proceedings on Symposium on Applications and the Internet (SAINT2002)*, pages 230–237. IEEE, 2002.

[11] G. L. Zuniga. Ontology: Its transformation from philosophy to information systems. In *FOIS'01*, pages 187–197. ACM, 2001.