10

Retrieving and Categorizing Bioinformatics Publications through a MultiAgent System

Andrea Addis[1], Giuliano Armano[1], Eloisa Vargiu[1] and Andrea Manconi[2]
[1]University of Cagliari - Dept. of Electrical and Electronic Engineering
[2]Institute for Biomedical Technologies, National Research Council
Italy

1. Introduction

The huge and steady increase of available digital documents, together with the corresponding volume of daily updated contents, makes the problem of retrieving and categorizing documents and data a challenging task. To this end, automated content-based document management systems have gained a main role in the field of intelligent information access (Armano et al., 2010).

Web retrieval is highly popular and presents a technical challenge due to the heterogeneity and size of the Web, which is continuously growing (see (Huang, 2000), for a survey). In particular, it becomes more and more difficult for Web users to select contents that meet their interests, especially if contents are frequently updated (e.g., news aggregators, newspapers, scientific digital archives, RSS feeds, and blogs). Supporting users in handling the huge and widespread amount of Web information is becoming a primary issue.

Among other kinds of information, let us concentrate on publications and scientific literature, largely available on the Web for any topic. As for bioinformatics, it can be observed that the steady work of researchers, in conjunction with the advances in technology (e.g., high-throughput technologies), has arisen in a growing amount of known sequences. The information related with these sequences is daily stored in the form of scientific articles. Digital archives like BMC Bioinformatics[1], PubMed Central[2] and other online journals and resources are more and more searched for by bioinformaticians and biologists, with the goal of downloading articles relevant to their scientific interests. However, for researchers, it is still very hard to find out which publications are in fact of interest without an explicit classification of the relevant topics they describe.

Traditional filtering techniques based on keyword search are often inadequate to express what the user is really searching for. This principle is valid also in the field of scientific publications retrieval, where researchers could obtain a great benefit from the adoption of automated tools able to search for publications related with their interests.

To be effective in the task of selecting and suggesting to a user only relevant publications, an automated system should at least be able (i) to extract the required information and (ii) to encode and process it according to a given set of categories. Personalization could also be provided according to user needs and preferences.

[1] http://www.biomedcentral.com/bmcbioinformatics/
[2] http://www.pubmedcentral.gov/

In this chapter, we present PUB.MAS, a multiagent system able to retrieve and categorize bioinformatics publications from selected Web sources. The chapter extends and revises our previous work (Armano et al., 2007). The main extensions consist of a more detailed presentation of the information extraction task, a deep explanation of the adopted hierarchical text categorization technique, and the description of the prototype that has been implemented. Built upon X.MAS (Addis et al., 2008), a generic multiagent architecture aimed at retrieving, filtering and reorganizing information according to user interests, PUB.MAS is able to: (i) extract information from online digital archives; (ii) categorize publications according to a given taxonomy; and (iii) process user's feedback. As for information extraction, PUB.MAS provides specific wrappers able to extract publications from RSS-based Web pages and from Web Services. As for categorization, PUB.MAS performs Progressive Filtering (PF), the effective hierarchical text categorization technique described in (Addis et al., 2010). In its simplest setting, PF decomposes a given rooted taxonomy into pipelines, one for each existing path between the root and each node of the taxonomy, so that each pipeline can be tuned in isolation. To this end, a threshold selection algorithm has been devised, aimed at finding a sub-optimal combination of thresholds for each pipeline. PUB.MAS provides also suitable strategies to allow users to express what they are really interested in and to personalize search results accordingly. Moreover, PUB.MAS provides a straightforward approach to user feedback with the goal of improving the performance of the system depending on user needs and preferences.

The prototype allows users to set the sources from which publications will be extracted and the topics s/he is interested in. As for the digital archives, the user can choose between BMC Bioinformatics and PubMed Central. As for the topics of interest, the user can select one or more categories from the adopted taxonomy, which is taken from the TAMBIS ontology (Baker et al., 1999).

The overall task begins with agents able to handle the selected digital archives, which extract the candidate publications. Then, all agents that embody a classifier trained on the selected topics are involved to perform text categorization. Finally, the system supplies the user with the selected publications through suitable interface agents.

The chapter is organized as follows. First, we give a brief survey of relevant related work on: (i) scientific publication retrieval; (ii) hierarchical text categorization; and (iii) multiagent systems in information retrieval. Subsequently, we concentrate on the task of retrieving and categorizing bioinformatics publications. Then, PUB.MAS is illustrated together with its performances and the implemented prototype. Conclusions end the chapter.

2. Background

Information Retrieval (IR) is the task of representing, storing, organizing, and accessing information items. IR has changed considerably in recent years with the expansion of the World Wide Web and the advent of modern and inexpensive graphical user interfaces and mass storage (Baeza-Yates & Ribeiro-Neto, 1999). The most relevant IR issues that help to clarify the contextual setting of this chapter are: (i) the work done on scientific publication retrieval, (ii) the work done on Hierarchical Text Categorization (HTC), and (iii) the work done on multiagent systems (MAS) for information retrieval.

2.1 Scientific publication retrieval

In the academic area, online search engines are used to find out scientific resources, as journals and conference proceedings. However, finding and selecting appropriate information on the

Web is still difficult. To simplify this process, several frameworks and systems have been developed to retrieve scientific publications from the Web.

Bollacker et al. (2000) developed CiteSeer[3], the well-known automatic generator of digital libraries of scientific literature. Being aimed at eliminating most of the manual effort of finding useful publications on the Web, CiteSeer uses sophisticated acquisition, parsing, and presentation methods. In particular, CiteSeer uses a three-stage process: database creation and feature extraction; personalized filtering of new publications; and personalized adaptation and discovery of interesting research and trends. These functions are interdependent: information filtering affects what is discovered, whereas useful discoveries tune the information filtering. In (McNee et al., 2002), the authors study how to recommend research papers using the citation between papers to create the user-item matrix. In particular, they test the ability of collaborative filtering to recommend citations that could be additional references for a target research paper. Janssen & Popat (2003) developed UpLib, a personal digital library system that consists of a full-text indexed repository accessed through an active agent via a Web interface. UpLib is mainly concerned with the task of collecting personal collections comprising tens of thousands of documents. In (Mahdavi et al., 2009), the authors start from the assumption that trend detection in scientific publication retrieval systems helps scholars to find relevant, new and popular special areas by visualizing the trend of the input topic. To this end, they developed a semi-automatic system based on a semantic approach.

As for the specific task of retrieving information in the field of bioinformatics, a lot of work has been done –some of it being recalled hereafter. Tanabe et al. (1999) developed MedMiner, an Internet-based hypertext program able to filter and organize large amounts of textual and structured information returned from public search engines –like GeneCards and PubMed. Craven & Kumlien (1999) applied machine learning techniques to automatically map information from text sources into structured representations, such as knowledge bases. Friedman et al. (2001) propose GENIES, a system devised to extract information about cellular pathways from the biological literature in accordance with a given domain knowledge. Ramu (2003) developed a Web Server for SIR (Ramu, 2001), a simple indexing and retrieval system that combines sequence motif search with keyword search. The Web Server, called SIRW, is a generic tool used by the bioinformatics community for searching and analyzing biological sequences of interest. Rocco & Critchlow (2003) propose a system aimed at finding classes of bioinformatics data sources and integrating them behind a unified interface. The main goal of the system is to eliminate the human effort required to maintain a repository of information sources. Kiritchenko et al. (2004) propose a system aimed at retrieving Medline articles that mention genes. After being retrieved, articles are categorized according to the Gene Ontology (GO) codes. Delfs et al. (2004) developed and implemented GoPubMed, a system that allows to submit keywords to PubMed, extracts GO terms from the retrieved abstracts, and supplies the user with the relevant ontology for browsing. Corney et al. (2004) propose BioRAT (Biological Research Assistant for Text mining), a new information extraction tool specifically tailored for biomedical tasks. Able to access and analyze both abstracts and full-length papers, it incorporates a domain specific document search ability.

2.2 Hierarchical text categorization

In recent years several researchers have investigated the use of hierarchies for text categorization.

Until the mid-1990s researchers mostly ignored the hierarchical structure of categories that occur in several domains. In 1997, Koller & Sahami (1997) carry out the first proper study

[3] http://citeseer.ist.psu.edu/

on HTC on the Reuters-22173 collection. Documents are classified according to the given hierarchy by filtering them through the single best-matching first-level class and then sending them to the appropriate second level. This approach shows that hierarchical models perform well when a small number of features per class is used, as no advantages were found using the hierarchical model for large numbers of features. McCallum et al. (1998) propose a method based on naïve Bayes. The authors compare two techniques: (i) exploring all possible paths in the given hierarchy and (ii) greedily selecting at most two branches according to their probability, as done in (Koller & Sahami, 1997). Results show that the latter is more error prone while computationally more efficient. Mladenić & Grobelnik (1998) use the hierarchical structure to decompose a problem into a set of subproblems, corresponding to categories (i.e., the nodes of the hierarchy). For each subproblem, a naïve Bayes classifier is generated, considering examples belonging to the given category, including all examples classified in its subtrees. The classification applies to all nodes in parallel; a document is passed down to a category only if the posterior probability for that category is higher than a user-defined threshold. D'Alessio et al. (2000) propose a system in which, for a given category, the classification is based on a weighted sum of feature occurrences that should be greater than the category threshold. Both single and multiple classifications are possible for each document to be tested. The classification of a document proceeds top-down possibly through multiple paths. An innovative contribution of this work is the possibility of restructuring a given hierarchy or building a new one from scratch. Dumais & Chen (2000) use the hierarchical structure for two purposes: (i) training several SVMs, one for each intermediate node and (ii) classifying documents by combining scores from SVMs at different levels. The sets of positive and negative examples are built considering documents that belong to categories at the same level, and different feature sets are built, one for each category. Several combination rules have also been assessed. In the work of Ruiz & Srinivasan (2002), a variant of the Hierarchical Mixture of Experts model is used. A hierarchical classifier combining several neural networks is also proposed in (Weigend et al., 1999). Gaussier et al. (2002) propose a hierarchical generative model for textual data, i.e., a model for hierarchical clustering and categorization of co-occurrence data, focused on documents organization. In (Rousu et al., 2005), a kernel-based approach for hierarchical text classification in a multi-label context is presented. The work demonstrates that the use of the dependency structure of microlabels (i.e., unions of partial paths in the tree) in a Markovian Network framework leads to improved prediction accuracy on deep hierarchies. Optimization is made feasible by utilizing decomposition of the original problem and making incremental conditional gradient search in the subproblems. Ceci & Malerba (2007) present a comprehensive study on hierarchical classification of Web documents. They extend a previous work (Ceci & Malerba, 2003) considering hierarchical feature selection mechanisms, a naïve Bayes algorithm aimed at avoiding problems related to different document lengths, the validation of their framework for a probabilistic SVM-based classifier, and (iv) an automated threshold selection algorithm. More recently, in (Esuli et al., 2008), the authors propose a multi-label hierarchical text categorization algorithm consisting of a hierarchical variant of ADABOOST.MH, a well-known member of the family of "boosting" learning algorithms. Bennett & Nguyen (2009) study the problem of the error propagation under the assumption that the higher the node in the hierarchy is the worse is the mistake, as well as the problem of dealing with increasingly complex decision surfaces. Brank et al. (2010) deal with the problem of classifying textual documents into a topical hierarchy of categories. They construct a coding matrix gradually, one column at a time, each new column being defined in a way that the corresponding binary classifier attempts to correct the most common mistakes of the current ensemble of binary classifiers. The goal is to achieve good performance while keeping reasonably low the number of binary classifiers.

2.3 MultiAgent Systems in information retrieval

Autonomous agents and MAS have been successfully applied to a number of problems and have been largely used in different application domains (Wooldridge & Jennings, 1995).

As for MAS in IR, in the literature, several centralized agent-based architectures aimed at performing IR tasks have been proposed. Among others, let us recall NewT (Sheth & Maes, 1993), Letizia (Lieberman, 1995), WebWatcher (Armstrong et al., 1995), and SoftBots (Etzioni & Weld, 1995). NewT is composed by a society of information-filtering interface agents, which learn user preferences and act on her/his behalf. These information agents use a keyword-based filtering algorithm, whereas adaptive techniques are relevance feedback and genetic algorithms. Letizia is an intelligent user-interface agent able to assist a user while browsing the Web. The search for information is performed through a cooperative venture between the user and the software agent: both browse the same search space of linked Web documents, looking for interesting ones. WebWatcher is an information search agent that follows Web hyperlinks according to user interests, returning a list of links deemed interesting. In contrast to systems for assisted browsing or IR, SoftBots accept high-level user goals and dynamically synthesize the appropriate sequence of Internet commands according to a suitable ad-hoc language.

Despite the fact that a centralized approach could have some advantages, in IR tasks it may encompass several problems, in particular how to scale up the architectures to large numbers of users, how to provide high availability in case of constant demand of the involved services, and how to provide high trustability in case of sensitive information, such as personal data.

To overcome the above drawbacks, suitable MAS devoted to perform IR tasks have been proposed. In particular, Sycara et al. (2001) propose Retsina, a MAS infrastructure applied in many domains. Retsina is an open MAS infrastructure that supports communities of heterogeneous agents. Three types of agents have been defined: (i) *interface agents*, able to display the information to the users; (ii) *task agents*, able to assist the user in the process of handling her/his information; and (iii) *information agents*, able to gather relevant information from selected sources.

Among other MAS, let us recall IR-agents (Jirapanthong & Sunetnanta, 2000), CEMAS (Bleyer, 1998) and the cooperative multiagent system for Web IR proposed in (Shaban et al., 2004). IR-agents implement an XML-based multiagent model for IR. The corresponding framework is composed of three kinds of agents: (i) *managing agents*, aimed at extracting the semantics of information and at performing the actual tasks imposed by coordinator agents, (ii) *interface agents*, devised to interact with the users, and (iii) *search agents*, aimed at discovering relevant information on the Web. IR-agents do not take into account personalization, while providing information in a structured form without the adoption of specific classification mechanisms. In CEMAS, Concept Exchanging MultiAgent System, the basic idea is to provide specialized agents for exchanging concepts and links, representing the user, searching for new relevant documents matching existing concepts, and supporting agent coordination. Although CEMAS provides personalization and classification mechanisms based on a semantic approach, and it is mainly aimed at supporting scientists while looking for comprehensive information about their research interests. Finally, in (Shaban et al., 2004) the underlying idea is to adopt intelligent agents that mimic everyday-life activities of information seekers. To this end, agents are also able to profile the user in order to anticipate and achieve her/his preferred goals. Although interesting, the approach is mainly focused on cooperation among agents rather than on IR issues.

3. The proposed approach

A system for information retrieval must take into account several issues, the most relevant being:

1. how to deal with different information sources and to integrate new information sources without re-writing significant parts of it;
2. how to suitably encode data in order to put into evidence the informative content useful to discriminate among categories;
3. how to control the imbalance between relevant and irrelevant articles (the latter being usually much more numerous than the former);
4. how to allow the user to specify her/his preferences;
5. how to exploit the user's feedback to improve the overall performance of the system.

The above issues are typically strongly interdependent in state-of-the-art systems. To better concentrate on these aspects separately, we adopted a layered multiagent architecture, able to promote the decoupling among all aspects deemed relevant.

To perform the task of retrieving scientific publications, the actual system –sketched in Figure 1– involves three main activities: extracting the required information from selected online sources, categorizing it according to a given taxonomy while taking into account also users preferences, and providing suitable feedback mechanisms.

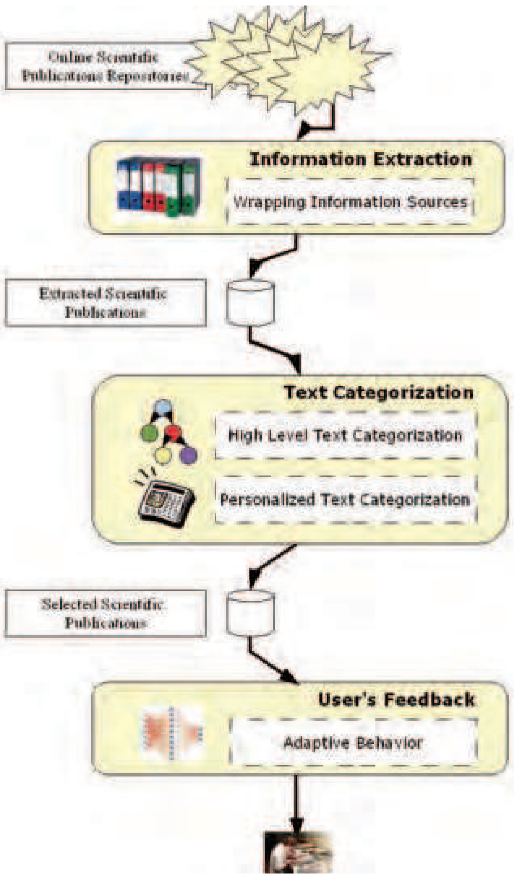
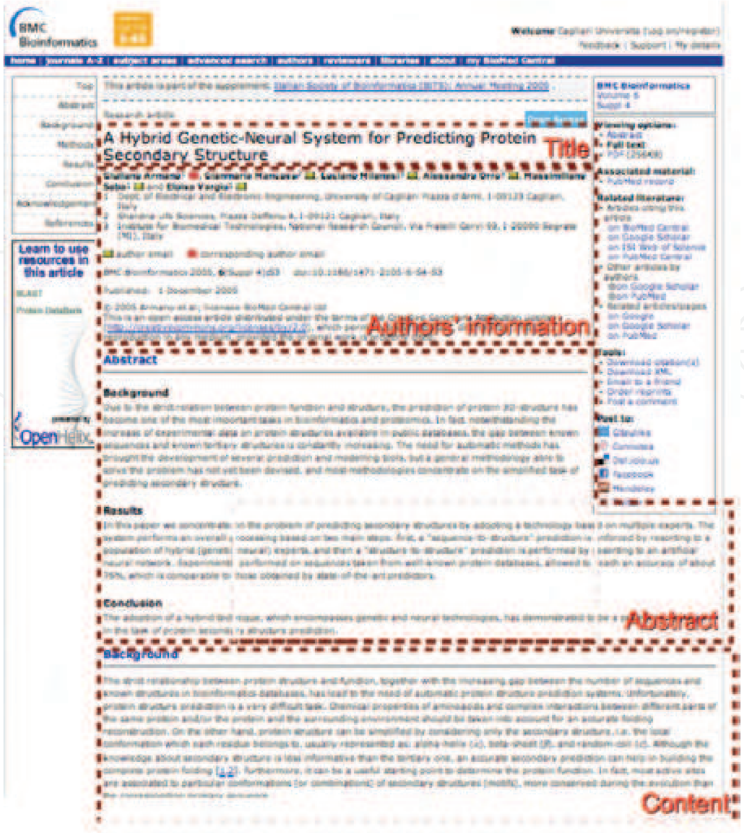

Online Scientific Publications Repositories
Information Extraction
Wrapping Information Sources
Extracted Scientific Publications
Text Categorization
High Level Text Categorization
Personalized Text Categorization
Selected Scientific Publications
User's Feedback
Adaptive Behavior

Fig. 1. PUB.MAS: the multiagent system devised for classifying bioinformatics publications.



BMC Bioinformatics

A Hybrid Genetic-Neural System for Predicting Protein Secondary Structure

Abstract

Background

Results

Conclusion

Background

Learn to use resources in this article

Fig. 2. The structure of a BMC Bioinformatics page.

3.1 Information extraction

This phase is devoted to deal with the huge amount of information provided by information sources. To this end suitable wrappers have been implemented, able to handle the structure of a document by saving the information about the corresponding metadata. In general, given a Web source, a specific wrapper must be implemented, able to map each Web page, designed according to the constraints imposed by the Web source, to a suitable description, which contains relevant data in a structured form –such as title, text content, and references.

To make this point clearer, let us consider the structure of the BMC Bioinformatics page of the paper "A Hybrid Genetic-Neural System for Predicting Protein Secondary Structure" (Armano et al., 2005) reported in Figure 2. In this case, it is quite easy to implement the mapping function, since, for each description field, a corresponding tag exists, making it very simple to process the pages.

A suitable encoding of the text content has also been enforced during this phase: all non-informative words such as prepositions, conjunctions, pronouns and very common verbs are deleted using a stop-word list; after that, a standard stemming algorithm (Porter, 1980) removes the most common morphological and inflexional suffixes. The subsequent step requires the adoption of suitable domain knowledge. For each category of the underlying taxonomy, feature selection (based on the information-gain heuristics) has been adopted to reduce the dimensionality of the feature space.

3.2 Text categorization

Scientific publications are classified according to a high-level taxonomy, which is independent from the specific user. To this end, classifiers are combined according to the links that hold within the taxonomy, giving rise to "vertical" and "horizontal" combinations of classifiers.

3.2.1 Vertical combination

The Approach

Vertical combination is currently performed by resorting to Progressively Filtering (PF), a simple categorization technique framed within the local classifier per node approach, which admits only binary decisions. In PF, each classifier is entrusted with deciding whether the input in hand can be forwarded or not to its children. The first proposals in which sequential boolean decisions are applied in combination with local classifiers per node can be found in (D'Alessio et al., 2000), (Dumais & Chen, 2000), and (Sun & Lim, 2001). In Wu et al. (2005), the idea of mirroring the taxonomy structure through binary classifiers is clearly highlighted; the authors call this technique "binarized structured label learning".

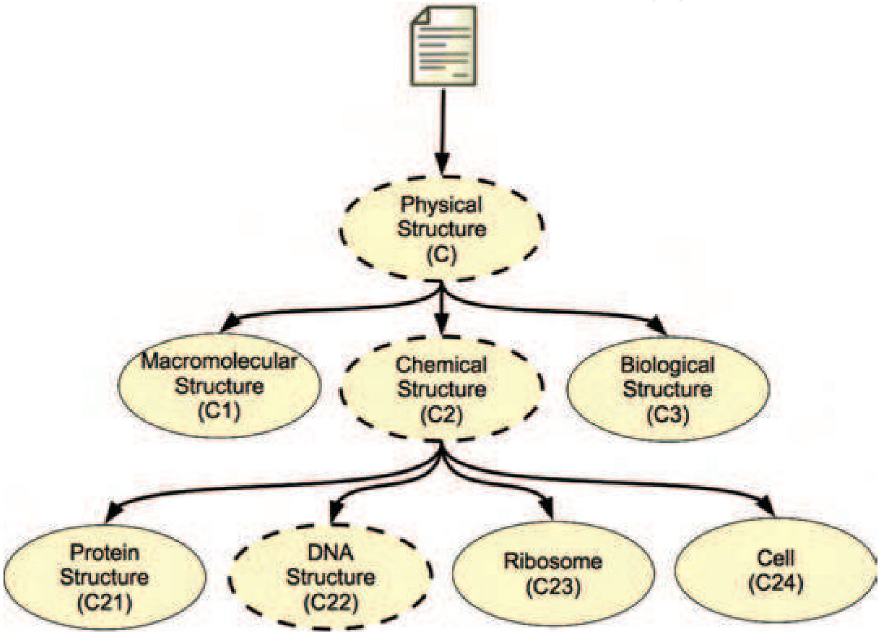


Fig. 3. An example of PF (highlighted with bold-dashed lines).

In PF, given a taxonomy, where each node represents a classifier entrusted with recognizing all corresponding positive inputs (i.e., interesting documents), each input traverses the taxonomy as a "token", starting from the root. If the current classifier recognizes the token as positive, it is passed on to all its children (if any), and so on. A typical result consists of activating one or more branches within the taxonomy, in which the corresponding classifiers have accepted the token. Figure 3 gives an example of how PF works. A theoretical study of the approach is beyond the scope of this chapter, the interested reader could refer to (Armano, 2009) for further details.

A simple way to implement PF consists of unfolding the given taxonomy into pipelines of classifiers, as depicted in Figure 4. Each node of the pipeline is a binary classifier able to recognize whether or not an input belongs to the corresponding class (i.e., to the corresponding node of the taxonomy). Partitioning the taxonomy in pipelines gives rise to a set of new classifiers, each represented by a pipeline.

Finally, let us note that the implementation of PF described in this chapter performs a sort of "flattening" though *preserving* the information about the hierarchical relationships embedded in a pipeline (Addis et al., 2010). For instance, the pipeline $\langle C, C2, C21 \rangle$ actually represents the classifier $C21$, although the information about the existing subsumption relationships (i.e., $C21 \leq C2 \leq C$) is preserved.

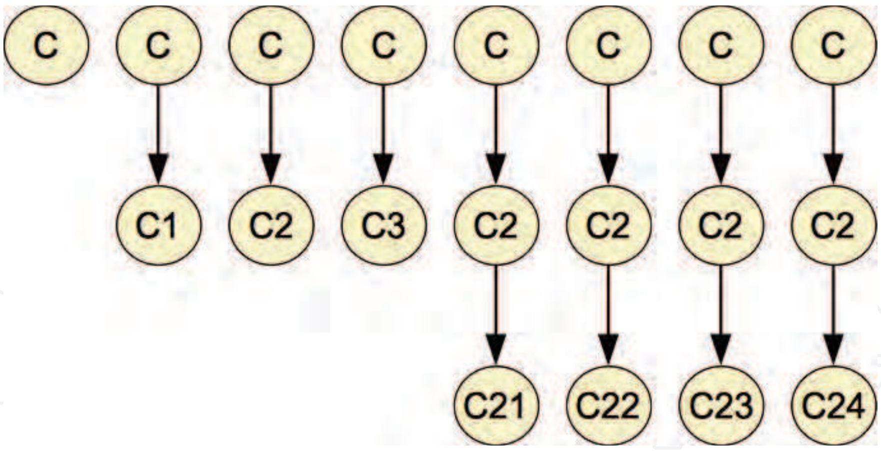


Fig. 4. The pipelines corresponding to the taxonomy in Figure 3.

The Threshold Selection Algorithm

As we know from classical text categorization, given a set of documents D and a set of labels C, a function $CSV_i : D \rightarrow [0,1]$ exists for each $c_i \in C$. We assume that the behavior of c_i is controlled by a threshold θ_i, responsible for relaxing or restricting the acceptance rate of the corresponding classifier. Given $d \in D$, $CSV_i(d) \geq \theta_i$ permits to categorize d under c_i, whereas $CSV_i(d) < \theta_i$ is interpreted as a decision not to categorize d under c_i.

In PF, let us still assume that CSV_i exists for each $c_i \in C$, with the same semantics adopted in the classical case. Considering a pipeline π, composed of n classifiers, the acceptance policy strictly depends on the vector $\theta_\pi = \langle \theta_1, \theta_2, \cdots, \theta_n \rangle$ that embodies the thresholds of all classifiers in π. In order to categorize d under π, the following constraint must be satisfied: $\forall k = 1 \ldots n$, $CSV_i(d) \geq \theta_k$; otherwise, d is not categorized under c_i.

A further simplification of the problem consists of allowing a classifier to have different behaviors, depending on which pipeline it is embedded in. Each pipeline can be considered in isolation from the others. For instance, given $\pi_1 = \langle C, C2, C21 \rangle$ and $\pi_2 = \langle C, C2, C22 \rangle$, the classifier C is not compelled to have the same threshold in π_1 and in π_2 (the same holds for $C2$).

Given a utility function[4], we are interested in finding an effective and computationally "light" way to reach a sub-optimum in the task of determining the best vector of thresholds. Unfortunately, finding the best acceptance thresholds is a difficult task. Exhaustively trying each possible combination of thresholds (brute-force approach) is unfeasible, the number of thresholds being virtually infinite. However, the brute-force approach can be approximated by defining a granularity step that requires to check only a finite number of points in the range $[0,1]$, in which the thresholds are permitted to vary with step δ. Although potentially useful, this "relaxed" brute force algorithm for calibrating thresholds (RBF for short) is still too heavy from a computational point of view. On the contrary, the threshold selection algorithm described in this chapter is characterized by low time complexity while maintaining the capability of finding near-optimum solutions.

Bearing in mind that the lower the threshold the less restrictive is the classifier, we adopt the greedy bottom-up algorithm for selecting decision threshold that relies on two functions described in (Addis et al., 2011):

[4] Different utility functions (e.g., precision, recall, F_β, user-defined) can be adopted, depending on the constraints imposed by the underlying scenario.

- *Repair* ($\mathcal{R}$), which operates on a classifier C by increasing or decreasing its threshold –i.e., $\mathcal{R}(up, C)$ and $\mathcal{R}(down, C)$, respectively– until the selected utility function reaches and maintains a local maximum.
- *Calibrate* ($\mathcal{C}$), which operates going downwards from the given classifier to its offspring. It is intrinsically recursive and, at each step, calls $\mathcal{R}$ to calibrate the current classifier.

Given a pipeline $\pi = \langle C_1, C_2, \ldots, C_L \rangle$, *TSA* is defined as follows (all thresholds are initially set to 0):

$$TSA(\pi) := for \;\; k = L \;\; downto \;\; 1 \;\; do \;\; \mathcal{C}(up, C_k) \tag{1}$$

which asserts that $\mathcal{C}$ is applied to each node of the pipeline, starting from the leaf ($k = L$). The *Calibrate* function is defined as follows:

$$\begin{array}{l} \mathcal{C}(up, C_k) := \mathcal{R}(up, C_k), \quad k = L \\ \mathcal{C}(up, C_k) := \mathcal{R}(up, C_k); \mathcal{C}(down, C_{k+1}), \quad k < L \end{array} \tag{2}$$

and

$$\begin{array}{l} \mathcal{C}(down, C_k) := \mathcal{R}(down, C_k), \quad k = L \\ \mathcal{C}(down, C_k) := \mathcal{R}(down, C_k); \mathcal{C}(up, C_{k+1}), k < L \end{array} \tag{3}$$

where the ";" denotes a sequence operator, meaning that in "*a;b*" action *a* is performed *before* action *b*. The reason why the direction of threshold optimization changes at each call of Calibrate (and hence of Repair) lies in the fact that increasing the threshold θ_{k-1} is expected to forward less false positives to C_k, which allows to decrease θ_k. Conversely, decreasing the threshold θ_{k-1} is expected to forward more false positives to C_k, which must react by increasing θ_k.

It is worth pointing out that, as also noted in (Lewis, 1995), the sub-optimal combination of thresholds depends on the adopted dataset, hence it needs to be recalculated for each dataset.

3.2.2 Horizontal combination

To express what the user is really interested in, we implemented suitable horizontal composition strategies by using extended boolean models (Lee, 1994). In fact, a user is typically not directly concerned with topics that coincide with classes of the given taxonomy. Rather, a set of arguments of interest can be obtained by composing such generic topics with suitable logical operators (i.e., *and*, *or*, and *not*). For instance, a user might be interested in being kept informed about all articles that involve both "cell" *and* "nucleus". This compound topic can be dealt with by composing the *cell* and the *nucleus* classifiers. To address this issue, we adopted a soft boolean perspective, in which the combination is evaluated using *P*-norms (Golub & Loan, 1996).

3.3 Users' feedback

So far, a simple solution based on the *k*-NN technology has been implemented and experimented to deal with the problem of supporting the user's feedback. When a irrelevant article is evidenced by the user, it is immediately embedded in the training set of the *k*-NN classifier that implements the feedback. A check performed on this training set after inserting the negative example allows to trigger a procedure entrusted with keeping the number of negative and positive examples balanced. In particular, when the ratio between negative and positive examples exceeds a given threshold (by default set to 1.1), some examples are randomly extracted from the set of "true" positive examples and embedded in the above training set.

4. PUB.MAS

To retrieve and categorize scientific publications, we customized X.MAS (Addis et al., 2008), a generic multiagent architecture built upon JADE (Bellifemine et al., 2007) devised to facilitate the implementation of information retrieval and information filtering applications. The motivation for adopting a MAS lies in the fact that a centralized classification system might be quickly overwhelmed by a large and dynamic document stream, such as daily-updated online publications. Furthermore, the Web is intrinsically a pervasive system and offers the opportunity to take advantage of distributed computing paradigms and spread knowledge resources.

4.1 The system

PUB.MAS, is organized in the three "virtual" layers depicted in Figure 1, by customizing X.MAS as follows:

- *Information Level*. Agents at this level are devoted to deal with the selected information sources. A wrapper able to deal with the RSS (Really Simple Syndication) format has been implemented, aimed at extracting publications from BMC Bioinformatics. It is worth pointing out that the RSS format allows to easily process any given page, since a corresponding RSS tag exists for each relevant item. Furthermore, the growing amount of Web Services providing scientific publications requires the implementation of wrappers explicitly devoted to extract information from them. In order to invoke Web Services from our multiagent system, required to access the PubMed Central Web Service, we implemented an ad-hoc wrapper by adopting WSIG (Greenwood & Calisti, 2004).
- *Filter Level*. Filter agents are devoted to select information deemed relevant to the users, and to cooperate to prevent information from being overloaded and redundant. A suitable encoding of the text content has been enforced at this level to facilitate the work of agents belonging to the task level. As already pointed out, all non-informative words such as prepositions, conjunctions, pronouns and very common verbs are removed using a stop-word list. After that, a standard stemming algorithm removes the most common morphological and inflexional suffixes. Then, for each category, feature selection, based on the information-gain heuristics, has been adopted to reduce the dimensionality of the feature space.
- *Task Level*. Task agents are devoted to identify relevant scientific publications, depending on user interests. Agents belonging to this architectural level are devoted to perform two kinds of actions: to classify any given input in accordance with the selected set of categories, and to decide whether it may be of interest to the user or not. Each task agent has been trained by resorting to a state-of-the-art technique, i.e. k-NN, in its *"weighted"* variant (Cost & Salzberg, 1993). The choice of adopting weighted k-NN stems from the fact that it does not require specific training and is very robust with respect to the impact of noisy data. Furthermore, the adoption of weighted k-NN is related with the choice of P-norms for implementing the "and" operation, as P-norms combination rules require values in [0,1].
- *Interface Level*. Interface agents are devoted to perform the feedback that originates from the users –which can be exploited to improve the overall ability of discriminating relevant from irrelevant inputs. To this end, PUB.MAS uses the k-NN solution previously described.

4.2 The prototype

Since our primary interest consists of classifying scientific articles for bioinformaticians or biologists, a high-level *is-a* taxonomy has been extracted from the TAMBIS ontology (Baker et al., 1999). A fragment of the taxonomy is depicted in Figure 5.

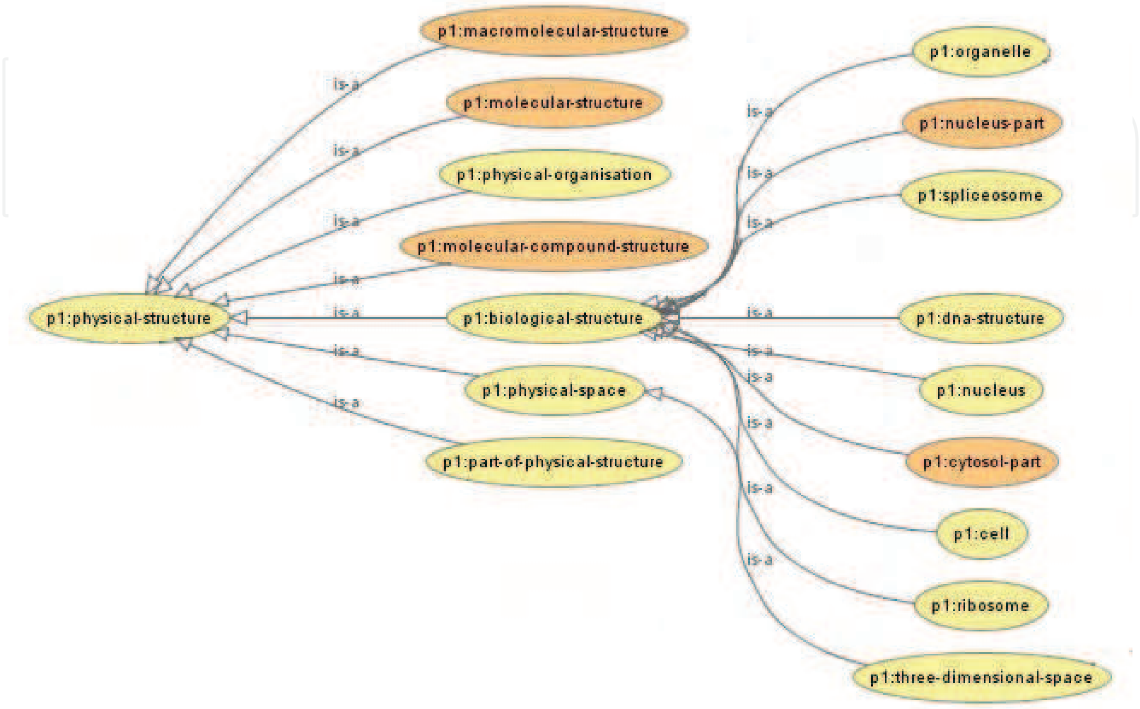


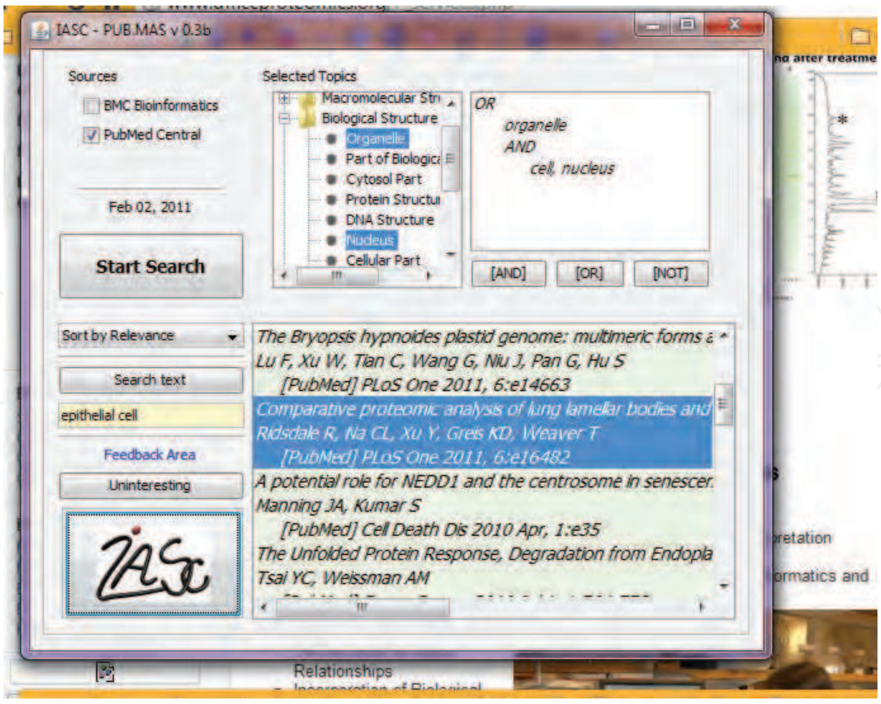
Fig. 5. A fragment of the adopted taxonomy

Through a suitable user interface (see Figure 6), the user can set the sources from which publications will be extracted and the topics s/he is interested in. As for the digital archives, the user can choose between BMC Bioinformatics and PubMed Central. As for the topics of interest, the user can select one or more categories in accordance with the adopted taxonomy and compose them in order to build her/his personal document collection. For instance, in the example reported in Figure 6, the user queries the system on *(cell AND nucleus) OR (organelle)*. The search for relevant documents is activated by clicking on the *Start Search* button. First, information agents devoted to handle Web sources extract the documents. Then, all agents that embody a classifier trained on the selected topics are involved to perform text categorization. Finally, the system supplies the user with the selected articles through suitable interface agents (see Figure 7).

4.3 Experimental results

To assess the system, different kinds of tests have been performed, each aimed at highlighting (and getting information about) a specific issue. First, we estimated the *normalized* confusion matrix for each classifier belonging to the highest level of the taxonomy. Afterwards, we tested the importance of defining user's interests by resorting to a relaxation of the logical operators. Finally, we assessed the solution devised for implementing user's feedback, based on the *k*-NN technique.

Tests have been performed using selected publications extracted from the BMC Bioinformatics site and from the PubMed Central digital archive. Publications have been classified by an expert of the domain according to the proposed taxonomy. For each item of the taxonomy, a


IASC - PUB.MAS v 0.3b

Sources

□ BMC Bioinformatics

☑ PubMed Central

Feb 02, 2011

Selected Topics

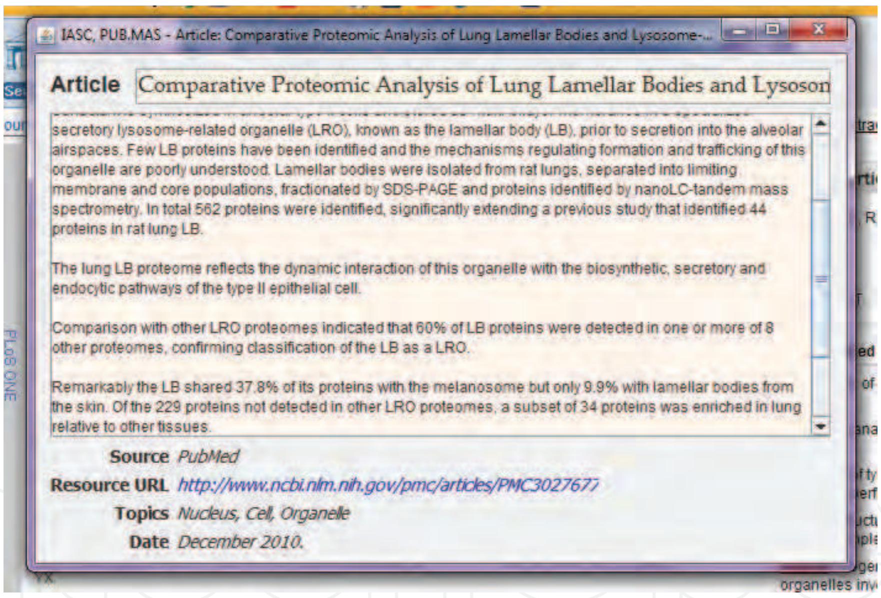
OR
organelle
AND
cell, nucleus

Sort by Relevance

epithelial cell

The Bryopsis hypnoides plastid genome: multimeric forms a
Lu F, Xu W, Tian C, Wang G, Niu J, Pan G, Hu S
[PubMed] PLoS One 2011, 6:e14663
Comparative proteomic analysis of lung lamellar bodies and
Ridsdale R, Na CL, Xu Y, Greis KD, Weaver T
[PubMed] PLoS One 2011, 6:e16482
A potential role for NEDD1 and the centrosome in senescer
Manning JA, Kumar S
[PubMed] Cell Death Dis 2010 Apr, 1:e35
The Unfolded Protein Response, Degradation from Endopla
Tsai YC, Weissman AM

Fig. 6. The user interface


IASC, PUB.MAS - Article: Comparative Proteomic Analysis of Lung Lamellar Bodies and Lysosome-...

Article Comparative Proteomic Analysis of Lung Lamellar Bodies and Lysoson

secretory lysosome-related organelle (LRO), known as the lamellar body (LB), prior to secretion into the alveolar airspaces. Few LB proteins have been identified and the mechanisms regulating formation and trafficking of this organelle are poorly understood. Lamellar bodies were isolated from rat lungs, separated into limiting membrane and core populations, fractionated by SDS-PAGE and proteins identified by nanoLC-tandem mass spectrometry. In total 562 proteins were identified, significantly extending a previous study that identified 44 proteins in rat lung LB.

The lung LB proteome reflects the dynamic interaction of this organelle with the biosynthetic, secretory and endocytic pathways of the type II epithelial cell.

Comparison with other LRO proteomes indicated that 60% of LB proteins were detected in one or more of 8 other proteomes, confirming classification of the LB as a LRO.

Remarkably the LB shared 37.8% of its proteins with the melanosome but only 9.9% with lamellar bodies from the skin. Of the 229 proteins not detected in other LRO proteomes, a subset of 34 proteins was enriched in lung relative to other tissues.

Source *PubMed*

Resource URL *http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3027677*

Topics *Nucleus, Cell, Organelle*

Date *December 2010.*

Fig. 7. A publication retrieved by PUB.MAS

set of about 100-150 articles has been selected to train the corresponding *k*-NN classifier, and 300-400 articles have been used to test it.

As for the estimation of the normalized confusion matrices (one for each classifier), we fed classifiers with balanced sets of positive and negative examples. Given a classifier, we performed several runs to obtain an averaged confusion matrix. Normalization has been imposed row by row on the averaged confusion matrix. In particular, true negatives and false positives are divided by the number of negative examples; conversely, the number of false negatives and true positives are divided by the number of positive examples. In so doing, we obtain an estimation of the conditional probability $P(\hat{c}(x)|c(x))$, where x is the input to be classified, $\hat{c}(x)$ is the output of the classifier, and $c(x)$ is the category of x

To assess the impact of exploiting a taxonomy over precision and recall, we selected some relevant samples of three classifiers in pipeline. They have been tested by imposing randomly-selected relevant and irrelevant inputs, their ratio being set to 1/100, to better approximate the expected behavior of the pipelines in real-world conditions. Averaging the results obtained in all experiments in which a pipeline of three classifiers was involved, PF allowed to reach an accuracy of 95%, a precision of 80%, and a recall of 44%.

Figure 8 and 9 report experimental results focused on average precision and recall, respectively. Experimental results are compared with those derived theoretically. Let us note that results show that the filtering effect of a pipeline is not negligible. In particular, in presence of imbalanced inputs, a pipeline of three classifiers is able to counteract a lack of equilibrium of about 10 irrelevant articles vs. one relevant article. Since, at least in principle, the filtering activity goes with the power of the number of classifiers involved in the pipeline, it is easy to verify that PF could also counteract a ratio between irrelevant and relevant articles with an order of magnitude of hundreds or thousands, provided that the number of levels of the underlying taxonomy is deep enough (at least 3-4).

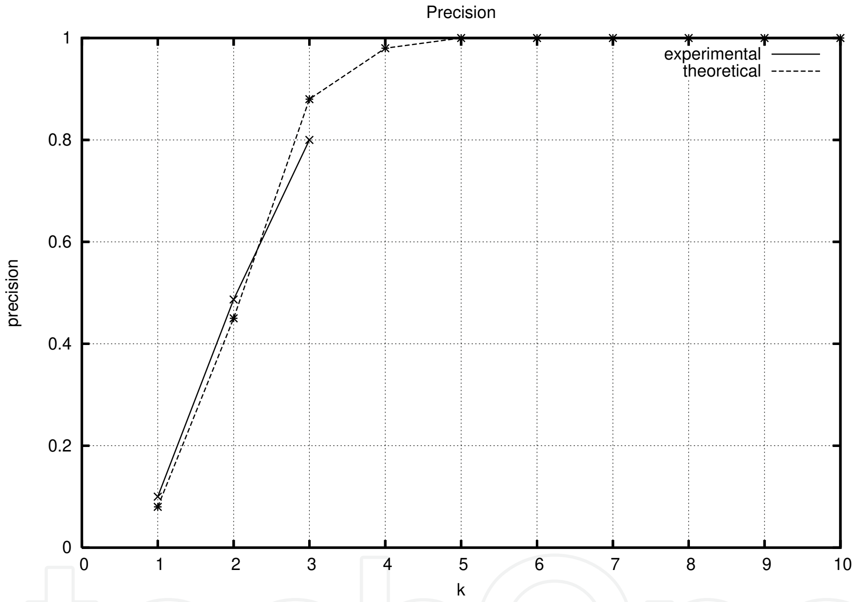


Fig. 8. Average precision using three classifiers in pipeline

To test the validity of the horizontal composition mechanisms, the system has been tested on 20 selected users. The behavior of the system has been monitored over a two-week period by conducting regular interviews with each user to estimate her/his satisfaction and the correctness of the process. All users stated their satisfaction with the system after just one or two days.

As for the user's feedback, we obtained an improvement of about 0.3% on the precision of the system by populating a k-NN classifier with examples selected as relevant by the system, taking care of balancing true positives with false positives.

5. Conclusions

It becomes more and more difficult for Web users to search for, find, and select contents according to their preferences. The same happens when researchers surf the Web searching

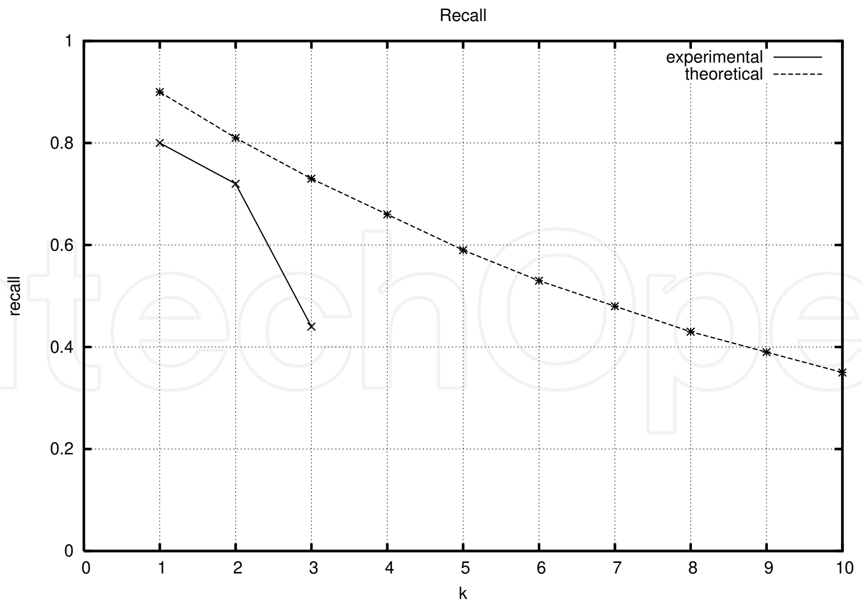


Fig. 9. Average recall using three classifiers in pipeline

for scientific publication of interest. Hence, supporting users in the task of dealing with the information provided by the Web is a primary issue. In this chapter, we focused on automatically retrieving and categorizing scientific publications and presented PUB.MAS, a system devoted to provide personalized search results in terms of bioinformatics publications. The system encompasses three main tasks: extracting scientific publications from online repositories, classifying them using hierarchical text categorization, and providing suitable feedback mechanisms. To validate the approach, we performed several experiments. Results show that the approach is effective and can be adopted in practice.

6. Acknowledgements

This work has been supported by the Italian Ministry of Education under the projects: FIRB ITALBIONET (RBPR05ZK2Z), Bioinformatics analysis applied to Populations Genetics (RBIN064YAT 003) and SHIWA.

7. References

Addis, A., Armano, G. & Vargiu, E. (2008). From a generic multiagent architecture to multiagent information retrieval systems, *AT2AI-6, Sixth International Workshop, From Agent Theory to Agent Implementation*, pp. 3–9.

Addis, A., Armano, G. & Vargiu, E. (2010). Assessing progressive filtering to perform hierarchical text categorization in presence of input imbalance, *Proceedings of International Conference on Knowledge Discovery and Information Retrieval (KDIR 2010)*, pp. 14–23.

Addis, A., Armano, G. & Vargiu, E. (2011). A comparative experimental assessment of a threshold selection algorithm in hierarchical text categorization, *Advances in Information Retrieval. The 33rd European Conference on Information Retrieval (ECIR 2011)*, pp. 32–42.

Armano, G. (2009). On the progressive filtering approach to hierarchical text categorization, *Technical report*, DIEE - University of Cagliari.

Armano, G., de Gemmis, M., Semeraro, G. & Vargiu, E. (2010). *Intelligent Information Access*, Springer-Verlag, Studies in Computational Intelligence series.

Armano, G., Manconi, A. & Vargiu, E. (2007). A multiagent system for retrieving bioinformatics publications from web sources, *IEEE TRANSACTIONS ON NANOBIOSCIENCE* 6(2): 104–109. Special Session on GRID, Web Services, Software Agents and Ontology Applications for Life Science.

Armano, G., Mancosu, G., Milanesi, L., Orro, A., Saba, M. & Vargiu, E. (2005). A hybrid genetic-neural system for predicting protein secondary structure, *BMC BIOINFORMATICS* 6 (suppl. 4):s3.

Armstrong, R., Freitag, D., Joachims, T. & Mitchell, T. (1995). Webwatcher: A learning apprentice for the world wide web, *AAAI Spring Symposium on Information Gathering*, pp. 6–12.

Baeza-Yates, R. A. & Ribeiro-Neto, B. (1999). *Modern Information Retrieval*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Baker, P., Goble, C., Bechhofer, S., Paton, N., Stevens, R. & Brass, A. (1999). An ontology for bioinformatics applications, *Bioinformatics* 15(6): 510–520.

Bellifemine, F., Caire, G. & Greenwood, D. (2007). *Developing Multi-Agent Systems with JADE (Wiley Series in Agent Technology)*, John Wiley and Sons.

Bennett, P. N. & Nguyen, N. (2009). Refined experts: improving classification in large taxonomies, *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, ACM, New York, NY, USA, pp. 11–18.

Bleyer, M. (1998). *Multi-Agent Systems for Information Retrieval on the World Wide Web*, PhD thesis, University of Ulm, Germany.

Bollacker, K. D., Lawrence, S. & Giles, C. L. (2000). Discovering relevant scientific literature on the web, *IEEE Intelligent Systems* 15(2): 42–47.

Brank, J., Mladenić, D. & Grobelnik, M. (2010). Large-scale hierarchical text classification using svm and coding matrices, *Large-Scale Hierarchical Classification Workshop*.

Ceci, M. & Malerba, D. (2003). Hierarchical classification of HTML documents with WebClassII, *in* F. Sebastiani (ed.), *Proceedings of ECIR-03, 25th European Conference on Information Retrieval*, Berlin Heidelberg NewYork: Springer, pp. 57–72.

Ceci, M. & Malerba, D. (2007). Classifying web documents in a hierarchy of categories: a comprehensive study, *Journal of Intelligent Information Systems* 28(1): 37–78.

Corney, D. P. A., Buxton, B. F., Langdon, W. B. & Jones, D. T. (2004). Biorat: Extracting biological information from full-length papers, *Bioinformatics* 20(17): 3206–3213.

Cost, W. & Salzberg, S. (1993). A weighted nearest neighbor algorithm for learning with symbolic features, *Machine Learning* 10: 57–78.

Craven, M. & Kumlien, J. (1999). Constructing biological knowledge-bases by extracting information from text sources, *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, Germany, pp. 77–86.

D'Alessio, S., Murray, K. & Schiaffino, R. (2000). The effect of using hierarchical classifiers in text categorization, *Proceedings of of the 6th International Conference on Recherche dŠInformation Assistée par Ordinateur (RIAO)*, pp. 302–313.

Delfs, R., Doms, A., Kozlenkov, A. & Schroeder, M. (2004). Gopubmed: ontology-based literature search applied to gene ontology and pubmed, *Proc. of German Bioinformatics Conference*, pp. 169–178.

Dumais, S. T. & Chen, H. (2000). Hierarchical classification of Web content, *in* N. J. Belkin, P. Ingwersen & M.-K. Leong (eds), *Proceedings of SIGIR-00, 23rd ACM International Conference on Research and Development in Information Retrieval*, ACM Press, New York, US, Athens, GR, pp. 256–263.

Esuli, A., Fagni, T. & Sebastiani, F. (2008). Boosting multi-label hierarchical text categorization, *Inf. Retr.* 11(4): 287–313.

Etzioni, O. & Weld, D. (1995). Intelligent agents on the internet: fact, fiction and forecast, *IEEE Expert* 10(4): 44–49.

Friedman, C., Kra, P., Yu, H., Krauthammer, M. & Rzhetsky, A. (2001). Genies: a natural-language processing system for the extraction of molecular pathways from journal articles, *Bioinformatics* 17: S74–S82.

Gaussier, É., Goutte, C., Popat, K. & Chen, F. (2002). A hierarchical model for clustering and categorising documents, *in* F. Crestani, M. Girolami & C. J. V. Rijsbergen (eds), *Proceedings of ECIR-02, 24th European Colloquium on Information Retrieval Research*, Springer Verlag, Heidelberg, DE, Glasgow, UK, pp. 229–247.

Golub, G. & Loan, C. V. (1996). *Matrix Computations*, Baltimore: The Johns Hopkins University Press.

Greenwood, D. & Calisti, M. (2004). An automatic, bi-directional service integration gateway, *IEEE Systems, Cybernetics and Man Conference*.

Huang, L. (2000). A survey on web information retrieval technologies, *Technical report, ECSL*.

Janssen, W. C. & Popat, K. (2003). Uplib: a universal personal digital library system, *DocEng '03: Proceedings of the 2003 ACM symposium on Document engineering*, ACM Press, New York, NY, USA, pp. 234–242.

Jirapanthong, W. & Sunetnanta, T. (2000). An xml-based multi-agents model for information retrieval on www, *Proceedings of the 4th National Computer Science and Engineering Conference (NCSEC2000)*.

Kiritchenko, S., Matwin, S. & Famili, A. F. (2004). Hierarchical text categorization as a tool of associating genes with gene ontology codes, *2nd European Workshop on Data Mining and Text Mining for Bioinformatics*, pp. 26–30.

Koller, D. & Sahami, M. (1997). Hierarchically classifying documents using very few words, *in* D. H. Fisher (ed.), *Proceedings of ICML-97, 14th International Conference on Machine Learning*, Morgan Kaufmann Publishers, San Francisco, US, Nashville, US, pp. 170–178.

Lee, J. (1994). Properties of extended boolean models in information retrieval, *Proceedings of the 17th Annual international ACM SIGIR Conference on Research and Development in information Retrieval*, pp. 182–190.

Lewis, D. D. (1995). Evaluating and optimizing autonomous text classification systems, *SIGIR '95: Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, ACM, New York, NY, USA, pp. 246–254.

Lieberman, H. (1995). Letizia: An agent that assists web browsing, *in* C. S. Mellish (ed.), *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, Morgan Kaufmann publishers Inc.: San Mateo, CA, USA, Montreal, Quebec, Canada, pp. 924–929.

Mahdavi, F., Ismail, M. A. & Abdullah, N. (2009). Semi-automatic trend detection in scholarly repository using semantic approach, *Proceedings of World Academy of Science, Engineering and Technology*, pp. 224–226.

McCallum, A. K., Rosenfeld, R., Mitchell, T. M. & Ng, A. Y. (1998). Improving text classification by shrinkage in a hierarchy of classes, *in* J. W. Shavlik (ed.), *Proceedings of ICML-98, 15th International Conference on Machine Learning*, Morgan Kaufmann Publishers, San Francisco, US, Madison, US, pp. 359–367.

McNee, S. M., Albert, I., Cosley, D., Gopalkrishnan, P., Lam, S. K., Rashid, A. M., Konstan, J. A. & Riedl, J. (2002). On the recommending of citations for research papers, *Proceedings* *of the 2002 ACM conference on Computer supported cooperative work*, CSCW '02, ACM, New York, NY, USA, pp. 116–125.

Mladenić, D. & Grobelnik, M. (1998). Feature selection for classification based on text hierarchy, *Text and the Web, Conference on Automated Learning and Discovery CONALD-98*.

Porter, M. (1980). An algorithm for suffix stripping, *Program* 14(3): 130–137.

Ramu, C. (2001). SIR: a simple indexing and retrieval system for biological flat file databases, *Bioinformatics* 17(8): 756–758.

Ramu, C. (2003). SIRW: A web server for the simple indexing and retrieval system that combines sequence motif searches with keyword searches., *Nucleic Acids Res* 31(13): 3771–3774.

Rocco, D. & Critchlow, T. (2003). Automatic discovery and classification of bioinformatics web sources, *Bioinformatics* 19(15): 1927–1933.

Rousu, J., Saunders, C., Szedmak, S. & Shawe-Taylor, J. (2005). Learning hierarchical multi-category text classification models, *ICML '05: Proceedings of the 22nd international conference on Machine learning*, ACM, New York, NY, USA, pp. 744–751.

Ruiz, M. E. & Srinivasan, P. (2002). Hierarchical text categorization using neural networks, *Information Retrieval* 5(1): 87–118.

Shaban, K., Basir, O. & Kamel, M. (2004). Team consensus in web multi-agents information retrieval system, *Team Consensus in Web Multi-agents Information Retrieval System*, pp. 68–73.

Sheth, B. & Maes, P. (1993). Evolving agents for personalized information filtering, *Proceedings of the 9th Conference on Artificial Intelligence for Applications (CAIA-93)*, pp. 345–352.

Sun, A. & Lim, E. (2001). Hierarchical text classification and evaluation, *ICDM '01: Proceedings of the 2001 IEEE International Conference on Data Mining*, IEEE Computer Society, Washington, DC, USA, pp. 521–528.

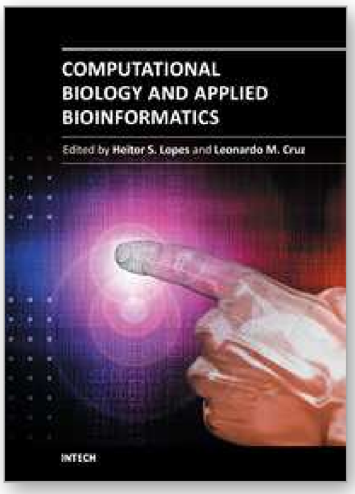
Sycara, K., Paolucci, M., van Velsen, M. & Giampapa, J. (2001). The RETSINA MAS infrastructure, *Technical Report CMU-RI-TR-01-05*, Robotics Institute Technical Report, Carnegie Mellon.

Tanabe, L., Scherf, U., Smith, L. H., Lee, J. K., Hunter, L. & Weinstein, J. N. (1999). Medminer: an internet text-mining tool for biomedical information, *BioTechniques* 27: 1210–1217.

Weigend, A. S., Wiener, E. D. & Pedersen, J. O. (1999). Exploiting hierarchy in text categorization, *Information Retrieval* 1(3): 193–216.

Wooldridge, M. J. & Jennings, N. R. (1995). Agent Theories, Architectures, and Languages: A Survey, *in* M. J. Wooldridge & N. R. Jennings (eds), *Workshop on Agent Theories, Architectures & Languages (ECAI'94)*, Vol. 890 of *Lecture Notes in Artificial Intelligence*, Springer-Verlag, Amsterdam, The Netherlands, pp. 1–22.

Wu, F., Zhang, J. & Honavar, V. (2005). Learning classifiers using hierarchically structured class taxonomies, *Proc. of the Symp. on Abstraction, Reformulation, and Approximation*, Vol. 3607, Springer Verlag, pp. 313–320.



Computational Biology and Applied Bioinformatics

Edited by Prof. Heitor Lopes

ISBN 978-953-307-629-4
Hard cover, 442 pages
Publisher InTech
Published online 02, September, 2011
Published in print edition September, 2011

Nowadays it is difficult to imagine an area of knowledge that can continue developing without the use of computers and informatics. It is not different with biology, that has seen an unpredictable growth in recent decades, with the rise of a new discipline, bioinformatics, bringing together molecular biology, biotechnology and information technology. More recently, the development of high throughput techniques, such as microarray, mass spectrometry and DNA sequencing, has increased the need of computational support to collect, store, retrieve, analyze, and correlate huge data sets of complex information. On the other hand, the growth of the computational power for processing and storage has also increased the necessity for deeper knowledge in the field. The development of bioinformatics has allowed now the emergence of systems biology, the study of the interactions between the components of a biological system, and how these interactions give rise to the function and behavior of a living being. This book presents some theoretical issues, reviews, and a variety of bioinformatics applications. For better understanding, the chapters were grouped in two parts. In Part I, the chapters are more oriented towards literature review and theoretical issues. Part II consists of application-oriented chapters that report case studies in which a specific biological problem is treated with bioinformatics tools.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Andrea Addis, Giuliano Armano, Eloisa Vargiu and Andrea Manconi (2011). Retrieving and Categorizing Bioinformatics Publications through a MultiAgent System, Computational Biology and Applied Bioinformatics, Prof. Heitor Lopes (Ed.), ISBN: 978-953-307-629-4, InTech, Available from: http://www.intechopen.com/books/computational-biology-and-applied-bioinformatics/retrieving-and-categorizing-bioinformatics-publications-through-a-multiagent-system