Text Categorization Using a Personalized, Adaptive, and Cooperative MultiAgent System

Giancarlo Cherchi, Andrea Manconi, Eloisa Vargiu
University of Cagliari
Piazza d'Armi, I-09123, Cagliari, Italy
Email: {cherchi,manconi,vargiu}@diee.unica.it

Dario Deledda
Arcadia Design
Loc. Is Coras, I-09028 Sestu, Cagliari, Italy
Email: dario.deledda@arcadiadesign.it

Abstract— **In this paper, a multiagent system for supporting users in retrieving information from heterogeneous data sources, and classifying them according to users' personal preferences, is presented. The system is built upon PACMAS, a generic architecture that supports the implementation of Personalized, Adaptive, and Cooperative MultiAgent Systems. Preliminary tests have been conducted to evaluate the effectiveness of the system in retrieving and classifying newspaper articles. Results show an avarage accuracy of about 80%.**

I. Introduction

The information available on the WWW is continuously growing from different points of view: information sources are increasing, topics discussed are becoming more and more heterogeneous, and stored data has reached a considerable size. It has become a difficult task for Internet users to select contents according to their personal interests, especially if contents are continuously updated (e.g., news, newspaper articles, reuters, rss feeds, blogs, etc.). Unfortunately, traditional filtering techniques based on keyword search are often inadequate to express what the user is really searching for. Furthermore, users often need to refine by hand the achieved results.

Supporting users in handling with the enormous and widespread amount of web information is becoming a primary issue. To this end, an automated system able to retrieve information from the Internet, and to select the contents really deemed relevant for the user, through a text categorization process, would be very helpful.

In the literature, software agents have been widely proposed for retrieving information from the web (see for example [9] [7] [11]). Furthermore, several machine learning techniques have been applied to text categorization (see [18] for a detailed comparison).

In this paper, we focus on the problem of retrieving articles from italian online newspapers, and classifying them using suitable machine learning techniques. In particular, we exploit the PACMAS architecture [2] to build a personalized, adaptive, and cooperative multiagent system.

The outline of the paper is organized as following: in Section II some related work on agent-based information retrieving is briefly recalled; Section III briefly illustrates the text categorization proble; Section IV sketches the PACMAS architecture; In Section V, all customizations devised for explicitly dealing with text categorization are presented, together with some experimental results; Section VI draws conclusions and points to future work.

II. Agent-based Systems for Information Retrieving

Several multiagent systems have been proposed to support the user in the task of retrieving information from the web. Among them let us recall NewT [16], Letizia [13], WebWatcher [3], and SoftBot [7].

NewT [16] is designed as a collection of information filtering interface agents. Interface agents are intelligent and autonomous computer programs, which learn users' preferences and act on their behalf. This system uses a keyword-based filtering algorithm. The learning mechanisms used are relevance feedback and genetic algorithms.

Letizia [13] is a user interface agent that assists a user browsing the World Wide Web. The model adopted by this system is that the search for information is a cooperative venture between the human user and an intelligent software agent. Letizia and the user both browse the same search space of linked web documents, looking for "interesting" ones.

WebWatcher [3] is an information search agent that follows web hyperlinks according to users' interests, returning a list of interesting links to the user.

In contrast to systems for assisted browsing or information retrieval, the SoftBot [7] accepts high level user goals and dynamically synthesizes the appropriate sequence of Internet commands using a suitable ad-hoc language to satisfy those goals.

Finally, let us point out that current web search engines basically rely only on purely syntactical textual information retrieval. There are only a few approaches that try to integrate a set of different and specialized sources, but unfortunately it is very difficult to maintain and to develop this kind of systems [9].

III. Text Categorization

The main goal of text categorization is to classify documents into a set of predefined categories. Each document can be in multiple or exactly one category. Using machine learning, the objective is to learn classifiers from examples, which perform the category assignments automatically, according to a supervised learning approach.

A major characteristic, or difficulty, of text categorization problems is the high dimensionality of the feature space. The native feature space consists of the unique terms (words or phrases) that occur in documents, which can be tens or hundreds of thousands of terms, even for a moderate-sized text collection. This is prohibitively complex for many learning algorithms. Thus, the first step in text categorization is to transform documents into a representation suitable for the underlying learning algorithm and the classification task.

After counting the number of occurences of a word `w` in a document –giving rise to an unordered *bag of words* [1]– suitable stemming algorithms [15] are applied to avoid unnecessarily large feature vectors. Each distinct word stem w_i corresponds to a feature, with the number of occurrences (in the entire document) of the word w_i as value. Words are considered as features only if they occur in the training data at least a predefined number of times except when they are considered as *stop-words* (like *and*, *or*, *is*, etc.).

To further reduce the number of considered terms, suitable feature selection methods can be applied. Automatic feature selection methods include the removal of non-informative terms according to corpus strategies, and the construction of new features which combine lower-level features (i.e., terms) into higher-level orthogonal dimension. Among different feature selection methods, let us recall document frequency, information gain, mutual information, a χ^2 statistic, and term strength (see [21] for a detailed comparison among them).

After selecting the terms, for each document a feature vector is generated, whose elements are the feature values of each term. A commonly used feature value is the `TF` (Term Frequency) `x IDF` (Inverse Document Frequency) measure.

Among machine learning techniques applied to text categorization, let us cite multivariant regression models [19], *k*Nearest Neighbor classification [20], Bayes probabilistic approaches [17], decision trees [12], neural networks [6], symbolic rule learning [14] and inductive learning algorithms [4].

IV. The PACMAS Architecture

PACMAS, which stands for Personalized Adaptive and Cooperative MultiAgent System, is a generic multiagent architecture, aimed at retrieving, filtering and reorganizing information according to the users' interests. PACMAS agents can be personalized, adaptive, and cooperative, depending on their specific role (see [2] for details).

PACMAS Macro-Architecture

The overall architecture (depicted in Figure 1) encompasses four main levels (i.e., information, filter, task, and interface), each being associated to a specific role. The communication between adjacent levels is achieved through suitable middle agents, which form a corresponding mid-span level.

Each level is populated by a society of agents, so that communication may occur both horizontally and vertically. The former kind of communication supports cooperation among agents belonging to a specific level, whereas the latter supports the flow of information and/or control between adjacent levels through suitable middle-agents.

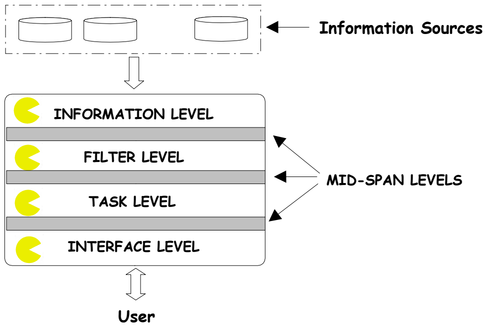


Fig. 1. The PACMAS Architecture.

Information Level: At the information level, agents are entrusted with extracting data from the information sources. Each information agent is associated to one information source, playing the role of wrapper.

Filter Level: At the filter level, agents are aimed at selecting information deemed relevant to the users, and cooperate to prevent information from being overloaded and redundant. Two filtering strategies can be adopted: generic and personal. The former applies the same rules to all users; whereas the latter is customised for a specific user.

Task Level: At the task level, agents arrange data according to users' personal needs and preferences. In a sense, they can be considered as the core of the architecture. In fact, they are devoted to achieve users' goals by cooperating together and adapting themselves to the changes of the underlying environment.

Interface Level: At the interface level, a suitable interface agent is associated with each different user interface. In fact, a user can generally interact with an application through several interfaces and devices (e.g., pc, pda, mobile phones, etc.).

Mid-span Level: At the mid-span level, agents are aimed at establishing communication among requesters and providers. In the literature, several solutions have been proposed: e.g., blackboard agents, matchmaker or yellow page agents, and broker agents (see [5] for further details). In the PACMAS architecture, agents at the mid-span level can be implemented as matchmakers or brokers, depending on the specific application.

PACMAS Micro-Architecture

Keeping in mind that agents may be classified along several ideal and primary capabilities that they should embed, in our view agents are always autonomous and flexible. Moreover, we claim that personalization, adaptation and cooperation should be taken into account as a primary feature while depicting the characteristics of software agents.

Personalization: As for personalization, an initial user profile is provided in form of a list of keywords, representing users' interests. The information about the user profile is stored by the agents belonging to the interface level. It is worth noting that, to exhibit personalization, filter and task agents may need information about the user profile. This flows up from the interface level to the other levels through the middle-span levels. In particular, agents belonging to mid-span levels (i.e., middle agents) take care of handling synchronization and avoiding potential inconsistencies. Moreover, the user behavior is tracked during the execution of the application to support explicit feedback, in order to improve her/his profile.

Adaptation: As for adaptation, a model centered on the concept of "mixtures of experts" has been employed. Each expert is implemented by an agent able to select relevant information according to an embedded string of feature-value pairs, features being selectable from an overall set of relevant features defined for the given application. The decision of adopting a subset of the available features has been taken for efficiency reasons, being conceptually equivalent to the one usually adopted in a typical GA-based environment [8], which handles also dont-care symbols. The system starts with an initial population of experts, during the evolution of the system further experts are created according to a covering, crossover, or mutation mechanism.

Cooperation: As for cooperation, agents at the same level exchange messages and/or data to achieve common goals, according to the requests made by the user. The most important form of cooperation concerns the "horizontal" control flow that occurs between peer agents. For instance, filter agents can interact in order to reduce the information overload and redundancy, whereas task agents can work together to solve problems that require social interactions to be solved.

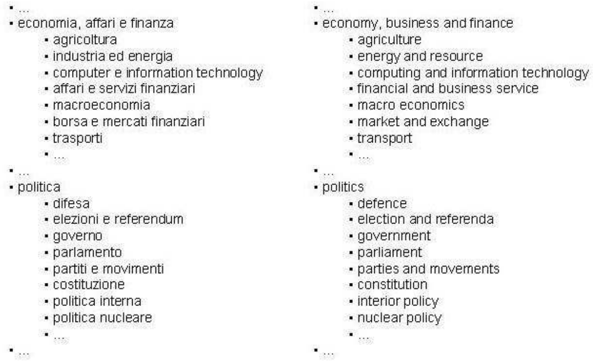
V. PACMAS FOR TEXT CATEGORIZATION

In this section, we describe how the generic architecture has been customized to implement a system to perform text categorization.

The PACMAS Levels

In the following, we illustrate how each level of the architecture supports the implementation of the proposed application.

Information Level: At the information level, agents play the role of wrappers, each one being associated to a different information source. In particular, in the current implementation a set of agents wraps databases containing italian news articles [1]. Furthermore, an agent wraps the adopted taxonomy that is a subset of the one proposed by the International Press Telecommunications Council [2] (a fragment is depicted in Figure 2).



- ...
- economia, affari e finanza
  - agricoltura
  - industria ed energia
  - computer e information technology
  - affari e servizi finanziari
  - macroeconomia
  - borsa e mercati finanziari
  - trasporti
  - ...
- ...
- politica
  - difesa
  - elezioni e referendum
  - governo
  - parlamento
  - partiti e movimenti
  - costituzione
  - politica interna
  - politica nucleare
  - ...
- ...

- ...
- economy, business and finance
  - agriculture
  - energy and resource
  - computing and information technology
  - financial and business service
  - macro economics
  - market and exchange
  - transport
  - ...
- ...
- politics
  - defence
  - election and referenda
  - government
  - parliament
  - parties and movements
  - constitution
  - interior policy
  - nuclear policy
  - ...
- ...

Fig. 2. A fragment of the adopted (italian) taxonomy and its english translation.

Information agents are not personalized, not adaptive, and not cooperative (shortly $\overline{PAC}$). Personalization is not supported at this level, since information agents are only devoted to wrap information sources. Adaptation is also not supported, since we assume that information sources are invariant for the system and are not user-dependent. Cooperation is also not supported by the information agents, since each agent retrieves information from different sources, and each information source has a specific role in the chosen application.

Filter Level: At the filter level, a population of agents manipulates the information belonging to the information level through suitable filtering strategies. First, a set of filter agents removes all non-informative words such as prepositions, conjunctions, pronouns and very common verbs by using a standard stop-word list. After removing the stop words, a set of filter agents, performs a stemming algorithm to remove the most common morphological and inflexional suffixes from all the words. Then, for each class, a set of filter agents selects the features relevant to the classification task according to the information gain method. Let us recall that information gain measures the number of bits of information obtained for category prediction by knowing the presence or absence of a term in a document.

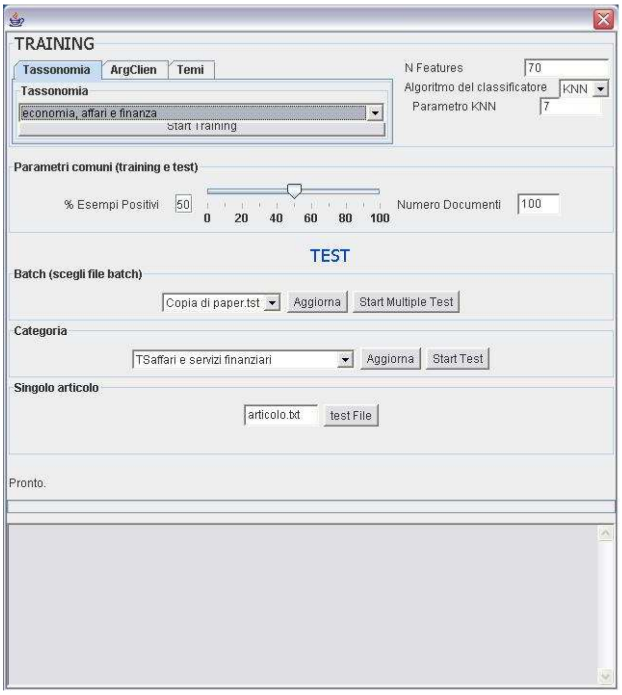
Filter agents are not personalized, not adaptive, and cooperative (shortly $\overline{PA}C$). Personalization is not supported at this level, since all the adopted filter strategies are user-independent. Adaptation is also not supported, since all the adopted strategies do not change during the agents activities. Cooperation is supported by the filter agents, since agents cooperate continously in order to perform the filtering activity.

Task Level: At the task level, a population of agents has been developed, each of them embedding a *k*NN classifier. Let us briefly recall that the *k*-nearest neighbor is a classification method based upon observable features. The algorithm selects a set which contains the *k* nearest neighbours and assigns the class label to the new data point based upon the most numerous class with the set. All the agents have been trained in order to recognize a specific class. Given a document in the test set, each agent, through its embedded *k*NN classifier, ranks its nearest neighbors among the training documents to a distance measure, and uses the most frequent category of the *k* top-ranking neighbors to predict the categories of the input document. Task agents are also devoted to measure the classification accuracy according to the confusion matrix [10].

Task agents are not personalized, adaptive, and cooperative (shortly $\overline{P}AC$). Personalization is not supported at this level,

[1] More generally, they may wrap any web site containing news (e.g., online journals).

[2] http://www.iptc.org/



TRAINING

Tassonomia | ArgClien | Temi

Tassonomia: economia, affari e finanza

N Features: 70

Algoritmo del classificatore: KNN

Parametro KNN: 7

Parametri comuni (training e test)

% Esempi Positivi: 50

Numero Documenti: 100

TEST

Batch (scegli file batch): Copia di paper.tst

Categoria: TSaffari e servizi finanziari

Singolo articolo: articolo.txt

Pronto.

Fig. 3. Interface for the news classifying system.

since, in the current implementation, the adopted classification strategies are user-independent. Adaptation is supported by the task agents since they continously adapt themselves to the underlying environment. Cooperation is supported by the task agents, since agents sometimes have to interact each other in order to achieve their own goals.

Interface Level: At the interface level, agents are aimed at interacting with the user. In the current implementation, agents and users interact through a suitable graphical interface that run on a pc. Interface agents are also devoted to handle user profile and propagate it by the intervention of middle agents.

Interface agents are personal, not adaptive, and not cooperative (shortly $P\overline{A}\,\overline{C}$). Personalization is required to allow each user the customization of her/his interface. In the current implementation, adaptation is not supported, but -at least in principle- an interface agent might adapt to the changes that occur in the preferences and interests of the corresponding user. Cooperation is not supported by agents that belong to this architectural level.

Table I summarizes the involved agents and their capabilities.

Training Task Agents

As for the training activity, task agents have been trained by a set of newspaper articles classified by human experts. Through a suitable graphical interface (see Figure 3), the user interacts with the interface agents setting her/him preferences. In particular, she/he can adjust the following parameters:

- the classification algorithm [3];

TABLE I
AGENTS ROLES AND CAPABILITIES

Agents	**The ability of ...**	**Capabilities**
information	wrapping databases containing news articles, and wrapping the taxonomy	$\overline{P}\,\overline{A}\,\overline{C}$
filter	preprocessing the documents	$\overline{P}\,\overline{A}C$
task	classifying news articles	$\overline{P}AC$
interface	interacting with the user	$P\overline{A}\,\overline{C}$
middle	allowing interactions among agents belonging to different levels	$\overline{P}\,\overline{A}C$

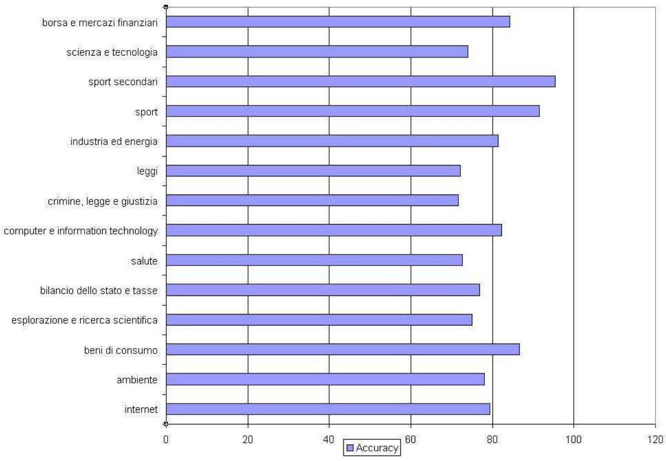


Fig. 4. Accuracy of the system.

- the number of documents forming the dataset;
- the training category;
- the percentage of positive examples;
- the number of features to be considered.

User choices are sent from the interface agent to the task level through the cooperation of the middle agent that belongs to the *task-interface* middle level (TI agent). The TI agent generates a task agent that embodies the corresponding classifier algorithm and asks it to perform the classification with the user preferences. The dataset needed for the classification is provided by information agents and subsequently pruned by the filter agents. After the classification activity, the task agent saves its own state in a suitable xml-like format in order to make it available for the test phase.

Experiments and Results

To evaluate the effectiveness of the system, several tests have been conducted using articles belonging to online newspapers. For each item of the taxonomy, a set of 200 documents has been selected to train the corresponding classifier, being *k*NN the adopted algorithm (with $k = 7$). To validate the training procedure, the system has been fed by the same dataset used in the training phase, showing an accuracy between 96% and 100%.

Then, random datasets for each category have been generated to test the performance of the system. The accuracy for fourtheen categories is summarized in Figure 4. On the average, the accuracy of the system is 80.05%. Particular

[3] in the current implementation only *k*NN is supported

care has been taken in limiting the phenomenon of “false negatives” (FN), which –nevertheless– had a limited impact on the percent of “false positives” (FP). In particular, the ratio $FN/(FN + FP)$ has been kept under 25% by weighting positive prototypes with an additional factor of 1.05 with respect to negative ones.

VI. Conclusions and Future Work

In this paper, we presented a system devoted to retrieve articles from italian online newspapers, and classify them using suitable machine learning techniques. The system has been built upon PACMAS, a generic architecture designed to support the implementation of applications explicitly tailored for information retrieval tasks. PACMAS stands for Personalized, Adaptive, and Cooperative MultiAgent Systems, since PACMAS agents are autonomous and flexible, and can be personalized, adaptive, and cooperative depending on the implemented application. The categorization capability has been evaluated using several newspaper articles, showing an average accuracy of about 80%.

As for the future work, we are extending the system to handle with an automatic composition of the categories taken from the taxonomy in order to better fit the user profile.

VII. Acknowledegments

We would like to thank Ivan Manca and Andrea Addis for participating in the development of the application.

References

[1] C. Apte, F. Damerau, and S. M. Weiss. Automated learning of decision rules for text categorization. *Information Systems*, 12(3):233–251, 1994.

[2] G. Armano, G. Cherchi, A. Manconi, and E. Vargiu. Pacmas: A personalized, adaptive, and cooperative multiagent system architecture. In *Workshop dagli Oggetti agli Agenti, Simulazione e Analisi Formale di Sistemi Complessi (WOA 2005)*, November 2005.

[3] R. Armstrong, D. Freitag, T. Joachims, and T. Mitchell. Webwatcher: A learning apprentice for the world wide web. In *AAAI Spring Symposium on Information Gathering*, pages 6–12, 1995.

[4] W. W. Cohen and Y. Singer. Context-sensitive learning methods for text categorization. In H.-P. Frei, D. Harman, P. Schaauble, and R. Wilkinson, editors, *Proceedings of SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval*, pages 307–315. ACM Press, New York, US, 1996.

[5] K. Decker, K. Sycara, and M. Williamson. Middle-agents for the internet. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI 97)*, pages 578–583, 1997.

[6] A. S. W. Erik Wiener, Jan O. Pedersen. A neural network approach to topic spotting. In *Proceedings of 4th Annual Symposium on Document Analysis and Information Retrieval*, pages 317–332, Las Vegas, US, 1995.

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

[8] D. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

[9] C. A. Knoblock, Y. Arens, and C.-N. Hsu. Cooperating agents for information retrieval. In *Proceedings of the Second International Conference on Cooperative Information Systems*, Toronto, Ontario, Canada, 1994. University of Toronto Press.

[10] R. Kohavi and F. Provost. Glossary of terms. *Special issue on applications of machine learning and the knowledge discovery process, Machine Learning*, 30(2/3):271–274, 1998.

[11] J. Kramer. Agent based personalized information retrieval, 1997.

[12] D. D. Lewis and M. Ringuette. A comparison of two learning algorithms for text categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 81–93, Las Vegas, US, 1994.

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

[14] I. Moulinier, G. Raskinis, and J.-G. Ganascia. Text categorization: a symbolic approach. In *Proceedings of 5th Annual Symposium on Document Analysis and Information Retrieval*, pages 87–99, Las Vegas, US, 1996.

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

[16] B. Sheth and P. Maes. Evolving agents for personalized information filtering. In I. Press, editor, *9th Conference on Artificial Intelligence for Applications (CAIA-93)*, pages 345–352, 2003.

[17] K. Tzeras and S. Hartmann. Automatic indexing based on Bayesian inference networks. In R. Korfhage, E. Rasmussen, and P. Willett, editors, *Proceedings of SIGIR-93, 16th ACM International Conference on Research and Development in Information Retrieval*, pages 22–34, Pittsburgh, US, 1993. ACM Press, New York, US.

[18] Y. Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1/2):69–90, 1999.

[19] Y. Yang and C. Chute. An example-based mapping method for text categorization and retrieval. *ACM Transactions on Information Systems*, 12(3):252–277, 1994.

[20] Y. Yang and X. Liu. A re-examination of text categorization methods. In M. A. Hearst, F. Gey, and R. Tong, editors, *Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval*, pages 42–49, Berkeley, US, 1999. ACM Press, New York, US.

[21] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *International Conference on Machine Learning*, pages 412–420, 1997.