Personalized Text Categorization Using a MultiAgent Architecture

Andrea Addis, Giuliano Armano, Giancarlo Cherchi, and Eloisa Vargiu
Department of Electrical and Electronic Engineering, University of Cagliari
Piazza d'Armi, I-09123
Cagliari, Italy
{addis, armano, cherchi, vargiu}@diee.unica.it

ABSTRACT

In this paper, a system able to retrieve contents deemed relevant for the users through a text categorization process, is presented. The system is built exploiting a generic multiagent architecture that supports the implementation of applications aimed at (i) retrieving heterogeneous data spread among different sources (e.g., generic html pages, news, blogs, forums, and databases); (ii) filtering and organizing them according to personal interests explicitly stated by each user; (iii) providing adaptation techniques to improve and refine throughout time the profile of each selected user. In particular, the implemented multiagent system creates personalized press-revies from online newspapers. Preliminary results are encouraging and highlight the effectiveness of the approach.

1. INTRODUCTION

We are assisting to a continuos growth in the availability of electronically stored information, due to telecommunications developments and cost reductions in technology. In particular, the World Wide Web offers a massive amount of data coming from different and heterogeneous sources. Unfortunately, it is becoming very difficult 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, and blogs). 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 results provided by the system. Therefore, is becoming a primary issue to support users in handling with this enormous and widespread amount of web information. To this aim, 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, several approaches have been proposed to separately face with information extraction and text categorization. As for information extraction, several tools have been proposed to better address the issue of generating wrappers for web data extraction [18]. Such tools are based on several distinct techniques such as declarative languages [7, 13], HTML structure analysis [8, 23], natural language processing [11, 25], machine learning [14, 16], data modeling [1, 22], and ontologies [10]. As for text categorization, several machine learning techniques have been exploited [29]. Let us recall multivariant regression models [28], k-Nearest Neighbor (k-NN) classification [30], Bayes probabilistic approaches [26, 31], decision trees [19], artificial neural networks (ANN) [27], symbolic rule learning [20] and inductive learning algorithms [5].

In this paper, we propose a multiagent system suitably tailored for generating press reviews by (i) extracting articles from italian online newspapers, (ii) classifying them using text categorization according to user's preferences, and (iii) providing suitable feedback mechanisms. In particular, we focus on the text categorization techniques adopted by the proposed multiagent system. The motivation for adopting a multiagent system lies in the fact that a centralized classification system may be overwhelmed by a large and dynamic document stream such as daily-updated online news [12]. Moreover, the Internet is intrinsecally a distributed system and offers the opportunity to take advantage of distributed computing paradigms and distributed knowledge resources for classification. From a conceptual point of view, the proposed system creates press reviews in separate steps at different levels of granularity, whereas from a technological point of view the system has been implemented upon the PACMAS architecture [3], thus giving rise to a personalized, adaptive, and cooperative multiagent system.

The remainder of the paper is organized as follows: Section 2 illustrates the multiagent system from both the abstract and the concrete perspective. Section 3 discusses preliminary experimental results. Section 4 draws conclusions and points to future work.

2. THE PROPOSED MULTIAGENT SYSTEM

In this section, we present a multiagent system suitably tailored for creating personalized press reviews. From a conceptual point of view, the system is organized into three logical layers – each one devoted to a specific task; from a technological point of view the system has been built upon the PACMAS architecture [3] – giving rise to a personalized, adaptive, and cooperative multiagent system.

AAMAS'06 May 8–12 2006, Hakodate, Hokkaido, Japan.

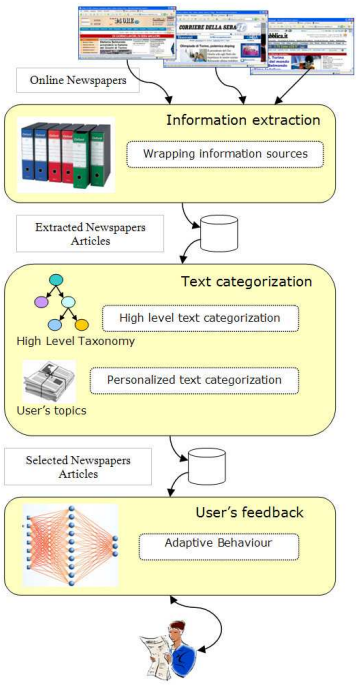

Figure 1: The Proposed Architecture.

2.1 The Abstract Architecture

Automatically generating personalized press reviews typically involves three main activities: (i) extracting the required information, (ii) classifying them according to users preferences, and (iii) providing suitable feedback mechanisms to improve the overall performances. Furthermore, personalization and adaptation should be taken into account, in order to allow users to set their preferences in advance and provide their feedback while the system is running. A generic architecture able to perform these activities is presented in Figure 1. This section illustrates each involved activity with particular emphasis on text categorization.

2.1.1 Information Extraction

The information extraction module extracts data from web sources through specialized wrappers. Each wrapper identifies data of interests and maps them according to a suitable function M. Given a web page P containing a set of objects $O = \{o_1, ..., o_n\}$, a mapping function $M : P \rightarrow O$ populates a data repository R with the elements of O. In particular, objects in O are: text content, title, half title, author/s, and figure captions. Furthermore, M is capable of recognizing and extracting data from any other page P' similar to P (i.e., pages provided by the same site or a web service).

Currently, two kinds of wrappers have been implemented, depending on the supported Internet sources: the *HTML / XHTML* and the *RSS* wrapper, respectively.

The former kind of wrapper extracts information by directly parsing Internet pages in HTML format. Let us point out that HTML is often bad-formed and thus needs ad-hoc algorithms to be correctly parsed. Therefore, the process of extracting data from HTML pages typically consists of two steps: (i) learning page structure; (ii) performing structured data extraction. The first step, currently supervised, allows the wrapper to detect the tags containing objects in the set O. The second step consists of applying the mapping function to populate the corresponding data repository.

The latter kind of wrapper extracts information from online newspapers articles in RSS format [1]. Being the RSS a well-structured format, it is very simple to process RSS pages, since RSS tags contain objects in O.

2.1.2 Text Categorization

The text categorization module progressively filters information that flows from external sources (i.e., online newspapers) to the end user by retaining only the relevant articles. First, newspapers articles are classified according to a high-level taxonomy, which is independent from the specific user. Being interested in classifying newspaper articles, we adopted the taxonomy proposed by the International Press Telecommunications Council [2] (a fragment is depicted in Figure 2).

Figure 2: A fragment of the adopted taxonomy (in italian and in english).

The corresponding classifiers, devised to perform text categorization, have been trained by resorting to state-of-the-art algorithms that implement the k-NN technique, in its "weighted" variant [6]. The choice of adopting this particular technique stems from the fact that it does not require specific training and is very robust wrt the impact of noisy data. To facilitate the work of such experts, a suitable encoding has been adopted. In particular, 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 [21] removes the most common morphological and inflexional suffixes. Then, for each class of the taxonomy, features selection, based on the information gain statistics, has been enforced to reduce the dimensionality of the feature space.

Text categorization is performed according to the user needs and preferences. In fact, typically, the user is not directly concerned with "generic" 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 defense *and* government. This "compound" topic can be dealt with by composing the

[1] Really Simple Syndication
[2] http://www.iptc.org/

defense and the *government* classifiers [3].

Particular care has been taken in limiting the phenomenon of "false negatives" (FN), which –nevertheless– produced the effect of augmenting the percent of "false positives" (FP). To reduce the impact of this latter, unwanted, effect, we exploit the filtering effect enforced by the combination of different classifier outputs, together with the existence of a suitable taxonomy. Let us consider both the "horizontal" and the "vertical" way of combining classifiers. The former occurs according to the typical linguistic interpretation of the logical connectives "and", "or", and "not". In text categorization, the most important connective is the first, since the remaining ones can be more easily dealt with after giving a suitable semantics to the first. Hence, let us concentrate on how to cope with an "and-based" combination of classifier outputs. There are several ways of combining them, according to the user's needs, from the standard (and trivial) logical "and" to more sophisticate tools. In particular, we adopted a –rather general– soft boolean perspective, in which the combination is evaluated using P-norms, with the "p" parameter set to 5 (we did not observe significant differences for $p \geq 5$). The latter way of combining classifier outputs occurs within a taxonomy of classifiers, and consists of exploiting the effect of a typical pipeline of classifiers that progressively filter out non relevant information according to their level of granularity.

For the sake of simplicity, let us assimilate horizontal and vertical combination of classifiers. [4] In particular, we expect that most articles are non relevant to the user, the ratio between negative and positive examples being very high (a typical order of magnitude is 10^3). Unfortunately, this aspect has a very negative impact on both precision and recall. On the other hand, combining classifiers allows to reduce this negative effect –in the best case exponentially with respect to the number of classifiers that occur in the combination. Experimental results confirm this hypothesis, although the actual impact of combination is not as high as the theoretical one, due to the existing correlation between the classifiers actually involved in the combination. Nevertheless, a combination of 5-7 classifiers (e.g., 3-4 combined horizontally and 2-3 in pipeline) greatly helps to reduce the given problem.

2.1.3 User's Feedback

The user's feedback module is devoted to deal with any feedback optionally provided by the end-user. Several solutions have been experimented to deal with the problem of supporting user's feedback, although how to improve this part of the architecture is considered priority with respect to other issues. So far, two trivial though effective solutions have been implemented and experimented, based on the neural and k-NN technology. The former solution consists of training an ANN with a set of examples classified as "of interest to the user" by the second layer. When the amount of feedback provided by the user has trespassed a given threshold, the ANN is trained again –after updating the previous training set with the information provided by the user. The latter solution consists of a k-NN classifier. When a non-interesting article is evidenced by the user, it is immediately embedded in the training set of the k-NN classifier. Of course, a suitable 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, some examples are randomly extracted from the set of "true" positive examples and embedded in the above training set. The solution based on the k-NN technology has shown to be slightly better than the one based on ANNs, although this result should be validated by further and more detailed experiments.

[3]The possibility of resorting to other, more specific, solutions is left to the knowledge engineer who is in charge of maintaining the taxonomy. If s/he deems that the user's interests are too difficult to obtain through composition, the alternative solution would consist of training a specific classifier.

[4]Which is sound, provided that the correlation between pipelined classifiers does not exceed a reasonable threshold.

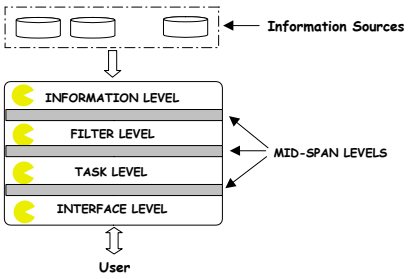

Figure 3: The PACMAS Architecture.

2.2 The Concrete Architecture

The functionalities of the abstract architecture, described in the previous section, have been implemented exploiting the PACMAS architecture. In this section, after briefly recalling the PACMAS architecture, all customizations made to create press reviews are described.

2.2.1 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. The PACMAS architecture (depicted in Figure 3) 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.

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. 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. 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. At the interface level, a suitable interface agent is associated with each different user interface. In fact, a user can

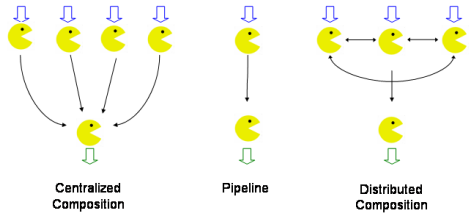

Figure 4: Agents Connections.

generally interact with an application through several interfaces and devices (e.g., pc, pda, mobile phones, etc.). At the mid-span level, agents are aimed at establishing communication among requesters and providers. Agents at these architectural levels can be implemented as matchmakers or brokers, depending on the specific application [9].

PACMAS agents can be personalized, adaptive, and cooperative, depending on their specific role. As for personalization, an initial user profile is provided to represent users' interests. The information about the user profile is stored by the interface agents, and 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. As for adaptation, different techniques may be employed depending on the application to be developed. In particular, the user behavior is tracked during the execution of the application to support explicit feedback in order to improve her/his profile as well as the system performances. 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. Cooperation is implemented in accordance with the following modes: centralized composition, pipeline, and distributed composition (see Figure 4). In particular: (i) centralized compositions can be used for integrating different capabilities, so that the resulting behavior actually depends on the combination activity; (ii) pipelines can be used to distribute information at different levels of abstraction, so that data can be increasingly refined and adapted to the user's needs; and (iii) distributed compositions can be used to model a cooperation among the involved components aimed at processing interlaced information.

2.2.2 PACMAS for Text Categorization

In this section, we describe how the generic architecture has been customized to implement a prototype of the system devoted to create press reviews. In particular, we illustrate how each level of PACMAS supports the implementation of the proposed application. The prototype has been implemented using JADE [4] as the underlying framework.

2.2.2.1 *Information Level.*

The agents at this architectural level are devoted to perform the information extraction. In particular, in the current implementation a set of agents wraps italian online newspapers containing newspapers articles in RSS and HTML format. In particular, an agent wraps the adopted "generic" taxonomy to be used during the high-level text categorization phase. 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.

2.2.2.2 *Filter Level.*

At the filter level, a population of agents processes the information belonging to the information level through suitable filtering strategies preparing the information for the text categorization phase. 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. Filter agents are not personalized, not adaptive, and cooperative (shortly $\overline{PAC}$). 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 continuously in order to perform the filtering activity according to the pipeline mode.

2.2.2.3 *Task Level.*

At the task level, a population of agents has been developed. Such agents are devoted to perform the text categorization activities. To perform the high-level text categorization activity, each task agent embodies a kNN classifier. All the involved agents have been trained in order to recognize a specific class. Given a document in the test set, each agent, through its embedded classifier, ranks its nearest neighbors among the training documents to a distance measure, and uses the most frequent category of the k top-ranking neighbours to predict the categories of the input document. Each task agent is also devoted to measure the classification accuracy according to the confusion matrix [15]. To perform the personalized text categorization activity, some agents at this architectural level are devoted to take into account users preferences automatically composing topics. Composition has been performed through the cooperation of the involved task agents. For instance, the "compound topic" defense *and* government is obtained by the cooperation of the task agent expert in recognizing *defense* together with the task agent expert in recognizing *government*. Task agents are personalized, not adaptive, and cooperative (shortly $P\overline{AC}$). Personalization is supported by the task agents, since they perform the classification taking into account users needs and preferences. Adaptation is not supported by the task agents since all the adopted strategies do not change during the agents activities. Cooperation is supported by the task agents, since agents have to interact each other in order to achieve the classification task.

2.2.2.4 *Interface Level.*

At the interface level, agents are aimed at interacting with

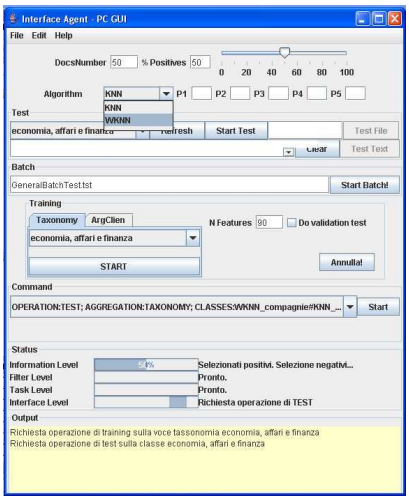

Figure 5: Interface for the news classifying system.

the user. In the current implementation, agents and users interact through a suitable graphical interface that runs on a pc (see Figure 5. Interface agents are also devoted to handle user profile and propagate it by the intervention of middle agents. Interacting with the interface agent, the user sets her/his preferences. In particular, s/he can set preferences regarding on the information sources, and the the topics of the required press review. Moreover, the interface agent is also devoted to deal with the feedback provided by the user.

Interface agents are personal, adaptive, and not cooperative (shortly $PA\overline{C}$). Personalization is supported to allow each user the customization of her/his interface. Adaptation is supported to take into account the user's feedback. Cooperation is not supported by agents that belong to this architectural level.

3. EXPERIMENTAL RESULTS

To evaluate the effectiveness of the system, several tests have been conducted using articles belonging to the selected online newspapers. [5]

First, several experiments have been performed to set the optimal parameters for the training activity. In particular, experiments have been conducted changing the following parameters: the number of documents forming the dataset; the percentage of positive examples; the number of features to be considered.

As for the training activity, [6] task agents have been provided with a set of newspaper articles previously classified by experts of the domain. For each item of the taxonomy, a set of 200 documents has been selected to train the corresponding classifier. Subsequently, to validate the training procedure of the first step of classification, 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 global accuracy for fourteen categories is summarized in Figure 6.

[5]In the current implementation, www.repubblica.it, www.corriere.it, and www.espressonline.it

[6]In fact, training for k-NN classifiers typically consists of storing known examples.

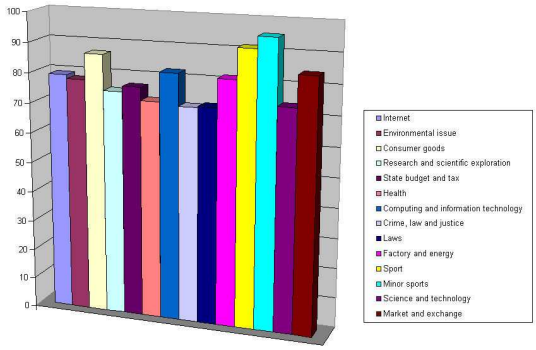

Figure 6: Accuracy of the system.

On the average, the accuracy of the system is 80.05%. It is worth pointing out that, in this specific task, the accuracy should not be directly considered as a measure of the system performance. On the other hand, it becomes important since the accuracy of a classifier (evaluated on a balanced test set, i.e., with a number of negative examples that does not differ much from the number of positive ones) indirectly affects the recall, under the hypothesis that classifiers are (dynamically) combined using logical operators and/or (statically) combined according to the given taxonomy (in this latter case, they are in fact in a pipeline).

As for the composition of taxonomy items, once trained the task agents, several experiments have been performed to test the performance of the system. The existence of a classifiers' taxonomy and the ability of resorting to combinations of classifiers allowed to reach a recall comparable with state-of-the-art systems by resorting to the composition of three classifiers (on average) and within a taxonomy with depth three. This result has been obtained by imposing a ratio between negative and positive instances of 10^2 and with an accuracy measured on single classifiers tested with an equal number of negative and positive examples (on average) between 90 and 95%. The categorization capability has been evaluated using several newspaper articles previously classified by hand by domain experts.

Results are very encouraging and show that the proposed approach is effective in the given application task, also taking into account that the system can be improved in several and important aspects.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a system devoted to retrieve articles from italian online newspapers and to 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, as PACMAS agents are autonomous and flexible, and can be personalized, adaptive, and cooperative –depending on their role within the given application.

As for the future work, we are implementing a new release of the system with improved text categorization functionalities by adopting different classifier algorithms, such as naive bayesian classifier. Furthermore, we are investigating how to enhance the feedback-related functionalities according to an

evolutionary computation framework. Finally, a graphical interface to compose items of the taxonomy is under study.

5. ACKNOWLEDEGMENTS

We would like to thank Ivan Manca for participating in the development of the system.

6. REFERENCES

[1] Brad Adelberg, 'NoDoSE–a tool for semi-automatically extracting structured and semistructured data from text documents', pp. 283–294, (1998).

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

[3] 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.

[4] F. Bellifemine, A. Poggi, and G. Rimassa. Developing multi-agent systems with jade. In *Eventh International Workshop on Agent Theories, Architectures, and Languages (ATAL-2000)*, 2000.

[5] 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.

[6] WS. Cost and S. Salzberg. A weighted nearest neighbor algorithm for learning with symbolic features. In Machine Learning, Vol. 10, 1993, pp. 57-78, 1993.

[7] V. Crescenzi and G. Mecca, 'Grammars have exceptions', *Information Systems*, **23**(8), 539–565, (1998).

[8] Valter Crescenzi, Giansalvatore Mecca, and Paolo Merialdo, 'Roadrunner: Towards automatic data extraction from large web sites', in *Proceedings of 27th International Conference on Very Large Data Bases*, pp. 109–118, (2001).

[9] 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.

[10] David W. Embley, Douglas M. Campbell, Y. S. Jiang, Stephen W. Liddle, Yiu-Kai Ng, Dallan Quass, and Randy D. Smith, 'Conceptual-model-based data extraction from multiple-record web pages', *Data Knowledge Engineering*, **31**(3), 227–251, (1999).

[11] Dayne Freitag, *Machine Learning for Information Extraction in Informal Domains*, Ph.D. dissertation, Carnegie Mellon University, 1998.

[12] Yueyu Fu, Weimao Ke, and Javed Mostafa, 'Automated text classification using a multi-agent framework.', in *JCDL*, pp. 157–158, 2005.

[13] Joachim Hammer, Héctor García-Molina, Svetlozar Nestorov, Ramana Yerneni, Marcus Breunig, and Vasilis Vassalos, 'Template-based wrappers in the TSIMMIS system', pp. 532–535, (1997).

[14] Chun-Nan Hsu and Ming-Tzung Dung, 'Generating finite-state transducers for semi-structured data extraction from the web', *Information Systems*, **23**(8), 521–538, (1998).

[15] 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.

[16] Nicholas Kushmerick, 'Wrapper induction: Efficiency and expressiveness', *Artificial Intelligence*, **118**(1-2), 15–68, (2000).

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

[18] Alberto H. F. Laender, Berthier A. Ribeiro-Neto, Altigran S. da Silva, and Juliana S. Teixeira, 'A brief survey of web data extraction tools', *SIGMOD Rec.*, **31**(2), 84–93, (2002).

[19] 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.

[20] 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.

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

[22] Berthier A. Ribeiro-Neto, Alberto H. F. Laender, and Altigran Soares da Silva, 'Extracting semi-structured data through examples', in *CIKM*, pp. 94–101, (1999).

[23] Arnaud Sahuguet and Fabien Azavant, 'Building intelligent web applications using lightweight wrappers', *Data Knowledge Engineering*, **36**(3), 283–316, (2001).

[24] 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.

[25] Stephen Soderland, 'Learning information extraction rules for semi-structured and free text', *Machine Learning*, **34**(1-3), 233–272, (1999).

[26] 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.

[27] 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.

[28] 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.

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

[30] 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.

[31] Z. Zheng. Naive Bayesian Classifier Committees. In *ECML'98*, pages 196–207, LNAI 1398, 1998.