Towards Automatic Digital Library Content-based Management

F. Esposito, S. Ferilli, N. Di Mauro, T.M.A. Basile

Dipartimento di Informatica

Università degli Studi di Bari

Abstract

Knowledge distribution and preservation caused automated document management to become a relevant research issue in Computer Science literature. Indeed, computers can offer a significant support to document management, indexing and retrieval based on meta-information that represents the knowledge they contain in a machine-understandable and processable format. This poses several sub-problems, that range from the analysis of the document layout, to the identification of the document type and of the role played by its components, up to the exploitation of techniques for document content indexing. Nowadays, documents are composed and produced directly in electronic form, of which the paper printout is just a facility that improves physical handling and readability. Hence, a whole research area focused on principles and techniques for setting up and managing document collections in the form of Digital Libraries has started and quickly developed.

This work presents the current version of the prototype **DOMINUS** (DOcument Management INtelligent Universal System), a system for automated electronic documents processing characterized by the intensive exploitation of intelligent techniques in all the steps involved from document acquisition to document indexing for categorization and information retrieval purposes. It can currently deal with documents in standard formats, such as PostScript (PS) or its evolution Portable Document Format (PDF). Since the system is general and flexible, it can be embedded as a document management engine into many different domain-specific applications. Here, we focus on the Conference Management domain, and show how DOMINUS can usefully support some of the more critical and knowledge-intensive tasks involved by the organization of a scientific conference, such as the assignment of the submitted papers to suitable reviewers.

The various document processing steps performed by DOMINUS to go from the original PDF/PS document to the text extraction and indexing are as follows.

1. *Layout analysis*: encompasses various sub-processes:
   1. The incoming document is pre-processed by a module called **WINE** (Wrapper for the Identification of Non-uniform Electronic document formats), that takes a PDF/PS document and produces the initial document's XML basic representation, that describes it as a set of pages made up of basic blocks.
   2. Basic blocks, often corresponding to fragment of words, are first aggregated based on their overlapping or adjacency into composite blocks surrounding whole words. Then words are grouped into lines according to proper parameters that avoid merging lines that belong to different columns; specifically, such parameters were automatically learned by casting the task as a Multiple Instance Problem and exploiting the algorithm proposed in [Dietterich97].
   3. Semantically related blocks are grouped into frames based on whitespace and background structure analysis by **DOC** (Document Organization Composer), a variant of Breuel's algorithm [Breuel02], that finds iteratively the maximal white rectangles in a page and then obtains the content areas as their complement. The modification consisted in the identification of empirical criteria that avoid the system to find insignificant white spaces such as inter-word or inter-line ones.
   4. *Layout Correction*: If the automatically recognized layout structure is affected by imperfections, a phase of layout correction is needed, that is automatically performed by a module called **CELLAR** (Correction Engine for Low-quality Layout Analysis Results). It applies embedded rules automatically learned by the first-order logic system **INTHELEX** [Esposito03] for this task. To this purpose, the module **IGT** (Interface for Graphic layout Transformation) collects the manual corrections performed by users on sample documents and describes them in a first-order language representing both the situations *before* and *after* the manual correction.
   5. The final document layout is described in a first-order language by a module called **DOCG** (Description of Organized Content Generator), so that INTHELEX can be applied to learn and exploit rules for identifying the document class and the role of its components.
2. *Classification*: associates the document to a class that expresses its type (e.g., scientific/newspaper article, etc.). INTHELEX is also exploited to learn rules for the automatic identification of the class.

3. *Understanding*: since the logical structure depends on the kind of document, classification of the document is a preliminary step to know which components are to be expected for that document (e.g., a sender is significant for a mail but non for a newspaper article). The understanding step identifies the significant layout components for the class previously recognized and associates to each of them a tag that expresses its role (e.g., title, author, abstract, etc.). Again, INTHELEX is exploited to learn rules for the automatic identification of the logical components.
4. *Text extraction*: Extracts the text from the significant components.
5. *Indexing*: Exploits the Latent Semantic Indexing technique to index the documents [Deerwester90] according to their semantic content.

The following scenario can give an idea of how **DOMINUS** can be exploited in some phases of an automatic conference management system, and of what advantages it can bring to the involved people. An Author connects to the Internet and (after registering, or after logging in if already registered) opens the submission page, where he can browse his hard disk and submit a paper by choosing the corresponding file in one of the accepted formats. The paper is received and undergoes the various processing steps. The layout analysis algorithm is applied, in order to single out its layout components. Then, it is translated into a first-order logic description and classified by a proper module according to the theory learned so far for the acceptable submission layout standards (e.g., full paper, poster, demo). Depending on the identified class, a further step exploits the same description to locate and label the layout components of interest for that class. In our case, we believe that title, author, abstract and references in a full paper are the most significant components to be exploited for indexing puroses, since we assume they compactly summarize the subject and research field the paper is concerned with. The text that makes up each of such components is read, stored and used to automatically file the submission record (e.g., by filling its title, authors and abstract fields).

If the system is unable to carry out any of these steps, such an event is notified to the Conference administrators, that can manually fix the problem and let the system complete its task. Such manual corrections are logged and used by the incremental learning component to refine the available classification/labelling theories in order to improve their performance on future submissions. Alternatively, the fixes can be logged and exploited all at once to refine the theory when its performance falls below a given threshold. Nevertheless, this is done off-line, and the updated theory replaces the old one only after the learning step has been successfully completed: this allows further submissions to take place in the meantime, and makes the refinement step transparent to the Authors. Successively a categorization of the paper content according to the text read is performed, with the purpose of allowing its content-based retrieval in future exploitations of the documents dataset.

One of such exploitations consists in matching the paper topics against the reviewers' expertise, in order to find the best associations for the final assignment. **GRAPE** (Global Review Assignment Processing Engine) [DiMauro05] is an expert system for solving the reviewers assignment problem, that takes advantage of both the papers content (topics) and the reviewers expertise and preferences (biddings). It could be used by exploiting, in addition to the papers topics, the reviewers expertise only, or both the reviewers expertise and biddings. Two measures were defined to guide the system during the search of the best solutions: the *reviewer's gratification* (based on the topics of the articles assigned to him and on the quantity of assigned articles that he had bid) and the *article's coverage* (based on the paper topics their reviewers are expert in and on the global experience of the reviewers). The assignment process is carried out in two phases. In the former, the system makes a tentative assignment based on paper/reviewer topics only; then, the results are presented as suggestions to the reviewers during the bidding phase; lastly, after the reviewers' biddings are collected, the system is run again to come up with a new assignment that takes into account both the topics and the bidding preferences.

References

[Breuel02] T.M. Breuel, “Two geometric algorithms for layout analysis,” Proc. Workshop on Document Analysis Systems, 2002.

[Deerwester90] S. Deerwester et al., “Indexing by Latent Semantic Analysis,” Journal of the American Society of Information Science, vol. 41, no. 6, 1990, pp. 391-407.

[DiMauro05] N. Di Mauro, T.M.A. Basile and S. Ferilli, “GRAPE: An Expert Review Assignment Component for Scientific Conference Management Systems,” Innovations in Applied Artificial Intelligence , F. Esposito and M. Ali, eds., LNCS 3533, Springer, 2005, pp. 789-798.

[Dietterich97] T.G. Dietterich, R.H. Lathrop, and T. Lozano-Perez, “Solving the multiple instance problem with axis-parallel rectangles,” Artificial Intelligence, vol. 89, no. 1/2, 1997, pp. 31-71.

[Esposito03] F. Esposito et al., “Incremental multistrategy learning for document processing,” Applied Artificial Intelligence: An International Journal, vol. 17, no. 8/9, 2003, pp. 859-883.