Lexisla: a Legislative Information Retrieval System*

Lexisla: un sistema de Recuperación de Información Legislativa

Ismael Hasan
IRLab, ICT Centre
Campus Elviña s/n
A Coruña
ihasan@udc.es

Javier Parapar
IRLab, A Coruña Univ.
Campus Elviña s/n
A Coruña
javierparapar@udc.es

Álvaro Barreiro
IRLab, A Coruña Univ.
Campus Elviña s/n
A Coruña
barreiro@udc.es

Resumen: Cada día se publican nuevos documentos legislativos en Internet que contienen cambios en la legislación: leyes, decisiones, resoluciones, etc. *Lexisla* pretende ofrecer acceso a esta información mediante una única aplicación de búsqueda que recupera, analiza y segmenta los documentos legislativos publicados diariamente.
Palabras clave: Sistema de búsqueda legislativa, segmentación de documentos, extracción de texto

Abstract: New legislative documents are published everyday in the Internet, comprising changes in the legislation: laws, decisions, resolutions, etc. *Lexisla* intends to offer access to this information through a single search application which crawls, analyses and segments the daily published legislative documents.
Keywords: Legislation search system, documents' segmentation, text extraction

1. Introduction

In the last years, the growth of Internet has favoured the use of electronic documents. Public administrations offer the printed documentation they generate also in an electronic way, being PDF the most used format. Legislative publications are a particular case. This kind of documents is produced on a daily basis, and they are supplied from the publishers web pages. The information they cover is useful for a wide variety of Internet users, being the most representative the lawyers community. Also, enterprises can use the information of the legal documents: a new regulation may affect their business model, for instance. Finally, the third target group of the legal information is the whole citizenship of a country: the documents can contain notifications to concrete people, important official dates, etc.

However, despite the fact that this information is very valuable, to search over it is a hard task: the official publishers offer search engines to access the information, but each one of those search engines offers access only to one source of documents. There are also commercial applications offering searches over several bulletins, but the results they return are not fully satisfactory (for instance, some applications search only over the summaries of the documents). In this work we present *Lexisla*[1], a system that offers searches over several different legislative publications; moreover, the information is processed and analysed, so a document (which can contain hundreds of pages) is segmented into the legislative units (resolutions, notifications, etc) it comprises. Also, the information of these units is analysed so the final users of the system can make complex searches over the information, including titles, publisher organisms, etc.

2. System Overview

Lexisla is a Web Application for accessing the legislation periodically published in the online official sources. It is divided into two subsystems, the user's application, offering searches over the information maintained by the system, and the management application, to manage the information. At the present moment, the information processed and maintained by the system comprises European and Spanish official bulletins, and several Spanish regional bulletins. The system allows the addition of new sources of official bulletins through the management applica-

* This work was funded by FEDER, *Ministerio de Ciencia e Innovación* and *Xunta de Galicia* under projects TIN2008-06566-C04-04 and 07SIN005206PR.

[1] An operative version for registered users is available from www.irlab.org/lexisla. An evaluation account can be requested at irlab@udc.es

tion. Also, the administrator of the system can schedule when the system automatically crawls new information from a source, and can create and assign search profiles to the users.

The documents automatically downloaded by the application are processed in the following way: first, it is obtained the text from each document in reading order (this task is specially difficult with PDF documents), next, the text is segmented into the legislative units it contains. Finally, these units are analysed and segmented. For each of them the following fields are stored: body, title, publisher organism, date, document and source, page numbers of the unit and type of the legislative unit (resolution, notification, etc).

This processing of the documents provides the users with the following features: results display in the web browser, download of the pages of a document containing an information unit, documents browsing ("Which are the resolutions of this document?") and advanced search features, like searching only in a few sources, searching in certain specific fields and use of regular expressions to search, so *Lexisla* can offer a wide range of available searches.

3. System Architecture

Lexisla was designed as a Model-View-Controller web application, following a component-based architecture. The most relevant components of the application are explained next, followed by an explanation about the data storage of *Lexisla*.

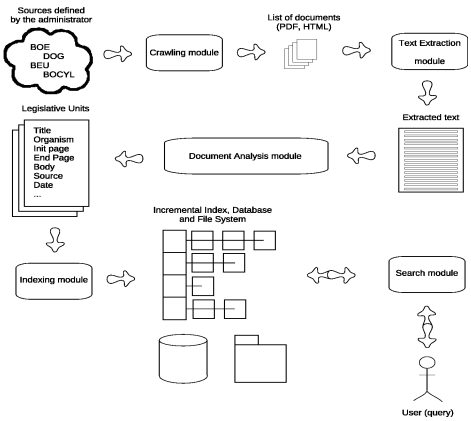

Figure 1: *Lexisla* Model Architecture

3.1. Crawling module

It accomplishes the access to the web pages of the publisher organisms and administrations (defined by the administrator of the application) and downloads all of the relevant documents.

3.2. Text Extraction module

PDF is the most usual format to distribute electronic documents. Currently, there are several tools to extract the text from this type of documents (Apache Software Foundation, 2010; Phelps, 2010), but it is very usual that this text contains errors: for instance, the paragraphs of a page may be returned disordered. This issue penalises the analysis of the information, so *Lexisla* contains its own text extraction module to bypass this problem, specially designed to accomplish this task.

3.3. Document Analysis module

This component processes the texts obtained by the Text Extraction module, and extracts all of the information contained inside each document. It also analyses each legislative unit to get its fields (title, publisher organism, start and end page, body, etc).

3.4. Indexing and Search modules

These components use incremental indexes to store the legislative units. They use IR algorithms for processing queries against inverted indexes, assuring an efficient and effective search.

3.5. Information storage

To satisfy the users information needs, the information is stored in three different systems. Integrity of references is maintained between the systems.

- Search index: contains information about the legislative units.
- Database: contains information about users, configuration, search profiles, documents, etc.
- File system: original documents.

4. Research Issues

Lexisla is an IR system that uses state-of-art algorithms and techniques for crawling, extraction of text, segmentation of information and search. In this section of the paper we will explain some of the most relevant research issues.

4.1. Extraction of Ordered Text

As explained earlier, one of the challenges is the extraction of correct ordered text from PDF documents. For this purpose, we developed a method which simulates the human reading order to obtain the text from documents. For each page, it works as follows:

1. Detection of the rectangular text regions which are present in the page.
2. Retrieval of the list of images and creation of regions using the images coordinates.
3. Split of the text regions which are crossed by image regions.
4. Sorting of the regions of a page in the following way:
   a) The region comprising the header of the page.
   b) The left top region.
   c) The regions on the right of the one obtained in (b).
   d) The region on the left of the page which is below the previously found regions.
   e) The regions on the right of the one obtained in (d).
   f) Steps (d) and (e) are repeated until no more regions are found.
5. Extraction of the text of each region, in the order stated in (4).

It is worthy to mention that *Lexisla* also deals with language identification issues. A legislative document can contain text in different languages: for instance, "Boletín Oficial del Territorio Histórico de Álava" contains sections in which the text in the left columns is written in Basque and the same text appears translated to Spanish in the right ones. *Lexisla* identifies the language of each region so in the result of the text extraction only the text in one language is returned.

To evaluate our method, its results where compared against the results obtained with PDFBox and an implementation of the XY-cuts algorithm (Mao and Kanungo, 2001); our method is coined as "LRE" (Left Regions Expansion). The metric used was the ratio of pages correctly extracted. The dataset comprises documents from the European Union (BEU, OJEU), America (FR), United Kingdom (UK), France (JO) and Spain (BOE, DOG, BOCYL). Our algorithm greatly outperformed XY-cuts, although it is fair to say that XY-cuts was not designed for this task. But, our algorithm also outperformed PDFBox: the overall ratio of pages correctly extracted with LRE was 96 %, and the overall ratio with PDFBox was 87 %. The difference of the mean between LRE and PDFBox is statistically significant, according to the Wilcoxon test ($p < 0{,}05$).

4.2. Documents Segmentation

Legal documents can contain a lot of resolutions, communications, etc. An user searching through a LegalIR system does not expect to receive complete documents as a response for a query; instead, he expects that the results are single information units. So, there is a need of segmentation of these full documents. It follows a briefing of the analysis process of documents in *Lexisla*, which uses a specialised lexicon. A extended version of this summary can be found in the work of Hasan, Parapar, and Blanco (2008).

Text pre-processing. PDF format was originally created to look good to the users. Because of this, when an application builds a PDF containing text this text is not exactly the same as in the original version. For instance, "fi" sequence (numeric code \102\105) is coded as the single character "fi" (numeric code \64257). So, when extracting the text from a PDF document, this issue must be taken into account.

Identification of the titles contained in the index of the document. The main characteristic of these titles is that they always start with a special word ("Resolution", "Notification", etc). *Lexisla* looks for phrases inside the index which begin with these specific words, or with variations of these words. This step returns the titles of the legislative units of the document.

Identification of resolutions and other legislative units using the titles. First, the lexicon terms are searched all over the text. With these list of terms, a list of title candidates is built. Then, this list is compared against the list obtained from the index of the document. Those titles from the content which exactly match a title in the index, and are found in the same order as in the index, are stored. In the case some of the

titles in the index were not matched, the comparison is softened by the use of a comparison using n-grams instead of an exact match comparison.

Identification of full legislative units. The full content of each unit will be the text between its title and the title of the next unit.

To evaluate our segmentation algorithm we built an evaluation set composed of 20 documents from heterogeneous sources, providing more than 1400 legislative units. The metrics used to evaluate the segmentation algorithm were recall (number of units correctly extracted divided by the total number of units) and precision (number of units correctly extracted divided by the total number of extracted units). The results are very good, with a mean precision of 97,85 % and a mean recall of 95,99 %. Also, for every source both values stand over 93 %. Regarding the computing time, the algorithm needs 0,13 seconds per legislative unit[2].

5. Conclusions and Future Work

In this work we presented a LegalIR system, *Lexisla*. The implementation of the application had special research challenges, like extraction of text from PDF documents, or segmentation of the documents into its comprised legislative units, which were successfully faced as it is shown in the evaluation of the results. To accomplish this, the system makes use of several NLP techniques, like string similarity comparisons, stemming, searches using regular expressions, the use of a specific lexicon to segment the documents and language identification features.

As for the future, there are several tasks to be considered in the domain of this application:

- Entities detection. It can be very interesting to infer which are the people or enterprises affected by a concrete notification, resolution, etc.
- Crossed references. It is very usual that a legislative unit makes a reference to another one. The automatic detection of this references can improve the users' experience. The work of Yang et al. (2009) can provide a good startpoint to this feature. In this work, the authors face the problem of using an entire document as a query to do a search. One of the main steps proposed is the extraction of phrases from the document to be used as queries; similar methods can be used to identify the crossed references inside a legislative text.
- Generalisation of the segmentation algorithm to be used in different domains.

[2]Pentium 4, 3GHz, 1GB of ram

References

Apache Software Foundation. 2010. Pdfbox. http://pdfbox.apache.org/.

Hasan, Ismael, Javier Parapar, and Roi Blanco. 2008. Segmentation of legislative documents using a domain-specific lexicon. In *DEXA '08: Proceedings of the 2008 19th International Conference on Database and Expert Systems Application*, pages 665–669, Washington, DC, USA. IEEE Computer Society.

Mao, Song and Tapas Kanungo. 2001. Empirical performance evaluation methodology and its application to page segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):242–256.

Phelps, Tom. 2010. Multivalent. http://multivalent.sourceforge.net/.

Yang, Yin, Nilesh Bansal, Wisam Dakka, Panagiotis Ipeirotis, Nick Koudas, and Dimitris Papadias. 2009. Query by document. In *WSDM '09: Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 34–43, New York, NY, USA. ACM.