Segmentation of legislative documents using a domain-specific lexicon

Ismael Hasan
Information Retrieval Lab
Computer Science Department
University of A Coruña
irlab@udc.es
http://www.dc.fi.udc.es/irlab

Javier Parapar
Information Retrieval Lab
Computer Science Department
University of A Coruña
javierparapar@udc.es
http://www.dc.fi.udc.es/irlab

Roi Blanco
Information Retrieval Lab
Computer Science Department
University of A Coruña
rblanco@udc.es
http://www.dc.fi.udc.es/irlab

***Abstract*—The amount of legal information is continuously growing. New legislative documents appear everyday in the Web. Legal documents are produced on a daily basis in briefing-format, containing changes in the current legislation, notifications, decisions, resolutions, etc. The scope of these documents includes countries, states, provinces and even city councils. This legal information is produced in a semi-structured format and distributed daily on official web-sites; however, the huge amount of published information makes difficult for an user to find a specific issue, being lawyers probably the most representative example, who need to access to these sources regularly. This motivates the need of legislative information search engines. Standard general web search engines return to the user full documents (web pages typically), within hundreds of pages. As users expect only the relevant part of the document, techniques that recognise and extract these *relevant bits* of documents are needed to offer quick and effective results. In this paper we present a method to perform segmentation based on domain-specific lexicon information. Our method was tested with a manually tagged data-set coming from different sources of Spanish legislative documents. Results show that this technique is suitable for the task achieving values of 97'85% recall and 95'99% precision.**

I. INTRODUCTION

The huge amount of legislative information available on the Internet requires the use of information retrieval (IR) systems to convey it to the final user. Nowadays, the most useful sources of legislative information are jurisprudence sources and official bulletins.

Focusing on the official bulletins, there is a big problem with users access to the information contained in official electronic publications: the daily amount of published information is too big to be manually analysed and searched over. The current situation in Spain is a paradigmatic example: there is a national bulletin[1], which is usually between 100 and 300 pages long, one regional bulletin per region (there are 17 regions in Spain), each of them about 70 pages long; moreover, there are several province and city's council bulletins. Region and state bulletins are produced almost everyday, which implies that there are each day more than 1000 document pages containing new resolutions, changes in laws, etc. Furthermore, the fact that each of these bulletins is published on its own place on the Internet, makes manual searching a cumbersome task. Some professional collectives, like lawyers for instance, need to consult this information continuously. All these reasons, fuel the need of specialised search engines that provide effective and efficient retrieval over the bulletins.

Bulletins are composed of many issues (resolutions) and users usually look for one particular resolution. Therefore, original files containing all the resolutions must be properly segmented and individual resolutions indexed and stored. Segmentation is also useful to offer advanced searching, like searching over the titles, over different fields or between two dates, and so on.

Most of the official bulletins are published in PDF format although some of them are also published in HTML format. At this point we have to face a new problem, namely the extraction of the text from PDF documents. PDF-text extraction utilities are far from being perfect - for example, they may invert the contents of a page -. Apart from plain text, pdf-text extraction software may allow to extract some formatting options (like typesetting, for instance); unfortunately, with a much higher error-rate.

The problem we must tackle is to detect the segments of the information published in the official bulletins by just using the plain text that these bulletins contain. We propose a method based on the structure of legal documents and the use of domain-specific lexicon information.

The rest of the paper is organised as follows. In section II we present some background on legislative IR in order to motivate this research. In section III we explain our developed proposed method: the purpose of using a lexicon in this method, a general overview, the algorithm in depth and its computational complexity. Finally, in section IV we show the experiments and results, and in section V conclusions and future work are reported

II. BACKGROUND

Electronic legislative information is widely spread across the Web. Publications on legal issues are very heterogeneous, because most of them are ruled by different government organisms. Despite the fact that these organisms often provide some search capabilities, the offered results only span these

[1] "Boletín Oficial del Estado", http://www.boe.es

organisms' published documents. There are few attempts to offer access to legislative information through a simple site [1], but this is not common.

Also, there are some approaches for tackling the problem of automatic indexing and retrieval of legislative information, (Gómez-Pérez et al. in [2], van Noortwijk et al. in [3]), but most of them focus on issues which are different from the segmentation of the retrieved documents, for instance, retrieval of full documents.

The fact that the legislative information we are working with is structured (although this structure is far from being homogeneous) marks the way to follow to build legislative IR systems. Professional sectors that use this kind of information (lawyers, for instance), must obtain legal information structured in such a way that resembles how those legal documents look like when printed on paper. Hence, it is necessary to develop techniques to offer legislative IR systems which preserve the structure of documents in order to search over or to return search results. Matthijssen introduced this issue in [4], where the author presents two facts to be taken into account, namely: first, to include in the system domain-specific information helps the user to query the system more effectively, and second, to include field information helps the system to offer more accurate results.

Finally, Smith, in [5], emphasises the use of specialised lexicon terms in legislative retrieval systems over legal databases, because the elements of this lexicon are familiar to the user and they allow the creation of an efficient labelling method.

We will now present two attempts to centralise the legislative information, two approaches to the segmentation of this information and finally we will describe and motivate our proposed solution.

A. Legislative documents centralisation cases

In this section we will comment two cases which are intended to centralise legislative information: "The Norme in Rete Project" (NIR) and "The Legal Information System of the Republic of Austria" (RIS).

1) The NIR project: The NIR project is discussed by Marchetti et al. in [6]. The XML format is proposed as a method to distribute electronic legislative documents. This way, information can follow a standard schema and its contents can be correctly extracted, analysed and segmented. However this standardisation requires acceptance from publisher organisms. This seems indeed a big problem, given that one country might have multiple publisher organisms; moreover, international standardisation obviously becomes more complicated.

2) RIS: This system is coordinated and operated by the Austrian Federal Chancellery [1, RIS]. Through this system, information of official documents is offered freely via the Internet. The system allows to make queries over this information, and the documents are categorised. To keep the information updated the official Austrian publishing organisations send electronic copies to the Chancellery; this information comes correctly tagged so it can be automatically treated and further indexed. However, this is not the general situation. Searching over centralised information is a users' desirable issue; but legal information is usually scattered and coming from different sources. Therefore, it is necessary to build methods to analyse heterogeneous information in order to offer this information to the final users.

B. Other approaches to the segmentation of legislative documents

Most of the research involving legislative documents has focused on document retrieval and classification. However, there have been some approaches concerned on retrieving *items* inside the documents.

Biagioli et al., in [7], propose a method to locate the *provisions* inside a text (pieces of legal documents containing a single resolution, law, or information item to be presented to the final user), and to classify them into the categories proposed by NIR. The technique builds a set of terms that comes from document testbed. This set of terms is chosen by preprocessing the data and collecting statistical information in the documents collection. Then, it is considered that every paragraph in the text is a *provision*, and this is classified using the terms which were previously collected. After the classification process more information is extracted from the paragraph (like the legal entities involved).

This method is suitable if we consider that a provision is one paragraph and we can obtain the paragraphs from the document structure. Unluckily, the actual situation is different; it is difficult to identify paragraphs in some situations, for instance, when one paragraph continues in the next page; and a provision usually cover several paragraphs.

Moens in [8] discusses the retrieval of XML-tagged legislative documents. These kind of documents are structured in fields, and there is a hierarchy between them. The fact that the information is previously tagged makes the segmentation of documents easier; however, as mentioned before, this is an unusual scenario.

C. Motivation

We showed in this section two rare cases (NIR and RIS) for legislation information processing. We also showed the need to segmentate legislative documents to offer precise querying over them, and we analysed two former approaches to the problem. We encompass a wider scenario, as our belief is that the afore mentioned situation is not common.

III. Developed method

In this section we explain the developed algorithm to segment official legislative documents. First we define our retrieval unit, called "resolution"; then we explain the use of lexicon items in this context, to continue with an overview of the method. After that, we discuss the algorithm in depth and finally we expose some of the problems found in the implementation and their solutions.

A. Previous definition

Resolution: Composing item of the documents we are working with, and our information retrieval unit. A resolution is composed by two items: the content of the resolution, and a title summarising that content (and preceding it). An example of resolution could have the title “Notification to Fictitious Enterprise of court trial involving Ficticious Enterprise”, and its content would be the explanation of the reasons of the trial, the date, the place, etc. Also, remark that the legislative documents we are working with are composed of several *resolutions*.

B. Use of the lexicon

The context of legislative documents, and particularly the official bulletins publications, has its own lexicon terms. The use of this specialised lexicon terms is advisable to offer results and query utilities to the user, and to segment the documents into resolutions. Since this lexicon depends on the document language, we should construct one lexicon for each language in our documents, and a lexicon is shared for all the documents in the same language.

Moreover, these documents share very often a similar structure: they begin with an index, referring to the official resolutions in the content, and then the content follows.

The proposed method tries to locate the titles contained in the index; to do this, it uses lexicon terms and their positions in the text. In Spanish documents, these terms appear always at the beginning of a title (“*Resolución* de 25 de Noviembre de...”). This method can be adapted to languages with different syntactic structures, like English, by just changing the position in which we expect that a lexicon term must appear. Once these titles are located, we can search for them in the content of the document to segment the resolutions that it contains.

So the legislative lexicon terms are very important in the developed method: by using it, we can identify the titles of the resolutions in the index. Also, the terms help to find the titles (previously identified) in the content part of legislative documents.

The method relies on the index of the documents to segment them. The fact that we use the index allows us to focus the search over the whole content in a few items.

C. General overview

To offer a generic view of the method we will make clear which are its goals: given a legislative document and a a set of lexicon terms for the language of the document, we are looking for a way to segment the document into resolutions. In this work we focus on the retrieval of documents written in Spanish. A set of documents has been used to build the algorithm for testing purposes, using different train and evaluation sets; these documents were also used to build the lexicon for the algorithm. Ten documents were randomly selected from each of the sources; these sources were BOA (region of Aragon), which documents provided 734 resolutions, BORM (region of Murcia), which documents provided 461 resolutions, and DOG (region of Galicia), which documents provided 604 resolutions.

These documents can be found in the Internet in PDF format and there is one new document almost everyday. These documents have in their very first pages an index containing the titles of the resolutions that are present in the text. Documents are stored in PDF format; as mentioned earlier, text extraction from this files has some errors, but given that this is the more common format employed electronically we have to deal with it.

From this set of documents we collected a list of domain specific lexicon terms, containing a total of 70 items. The lexicon was processed manually, going through the documents’ indices and keeping the representative words for the category of each title.

The algorithm is based upon the next assumptions:

- The document which will be segmented must have an index; this will refer to the content of the document. The titles of the different resolutions will appear in this index.
- The titles of the resolutions always contain specialised lexicon terms (like “resolution”, “law”, “announce”, etc).
- Every resolution begins with its title, which must be the same which appears in the index. Also, the resolutions in the content must appear in the same order as its titles in the index.

The first assumption usually happens: these documents are electronic versions of printed documents. Since they are very long (dozens of pages) they have an index referring to the content of the document.

The second assumption is also common: resolutions must have a title to appear in the index, so one user can look for information in the index; this is the reason why these titles must contain significant information (lexicon terms).

Finally, the third assumption is common too: the document must present the resolutions in the same order they are presented in the index.

The algorithm goes as follows:

1) Documents preprocessing and normalisation.
2) Identification of the titles contained in the index.
3) Search of titles of the index in the contents attending to their order in the index.
4) Search of the index titles that were not found in the previous search using string similarity techniques.
5) Attempt to rebuild the pages which have their content extracted in a wrong order.
6) After the titles have been located in the content of the document, building of resolutions adding to each title the corresponding content.

In the next section we will explain the algorithm.

D. Algorithm

1) Documents preprocessing and normalisation: The documents to treat are extracted text from PDF files, so it is needed to normalise them first: text extraction utilities fail often. At this point we have to deal with these errors and we have to make the text suitable to its further treatment as follows:

- Deletion of chains of space characters.
- Correction of known bad recognised characters. Since the method relies on a lexicon it is important that the words have the right characters (for example, it is common to find the character "fi" instead of "f i").

There is another text processing issue that is important in this phase, and it is not related to PDF extraction problems: replacement of words that are splitted across two different lines. Given that this method looks up lexicon terms in the text, replacing the split-words with their unsplitted forms is more efficient than looking in the text possible split variations of the lexicon terms.

2) Identification of the titles contained in the index: This part of the algorithm operates as follows:

1) Looks up lexicon terms appearances in the index.
2) Doesn't consider the lexicon terms appearances that do not appear in the beginning of a line.
3) Once the relevant lexicon terms appearances are selected, titles are extracted selecting the text between each appearance and the next line end.
   Once we have located the lexicon terms that we consider significant to each title, we mark as a title the text between each found term and the next fullstop.

3) Search of titles of the index in the contents of the whole document attending to their order in the index: Titles of indices may not be exactly the same titles in the content; for instance, if they appear with a different font size in the original document it may happen that new line characters are in different positions in both titles. Therefore, we must locate the lexicon terms in the content in the same way they were located in the index.

Titles located in the index are then compared with the titles in the content, deleting special characters and leaving letters only. Those which match exactly are stored as found content titles.

Notice that the process maintains the sequential order between different titles of the index as a method to avoid false positives.

4) Search of the index titles that were not found in the previous search using string similarity techniques: The method that looks for exact coincidences between the indices and actual content is too strict; these documents are manually written, so they may contain typing errors. Also, differences between the title in the index and the title in the content may happen due to the use of shortcuts ("no" instead of "number", for example).

In order to deal with these situations, the document is reviewed in this phase in a similar way as phase 3, with two main differences. First, only the titles which have not been found in phase 3 are locally searched, in the window of text in which they should be attending to the order of the index. Second, coincidences are searched using similarity techniques instead of exact matches. In our case we used similarity comparison using n-grams; we established a similarity threshold of 0.85 (between 0 and 1). The described problems are solved by using this threshold, and shortcuts and simple typos do not affect to titles identification in the content section.

5) Attempt to rebuild the pages which have their content extracted in a wrong order.: The content of a page may sometimes be inverted in the extraction process: this is, the contents in the end of the page appear before the contents in the beginning in the extracted text. This inversion may happen usually when there is a font style change or a font size change in the original document (when a new section or a resolution begins).

We try to deal with these text extractor errors in this phase: after going through the previous phases, there can be some *not found* titles. We search each of those titles locally in the same page in which the previous found title is; this search is made over the whole page (we are ignoring the expected sequential order). This way, if the not found title is found before a title which is expected to precede the not found title, we assume that there has been a page inversion and we can reconstruct the original content. If the title is not found a similar processing is made, using as reference the next found title instead of the previous one.

6) After the titles have been located in the content of the document, building of resolutions adding to each title the corresponding content.: The contents of each resolution will be the text between its title and the next one.

E. Complexity and efficiency issues

An analysis of the algorithm shows that it is linear -O(n)- over the text size.

There are two main operations in the algorithm: treatment of substrings and similarity comparisons between strings.

- Treatment of substrings: efficient methods for substrings treatment are necessary due to the size of the documents. Our implementation uses Java 1.6.1. By using the "StringBuffer" class to represent the text, processing time was reduced in a 95% respect to the time using the "String" class (to process a document of one hundred pages with "String" took several minutes, whereas to process it with "StringBuffer" lasted only a few seconds).
- Similarity comparison between strings: this operation is more expensive than exact comparisons; hence, the use of similarity comparisons is reduced only to *not found* titles and in a local context (and not to the full document).

IV. Evaluation

A data set was elaborated to evaluate the system, different from the one used to implement the method and to build the set of lexicon terms. The sources of this new documents set are also different from the sources of the set used to implement the method.

A. Test data set

There is not a collection of data oriented to test legislative documents segmentation methods, so the documents have been manually selected and tagged. The evaluation set is composed from 4 official documents sources, each of them contributing with 5 randomly selected documents produced in different days. These sources were: BOC (region of Canarias), which documents provided 102 resolutions, BOCYL (region of Castilla y Leon), which documents provided 465 resolutions, BOJA (region of Andalucia), which documents provided 444 resolutions, and DOE (region of Extremadura), which documents provided 409 resolutions.

The test set is not as wide as the usual data collections used in information retrieval tests, however it has been limited because checking if a resolution is correctly segmented was done manually (anyway, the data-set was above 1500 pages).

B. Evaluation method

The evaluation of the method was done attending to recall (number of true positives divided by the total number of real resolutions) and precision (number of true positives divided by the total number of retrieved resolutions) terms. We have been very strict to give a true positive.

1) False positives:

- Resolutions that are present in the index but were not found in the text.
- Resolutions that have inverted contents in one page (due to problems with PDF text extraction), regardless whether the limits of the resolution were correctly identified or not.
- Resolutions containing more text than the text they should have (for instance, because the next resolution was not correctly located and then the limits of the current resolution are not well defined).

2) True positive: A resolution is considered as a true positive when it meets every following requisite:

- It goes along with a element of the index.
- Its limits are correctly identified.
- It has not any inner page with its contents inverted due to the problems with PDF text extraction.

C. Results

NAME	PRECISION (%)	RECALL (%)	MEAN TIME (S)
BOC	98.00	96.08	4.24
BOCYL	98.45	95.70	7.73
BOJA	99.09	98.42	12.48
DOE	95.75	93.64	12.07
Overall	**97.85**	**95.99**	**9.13**

TABLE I
PRECISION, RECALL AND TIME RESULTS OF THE SEGMENTATION OVER THE DIFFERENT BULLETINS

The table shows excellent precision and recall levels; both values stand over 90% (in global result and in documents result), regardless we have been restrictive to consider a positive result. In terms of time the average time to treat a document has been 9,15 seconds[2]; the average size of the test documents is 99,2 pages, and each one contains a mean of 71 resolutions (0,13 seconds per resolution). We consider that this is an acceptable computing time in this particular context.

V. CONCLUSIONS

In this work, we faced the problem of the segmentation of legislative documents. We identified some characteristics, usually accomplished by the electronic published documents, and upon them we proposed a method, based on building a domain-specific lexicon and employing the common structure of legislative documents. In our case, we focused on Spanish documents, but this is extensible to other languages.

Finally, a flexible method has been developed; it works with several languages and it can be easily modified to other languages with different structures from the ones already issued. The method achieves good levels of precision and recall. Also, regardless the technique has not a 100% recall, the information is not lost by the method used to extract the resolutions content: if some resolution is not found its content is included in the previous one.

As future work it would be very interesting to extract meta-information from the segmented resolutions, like their dates or the people and organisms affected by a resolution. Also, it would be interesting to add "special" lexicon terms: those who represent resolutions which can have and inner segmentation (like laws, for example). Finally it would have some interest the automatic creation of the lexicon, given a set of documents.

ACKNOWLEDGEMENT

This work was cofunded by the "Secretaría de Estado de Universidades e Investigación" and FEDER funds under research project MEC TIN2005-08521-C02-02 and "Xunta de Galicia" under research project 07SIN005206PR. The authors also would like to thank Ávaro Barreiro for his useful comments, suggestions and the support obtained.

REFERENCES

[1] "The legal information system of the republic of Austria," http://www.ris.bka.gv.at/info/english.html, 2008.

[2] A. Gómez-Pérez, F. Ortiz-Rodriguez, and B. Villazón-Terrazas, "Ontology-based legal information retrieval to improve the information access in e-government," in *WWW '06: Proceedings of the 15th international conference on World Wide Web*. New York, NY, USA: ACM, 2006, pp. 1007–1008.

[3] K. van Noortwijk, J. Visser, and R. V. D. Mulder, "Re-usable retrieval concepts for the classification of legal documents," in *ICAIL '05: Proceedings of the 10th international conference on Artificial intelligence and law*. New York, NY, USA: ACM, 2005, pp. 252–253.

[4] L. J. Matthijssen, "An intelligent interface for legal databases," in *ICAIL '95: Proceedings of the 5th international conference on Artificial intelligence and law*. New York, NY, USA: ACM, 1995, pp. 71–80.

[5] J. C. Smith, "The use of lexicons in information retrieval in legal databases," in *ICAIL '97: Proceedings of the 6th international conference on Artificial intelligence and law*. New York, NY, USA: ACM, 1997, pp. 29–38.

[6] A. Marchetti, F. Megale, E. Seta, and F. Vitali, "Using XML as a means to access legislative documents: Italian and foreign experiences," *SIGAPP Appl. Comput. Rev.*, vol. 10, no. 1, pp. 54–62, 2002.

[7] C. Biagioli, E. Francesconi, A. Passerini, S. Montemagni, and C. Soria, "Automatic semantics extraction in law documents," in *ICAIL '05: Proceedings of the 10th international conference on Artificial intelligence and law*. New York, NY, USA: ACM, 2005, pp. 133–140.

[8] M.-F. Moens, "Combining structured and unstructured information in a retrieval model for accessing legislation," in *ICAIL '05: Proceedings of the 10th international conference on Artificial intelligence and law*. New York, NY, USA: ACM, 2005, pp. 141–145.

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