

Jurisprudence search in Colombia based on Natural Language Processing (NLP) and Lynked Data

Búsqueda de jurisprudencia en Colombia basada en Procesamiento de Lenguaje Natural (NLP) y Lynked Data

DOI: <http://doi.org/10.17981/ingecuc.16.2.2020.22>

Artículo de Investigación Científica. Fecha de Recepción: 22/07/2020. Fecha de Aceptación: 27/09/2020.

Cristian Camilo Ordoñez 

Fundación Universitaria de Popayán. Popayán (Colombia)
cordonezq@unicauca.edu.co

Jose Armando Ordoñez 

Fundación Universitaria de Popayán. Popayán (Colombia)
jaordonez@unicauca.edu.co

Hugo Armando Ordoñez Eraso 

Universidad del Cauca- Grupo de investigación GTI. Popayán (Colombia)
hugoordonez@unicauca.edu.co

Franco Arturo Urbano 

Fundación Universitaria de Popayán. Popayán (Colombia)
franco.urbano@docente.fup.edu.co

Para citar este artículo:

C. Ordoñez, J. Armando Ordoñez, H. Ordoñez Herazo & F. Urbano, “Jurisprudence search in Colombia based on natural language processing (NLP) and Lynked Data”, *INGECUC*, vol. 16, no. 2, pp. 277–284. DOI: <http://doi.org/10.17981/ingecuc.16.2.2020.22>

Resumen

Objetivo— Desarrollar un modelo de búsqueda de sentencias judiciales soportado en procesamiento del lenguaje natural que permita analizar el texto de las sentencias jurisprudenciales. Adicionalmente, se usa link-data con el propósito de aprovechar la interrelación del contenido en las sentencias judiciales relacionadas y mejorar los procesos de búsqueda.

Metodología— El modelo de búsqueda se desarrolló en dos fases: la primera es la fase de entrenamiento para generar los modelos requeridos para crear un índice, y en segundo lugar, una fase de búsqueda donde el usuario ingresa una cadena de búsqueda y se utiliza el índice creado en la fase anterior para encontrar los documentos (sentencias judiciales) relacionados con de búsqueda. Se realizó una comparación con otros buscadores existentes de la Corte Suprema de Justicia de Colombia. La evaluación se dividió en 2 pasos. 1) Evaluación de los resultados obtenidos en cada búsqueda, 2) Satisfacción del usuario ante los resultados obtenidos en las búsquedas.

Resultados— La plataforma desarrollada supera al sistema de búsqueda existente del tribunal en cuanto a satisfacción y precisión del usuario.

Conclusiones— El diseño e implementación del modelo de búsqueda de sentencias judiciales basada en Procesamiento del Lenguaje Natural (PNL) y linked data contribuyó a mejorar la experiencia del usuario y la precisión de la búsqueda de sentencias judiciales.

Palabras clave— Recuperación de documentos judiciales; procesamiento de lenguaje natural judicial; evaluación del sistema; resumen automatizado

Abstract

Objective— To develop a search model for judicial decisions supported by natural language processing that allows analyzing the text of jurisprudential sentences. Additionally, link-data is used to take advantage of the interrelation of content in related court decisions and improve search processes.

Methodology— The search model was built in two phases: the first is the training phase to generate the models required to create an index, and second, a search phase where the user enters a search string that is used to find the documents (court decisions) more related to the search. The model was compared with other existing search engines of the Supreme Court of Justice of Colombia. The evaluation was divided into 2 steps. 1) Evaluation of the results obtained in each search, 2) User satisfaction with the results obtained in the searches solution.

Results— The developed platform outperforms the existing search system of the court regarding user satisfaction and precision.

Conclusions— The designed model for judicial sentences based on Natural Language Processing (NLP) and linked contributes to improving the user experience and the precision of the jurisprudence search.

Keywords— Jurisprudence; retrieval; natural language processing; system evaluation; automated summary

I. INTRODUCCIÓN

Colombian law recognizes the importance of jurisprudence, understood as a set of decisions issued by judges, which can be used as a legal precedent and a formal source of law [1]. Judges can consider these previous sentences (i.e. jurisprudence) from the State Council and the constitutional court as an essential reference for their decision. Moreover, Colombian law also created the figure of the “jurisprudence extension “ that offers ordinary citizens the possibility of demanding authorities to make decisions based on previous and similar cases [2].

However, in practice, jurisprudence search requires two steps: first, it is necessary to search for statements from similar cases (e.g. robbery, assault, etc.). This search can be carried out through the available search engines, which are based on keywords and syntactic concordance [3], [4]. Second, the search requires identifying the central argument of each sentence that served as the basis for the decision [5].

In Colombia, there are different platforms for jurisprudence search, however, their low efficiency in terms of precision and usability make it difficult to locate legal documents. This situation is because these platforms are designed to explore the complete content of the sentences and without any other consideration.

Several approaches have addressed the issue of automatic processing and search of sentences (a.k.a judgements) [6], [7] [8]. Some techniques based on Natural Language Processing (NLP) help to understand text documents, such as jurisprudential sentences, by analyzing the semantic and syntactic relationships found within the text [10]. However, most of the existing approaches do not apply to the particular legal context of the Colombian Legal System. Moreover, they also left aside the analysis of relations between sentences [9].

Diverse researches [11], [8], uses a methodology of classification, grouping, and search of documents based on neural networks is used, this application helps to locate jurisprudential documents and to administer writings of criminal trials in favour of current processes in a more efficient way. They also do a treatment on the Chinese words to guarantee effectiveness to the process of a grouping of textual contents utilizing the term extraction scheme to select the keywords with the highest frequency as entries of the Propagation Network. Seven criminal categories were selected as the exit objective, presenting very high accuracy results when finding criminal cases useful to the user. Also [12], proposes a method to learn the classification of Chinese legal documents using Graph LSTM (Long Short-Term Memory) combined with the extraction of domain knowledge. First, it performs a judicial domain model which it builds based on ontologies that include higher-level ontology and domain-specific ontology. Second, legal documents are divided into different blocks of knowledge through the top-level ontology and the domain-specific ontology. Third, the information is extracted from the knowledge blocks according to the legal domain model. Similarly [13], a search system for judicial documents supported by artificial intelligence is evaluated. This system is developed to speed up the search and analysis processes of the documents.

unlike the aforementioned works, we introduce an application supported in natural language processing, to understand the text that makes up a judicial sentence, this is done by dividing the text into parts, for the identification of relevant elements in the text of the sentence. Besides, linked-data is used to exploit the interrelation of the documents. NLP and linked data are used for the identification, indexing, and recommendation of documents. Our model was trained with 28000 documents extracted from the Constitutional Court. The Platform developed was evaluated regarding users’ satisfaction and execution time.

II. METHODOLOGY

PROJLAW was developed in two phases: the first one is the training phase to generate the models required to create an index, and the second one is the search phase where the user enters a query and the platform uses the index created in the training phase to find documents related with the search string.

The PROJLAW platform was implemented using two approaches for generating the index. The first is based on the Latent Semantic Analysis (LSA) algorithm, which is to analyze the semantics of the set of documents. And the second is based on linked-data categories extracted from each document and related to categories of DBpedia [14].

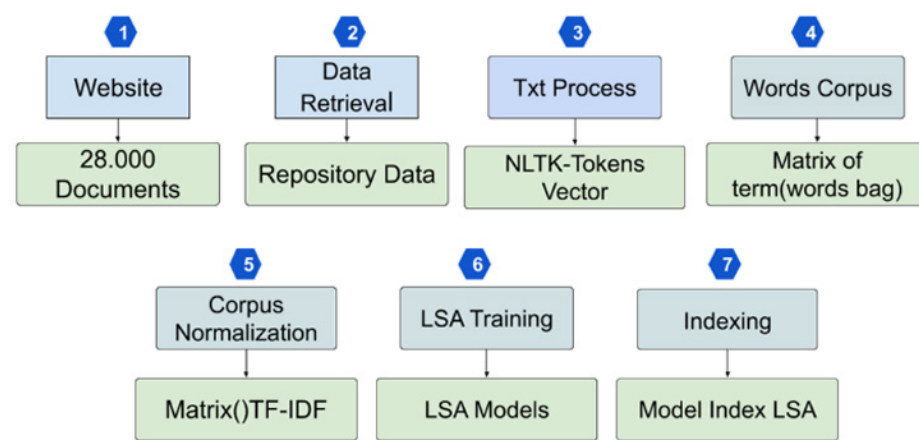


Fig. 1. Shows the steps that the LSA-based approach uses to generate the index.
Source: Authors.

In *step one*, the scraper (Scrapy) extracted 28000 jurisprudential documents (i.e. statements or judgements) from the repository of the constitutional court (Fig. 1).

In *step two*, each document is stored in the repository which is a MySQL database.

In *step three*, the text of each document within the repository (training set) is processed with NLTK. NLTK is a Python package that provides various natural language processing algorithms. This algorithm generates a token vector and a dictionary of terms for each document of the repository. At the end of this step, a matrix with rows containing documents and columns with tokens(words) extracted for each corresponding document. The dictionary of terms consists of tuples ID and word, which are used to improve the processing because the performance for processing numbers outperforms text processing.

In *step four*, a corpus (bag of words) is created using the matrix of documents, tokens(words), and the dictionary of terms. This corpus was created with the gensim library of python [15].

In *step five*, the algorithm TF-IDF is used to estimate the frequency of the tokens for each document. So each cell of the matrix contains the frequency, in which a token was found in each document, rows are documents and columns tokens. This matrix is then normalized to ease further calculations and training processes.

In *step six*, the algorithm LSA creates a model index LSA using the corpus normalized as a training set. This algorithm receives as parameters, a value k (number of topics), the dictionary of terms, and the corpus normalized. At the end of the training process, the technique Singular Value Decomposition (SVD) creates three k-dimensional matrices.

In *step seven*, the LSA index model, namely LSI, is created based on the normalized corpus the LSA model and the dictionary of terms. The LSI model indexes the documents of the repository to execute queries using the cosine similarity. The new index LSA model is a matrix of features with 28000 rows, one for each document, and a column for each feature. Finally, this module is used for improving the performance of queries.

A. Creation of the Linked data-based index

The following steps describe the linked-data processing for the semantic analysis for jurisprudence search (Fig. 2).

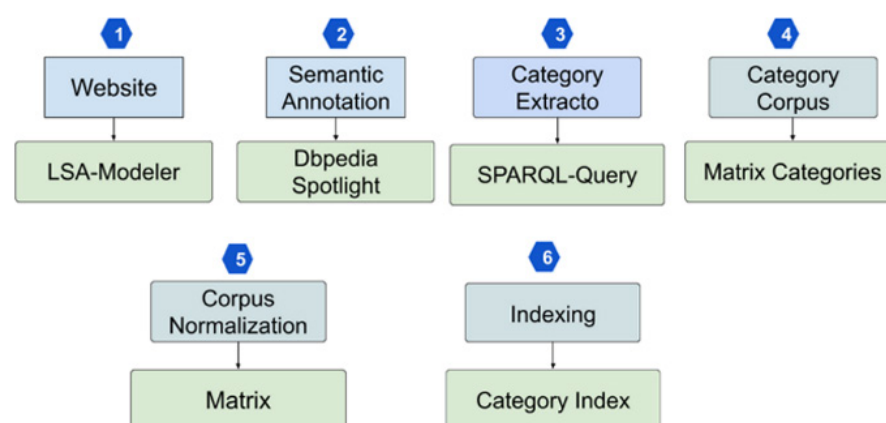


Fig. 2. Steps of the Linked Data-based approach for indexing.
Source: Authors.

B. LSA

Fig. 3 shows the steps that the Linked data-based approach uses to generate the semantic index.

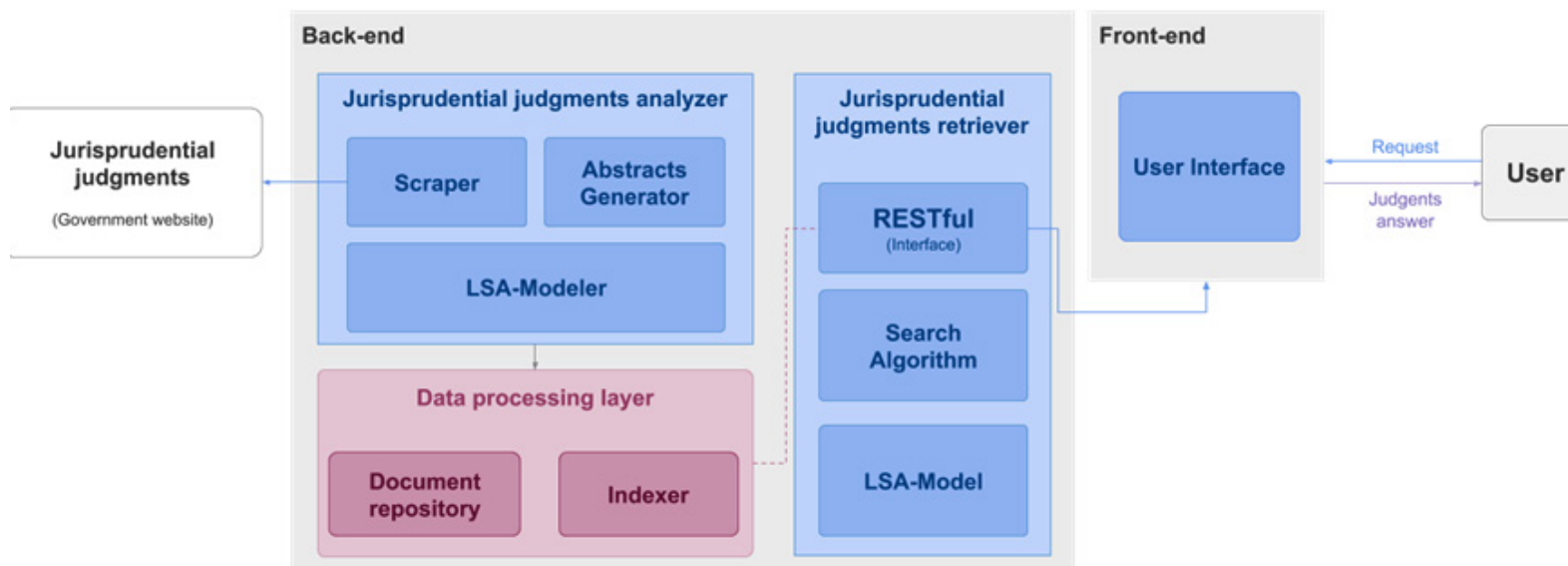


Fig. 3. Modular Diagram of the PROJLAW.
Source: Authors.

Step one is the same as in the previous section, the scraper (Scrapy) extracts jurisprudential documents, which are then stored into the repository.

In step two, the system uses a semantic annotator to extract relevant concepts (represented with a URI) linked to a semantic dataset. In this project, we have selected the semantic annotator DBpedia Spotlight [16], which interlinks text documents with DBpedia. DBpedia is one of the main datasets in the Linked Open Data cloud-based mainly in data extracted from Wikipedia. Linked Data is a set of good practices or principles for publishing and linking structured data on the Web [17]. We chose DBpedia spotlight because it uses DBpedia as dataset which directly connected to Wikipedia's vast, multilingual, pre-annotated corpus [18].

In step three, after DBpedia Spotlight has extracted the semantic concepts, the system uses an algorithm to extract categories from these concepts using a SPARQL query shown in Listing 1.

```
PREFIX dcterms:<http://purl.org/dc/terms/>
SELECT ? cat WHERE {
  <inURI> dcterms: subject ?
  cat.}
```

Listing 1. SPARQL query to obtain the base categories for the <inURI>. .
Source: Authors.

Listing 1 presents the SPARQL query used where <inURI> is the URI of each one of the semantic concepts that DBpedia spotlight has extracted from each jurisprudential document.

In step four, the most common categories are used to identify each document and to create a normalized matrix of frequencies, where rows are documents, columns categories, and each cell represents the frequency in which each category was extracted from each document.

Steps 5 and 6 are similar to the LSA-model creation process, described in the previous section.

C. Proposal

Fig. 4 shows the modular diagram of the PROJLAW platform divided into four main layers; one at the front-end for the user interface; and three at the back-end, namely a data processing layer, a jurisprudential judgments analyzer, and a jurisprudential judgments retriever.

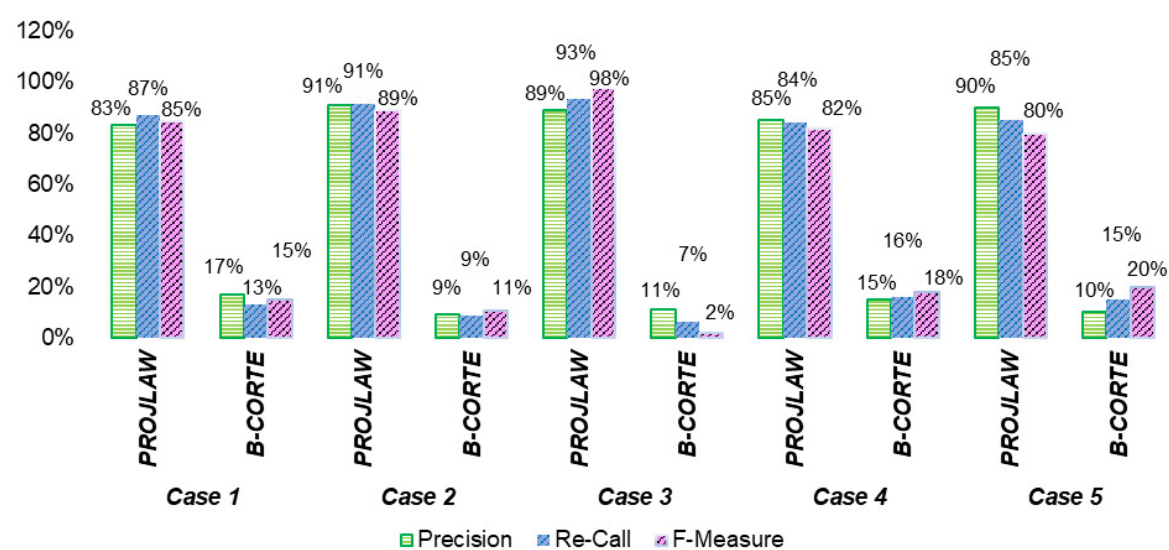


Fig. 4. Internal Evaluation.
Source: Authors.

1) Jurisprudential judgments analyzer

This layer reads all the documents within the website of the constitutional court of Colombia using a scraper tool. This is used once a week to update the documents in the repository, thereby updating the production database. This website contains about 28 000 jurisprudential documents that are processed and stored in a document repository. This layer contains three modules:

Scraper: The platform uses a web scraping tool to extract structured data from hyperlinks and descriptions from jurisprudential judgments published on the website of the constitutional court of Colombia. Specifically, this module was developed using the Scrapy API, which is a library written in python. It uses a crawl that makes requests and loops through elements in the website using a CSS selector.

Abstracts Generator: This module creates summaries of 300 words for each jurisprudential document to help the user to have an idea about the topic of the documents. This module is based on ranks of text sentences using a variation of the TextRank algorithm. This algorithm is graph-based and was selected because it is domain and language-independent, so it does not require corpora with a domain or language-specific annotations [10].

Creator of Models: This module creates three models useful for training the LSA-algorithm. Initially, the platform analyzes the text of each document to detect and remove of stop words and to obtain the main features. Then, this model assigns an ID for each feature and it creates a dictionary of terms consisting of a set of tuples with IDs and features. Moreover, this module creates a corpus (bag of words) of all the unique words occurring in all the documents of the repository (training set). The word corpus consists of a matrix where each row corresponds to a document from the repository and each column is a feature. Each cell of the corpus contains a number of the frequency of occurrence of a feature in each of the jurisprudential documents. After the corpus is generated, it is normalized using the TF-IDF algorithm, and it is used to train the LSA-algorithm and generates an LSA-Model. The LSA-Model is further the basis of the search algorithm.

2) Data Processing Layer

This layer contains modules for storing, indexing and retrieving jurisprudential documents.

Document Repository: This module stores the scrapped documents in a MySQL database. It stores the URL, year, and the full-text of each document.

Model Repository: This module stores the three models generated in the LSA modeller; i.e. the terms dictionary, the LSA Model, and the index.

3) Jurisprudential judgments retriever

This layer allows the system to search relevant documents from the repository according to a user's query written in natural language.

Search Algorithm: This module receives the user's queries from the REST API module and connects it with the data processing layer to retrieve the most relevant documents to that query in the JSON format.

RESTful interface: This module allows various types of clients to search for jurisprudential documents. These clients may be web, desktop or even mobile applications. This interface delivers the results in a JSON format.

4) User Interface

The only layer at the front-end is the web user interface. In this implementation, a web application was developed using usability principles in mind. Anyway, it can be replaced by any type of application such as a desktop or mobile applications.

III. EVALUATION AND RESULTS

This section describes the evaluation of the proposed search engine, compared to other existing search engines of the Colombian Supreme Court of justice. The evaluation is divided into 2 steps where. Internal evaluation of the results obtained, Satisfaction of the user in front of the results obtained in the searches.

A. Internal evaluation

In this phase, the results obtained by the Pro-Law search engine are compared with the results of the evaluation of the search engine of the constitutional court, for this evaluation there were 154 expert evaluators divided among law students, lawyers, profiles of the evaluators can be observed in [Table 1](#).

TABLE 1. EVALUATOR PROFILES.

University	Student	Professional	Lawyer	Judicial employees
Fundación Universitaria de Popayán	28	10	5	3
Universidad del Cauca	16	3	6	4
Universidad Mariana	15	2	3	1
Universidad Cooperativa	10	1	5	0
Universidad de Nariño	12	3	4	1
Universidad Comfacauca	5	3	5	1
Courthouse	0	0	5	3
Total	86	22	33	13

Source: Authors.

In addition to the comparison with the evaluation of the search engines, the quality of the results of the searches was using statistical measures from the field of retrieval systems [19], these measurements are Precision (P), Recall (R), F-Measure (FM) [20]. [Fig. 4](#) presents the precision of ROJLAW search engine for 5 cases given by legal experts. The results of the P in the search of cases are described below.

[Fig. 4](#) shows the precision of PROJLAW for different cases. For case 1, 78 of the evaluators found that PROJLAW generates better precision when finding documents to solve case 1. Likewise, for cases, 2, 3, 4, 5 PROJLAW retrieved more precise searches with 83%, 91%, 89% and 85% respectively, compared to the search engine of the constitutional court. The latter, the precision was 22% for case 1, 17% case number 2, 9% for case 3, and case 4 and 5 had 11% and 15%. This is due to the fact that the method based on natural language processing allows to better index the results that a legal expert would expect to solve his case in evaluation, on the other hand for the evaluation of re-call, PROJLAW obtained high values for the case 1 it was 87% case 2 it was 91%, case 3, 4, 5 it was 93%, 84%, 85% compared to the court search engine whose values do not exceed 20 per cent this is because the system recovers more quickly Also, the court search engine only generates 10 results during a search, leaving behind documents that may be relevant compared to its contender PROJLAW which guarantees a large number of indexed results in collaboration with the index that creates the recommendation given by the Links that appear as a result of the search, on the other hand, to end, the measure F obtained for the two search engines is evaluated and are presented in [Fig. 4](#). This measure represents the harmony of the results for P and R on average the search engine where PROJLAW has high values, for the different cases starting

from 1 with 85% case 2, 3, 4, 5 with values of 89%, 98%, 82%, 80% compared to the Court search engine which values do not exceed the 25% margin the harmonium of the documents is more relevant from the search engine PROJLAW compared to Court search engine.

B. Evaluation of user satisfaction

This section describes the evaluation of the satisfaction of the users of PROJLAW during the search. Fig. 5 shows the results of the following questions.

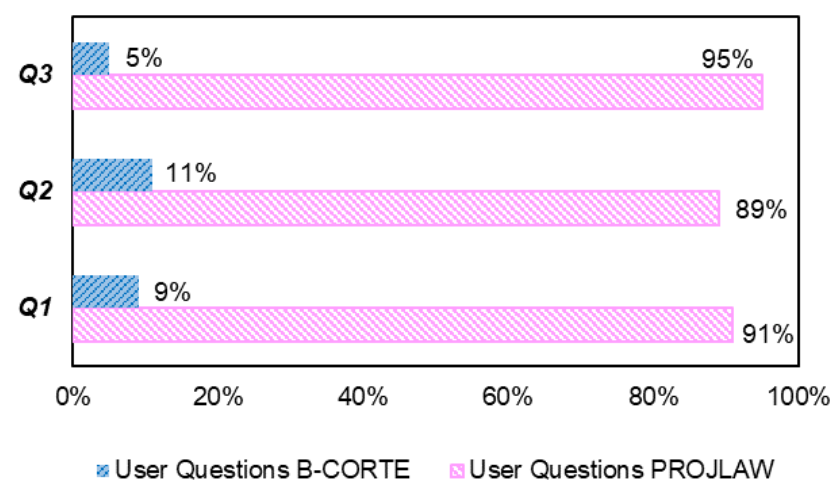


Fig. 5. Survey results.
Source: the authors.

Q1. ¿The results of the search engine are relevant and adequate to solve a particular case? 91% of the evaluators assure that the system retrieved adequate results to solve the evaluated case, this means that the obtained results help and contribute to the experts in the subject to solve the evaluated cases.

Q2. ¿What platform generates an adequate summary of the search performed? 89 per cent approve the summary generated by PROJLAW and 11% of the evaluators believe that the summary given by PROJLAW is not accurate.

Q3. ¿Which platform generates a suitable document recommendation of the search performed? 90% approve the recommendation of documents generated by PROJLAW, the remaining 10% consider that the recommendation provided by B-CORTE is more appropriate.

IV. CONCLUSIONS

This paper presented the design and implementation of the platform for searching judicial sentences based on (NLP) Natural Language Processing. Validation was carried out by experts and users. The user experience was improved as well as the accuracy of the search. A platform to search and summarize law documents is presented. This platform is compared to the Colombian Government search engine. To this end, different measures from the state of the art were used (Precision, Recall and F-Measure). Furthermore, user satisfaction was evaluated through surveys. It was determined that the proposed method generates better results according to the results of the metrics and the surveys. It was determined that the platform generates a response to searches efficiently since several keywords can be entered by the user the search. the more keywords, the more precise the result of the search. It was also found that the generated summary is in line with the need for the searches. It should be noted that this platform only has documents of the constitutional court. Other courts were not taken into account. Future work will include the search in other courts. Also, some recommendations will be included to improve the processes of search.

REFERENCES

- [1] M. V. Gaviria, "Aproximaciones a la historia del derecho en Colombia," *Hist Soc*, no. 22, pp. 131–156, 2012. Disponible en <https://revistas.unal.edu.co/index.php/hisysoc/article/view/32363>
- [2] M. R. Segura, "Precedente jurisprudencial vs unificación jurisprudencial," *ensayo inédito*, Bog, CO: UniLibre, 2011.
- [3] República de Colombia. "Sistema de Consulta de Jurisprudencia." *Portal Rama Judicial del Poder Público*. Disponible en <https://jurisprudencia.ramajudicial.gov.co/WebRelatoria/consulta/index.xhtml>. (accedido en 2011)

- [4] República de Colombia. “Sistema de Consulta de Jurisprudencia,” *Portal Consejo de Estado*. Disponible en <https://jurisprudencia.ramajudicial.gov.co/WebRelatoria/ce/index.html>. (accedido en 2015)
- [5] J. B. Vallet, “El razonamiento Judicial,” *An Fund Fco Elías Tejada*, no. 15, pp. 15–28, 2009. Recuperado de <http://fundacioneliasdetejada.org/wp-content/uploads/2014/03/ANA15-015-028.pdf>
- [6] A. Wyner, R. Mochales-Palau, M.-F. Moens & D. Milward, “Approaches to Text Mining Arguments from Legal Cases,” in *Semantic Processing of Legal Texts*, E. Francesconi, S. Montemagni, W. Peters and D. Tiscornia, Eds, vol. 6036. SXF, DEU: Springer, 2010, pp. 60–79. https://doi.org/10.1007/978-3-642-12837-0_4
- [7] G. Venturi, “Legal Language and Legal Knowledge Management Applications,” in *Semantic Processing of Legal Texts*, E. Francesconi, S. Montemagni, W. Peters and D. Tiscornia, Eds, vol. 6036. SXF, DEU: Springer, 2010, pp. 3–26. https://doi.org/10.1007/978-3-642-12837-0_1
- [8] L. O. de Colla & V. L. S. de Lima, “Clustering and Categorization of Brazilian Portuguese Legal Documents,” presented *Computational Processing of the Portuguese Language*, PROPOR 2012, Coi, PT, Apr. 17-20, 2012, pp. 272–283. https://doi.org/10.1007/978-3-642-28885-2_31
- [9] N. Zong, S. Lee, J. Ahn & H. G. Kim, “Supporting inter-topic entity search for biomedical Linked Data based on heterogeneous relationships,” *Comput Biol Med*, vol. 87, no. 1, Dec. 2016, pp. 217–229, 2017. <http://dx.doi.org/10.1016/j.combiomed.2017.05.026>
- [10] A. J. C. Trappey, C. V. Trappey, J.-L. Wu & J. W. C. Wang, “Intelligent compilation of patent summaries using machine learning and natural language processing techniques,” *Adv Eng Informatics*, vol. 43, no. 1, 101027, Jan. 2020. <http://dx.doi.org/10.1016/j.aei.2019.101027>
- [11] R. Kumar & K. Raghuvver, “Legal Documents Clustering using Latent Dirichlet Allocation,” *IJAIS*, vol. 2, no. 6, pp. 27–33, May. 2012. Available from <https://research.ijais.org/volume2/number6/ijais12-450384.pdf>
- [12] G. Li, Z. Wang & Y. Ma, “Combining Domain Knowledge Extraction With Graph Long Short-Term Memory for Learning Classification of Chinese Legal Documents,” *IEEE Access*, vol. 7, pp. 139616–139627, Oct. 2019. <http://dx.doi.org/10.1109/ACCESS.2019.2943668>
- [13] C. C. Ordoñez, E. Anchico, A. Ordoñez, C. Méndez & H. A. Ordoñez, “Sistema de Indexación de documentos Jurisprudenciales soportado en Inteligencia Artificial,” *Risti*, vol. E22, no. E22, pp. 41–52, 2019. Available from <http://www.risti.xyz/issues/ristie22.pdf>
- [14] K. Singh, I. Lytra, A. S. Radhakrishna, S. Shekarpour, M. E. Vidal & J. Lehmann, “No one is perfect: Analysing the performance of question answering components over the DBpedia knowledge graph,” *JWS*, vol. 65, pp. 1–12, 2020. <http://dx.doi.org/10.1016/j.websem.2020.100594>
- [15] V. N. Gudivada & K. Arbabifard, “Open-Source Libraries, Application Frameworks, and Workflow Systems for NLP,” in *Handbook of Statistics*. AMS, NL: Elsevier, 2018, pp. 31–50. <https://doi.org/10.1016/bs.host.2018.07.007>
- [16] P. N. Mendes, M. Jakob, A. García-Silva & C. Bizer, “DBpedia spotlight: shedding light on the web of documents,” presented *7th International Conference on Semantic Systems*, I-SEMANTICS 2011, GRZ, AUT, 7-9 Sept. 2011, pp. 1–8. <http://dx.doi.org/10.1145/2063518.2063519>
- [17] C. Bizer, T. Heath & T. Berners-Lee, “Linked Data - The Story So Far,” *IJSWIS*, vol. 5, no. 3, pp. 1–22, 2009. <http://dx.doi.org/10.4018/jswis.2009081901>
- [18] O. Rodríguez, I. Vagliano, C. Figueroa, F. Cairo, G. Futia, C. A. Licciardi, M. Marengo & F. Morando, “Semantic Annotation and Classification in Practice,” *IT Prof.*, vol. 17, no. 2, pp. 33–39, 2015. <http://dx.doi.org/10.1109/MITP.2015.29>
- [19] C. D. Manning, P. Raghavan & H. Schütze, *Introduction to Information Retrieval*. CBG, USA: Cambridge Univ Press, 2008.
- [20] R. Baeza-Yates & B. Ribeiro-Neto, *Modern information retrieval*. NY, USA: Addison-Wesley Professional, 1999.

Cristian Camilo Ordoñez. MSc. Computer Science from Universidad del Cauca (Colombia). He has worked for custom software development companies, for the national level, independent consultant, guiding consultancies for the improvement of software development processes. Full-time professor belonging to the IMS research group of the Systems Engineering program (Fundación Universitaria de Popayán, Colombia). <https://orcid.org/0000-0003-4157-1611>

Jose Armando Ordoñez. Electronics and Telecommunications Engineer (Universidad del Cauca Popayán, Colombia). Master and Phd. in Telematics Engineering (Universidad del Cauca, Colombia). Currently working as a full time professor (Fundación Universitaria de Popayán, Colombia). Leader of the IMS research group of the Faculty of Engineering. <https://orcid.org/0000-0001-6544-0283>

Hugo Armando Ordoñez Eraso. Systems Engineer (Fundación Universitaria San Martín, Colombia). Specialization in Computer Management (Corporación Universitaria Remington, Colombia). Master in Computer Science and Phd. in Telematics Engineering (Universidad del Cauca, Colombia). Currently works at the Universidad del Cauca and is part of the research group GTI. <https://orcid.org/0000-0002-3465-5617>

Franco Arturo Urbano. Electronics and Telecommunications Engineer (Universidad del Cauca, Colombia). Master in Telematics (Universidad del Cauca, Colombia). Currently a full time professor (Fundación Universitaria de Popayán, Colombia). Researcher of the LOGICIEL research group of the Cauca Engineering Faculty. <https://orcid.org/0000-0002-4120-8604>