Automated Reconstruction of Colombian Public Procurement Traceability

Laura Melissa Montes
Department of Industrial Engineering
Universidad de los Andes, Bogotá, Colombia
lm.montes10@uniandes.edu.co

In Colombia different open government data projects have been developed to improve transparency and enhance citizenship involvement and audit in governmental decision-making processes. We propose a methodology to automatically reconstruct the contracting chain of inter-administrative contracts, using the open contracting database SECOP. We perform the traceability with two approximate string matching techniques: Levenshtein distance and cosine similarity. Using the public entity INVIAS as our study case, we select a random set of contracts to analyze the accuracy and computational performance of each technique. Our automated reconstruction of the contracting chain facilitates the implementation of mechanisms that can increase the citizenship involvement in governmental processes.

***Index Terms*—Public Procurement, Traceability, Approximate String Matching, Levenshtein Distance, Cosine Similarity, Td-Idf.**

I. Introduction

Improving transparency of public management has become an increasing priority for administrations worldwide. In many cases, it is viewed as a feasible response to make up for the lack of government legitimacy and as a mean to cope with the democratic deficit [16]. Openness in the decision-making process can improve citizen trust in government actions [3], as it enhances interaction between regional and local governments with components of civil society [9].

In recent years, governments have been endorsing laws that promote access-to-information [24]. To channel transparency, they are fostering and implementing the widespread use of electronic means for both public administrations and the citizenship. [5]. Latin American countries are implementing public policies aimed at modernizing and increasing public access to governmental information [8], some times with the aid of external parties that support detecting and monitoring deviate behaviors such as corruption [3].

Providing open access to the public sector information is a primary responsibility of democratic governments. On this matter, 7 out of 11 Latin American countries have been working on a continuous strategy of open government data [11]. Colombia is one of the leading countries in the region in open government policies, and has been implementing several initiatives for transparency, accountability and fight against corruption [31]. The Colombian government has deployed several open data platforms in areas such as public contracting, domestic budget and electoral behavior, among others.

While making open data available is an enabler for transparency, it is not the sole step to be taken for ensuring it. Indeed, it exists today an evident detachment between the open government data and decision-making actors, such as the citizenship or the private sector [20]. Public open data may have complex structures that make difficult both data access and its subsequent analysis. Besides publishing the highest possible quantity of data, the efforts of open government policies should aim at providing the means for extracting valuable information out of it and make it available to the interested parties [11, 3].

The work reported in this paper aims at improving the usability of the Colombian public data system, focusing on a case study offered by the public procurement system SECOP. SECOP is a result of the largest open government data project in Colombia [33], which mandates all public contracting to be on-line and accessible. This procurement system has been populating over the last years a database that records information about all contracts issued by public administration entities in the country. It is a valuable source of information that allows breaking-down expenditures according to cost categories, in time and space. Public contracting moves budget from central administration bodies to third parties, sometimes passing through intermediary, decentralized public entities. For several analysis of interest, the traceability of contracts is required, i.e. it is necessary to track the resource destination in several contracts. Such a traceability is currently not provided by SECOP.

We aim in this work to define and implement an automated tool for reconstructing the traceability of the contractual stages of agreements included in SECOP. Our final outcome will allow decision-makers to perform macroscopic exploration of public resource assignment, besides detailed analysis on traced agreements.

As SECOP does not include by design elements that serve the purpose of tracing contracts, we aim at creating an algorithm based on text matching techniques that can be used to evaluate similarity between contract pairs. Our approach, based on the similarity scoring results, can reconstruct the contract chain from central entities down to the third party contractors that ultimately provide the goods or services to the government. Such traceability can provide valuable information for decision-makers to

perform further analysis of governmental spending.

The rest of this document is organized as follows. Section II provides the main elements of the context for this research, describing public contracting in Colombia and related research works. In Section III we describe the SECOP database and characterize its variables. Section IV sketches out the methodology we follow in this research project and describes the algorithmic steps at the basis of traceability reconstruction automation, while Section V reviews the text mining concepts used for its execution. In Section VI we evaluate and validate our methodology with both approximate text matching techniques. Finally, in Section VII we review and discuss the results of our work, providing hints for future developments.

II. Background

Colombia is among the countries with extreme trust losses in government. Recent corruption scandals are significantly affecting trust in Colombian governmental agencies [33], and in recent surveys, only 24% of the participants declared they have confidence in the government course of action [1].

Public administration has executed several initiatives to improve transparency, accountability and the effectiveness of actions against corruption, to enhance trust in governmental institutions [11]. A transparency policy has been defined by the national government to implement openness in the decision-making process. In 2009, the government launched one of its most ambitious projects of open government: the creation of an Electronic Public Procurement System (SECOP, as per its Spanish acronym), to guarantee both more efficiency and transparency in public contracting. Efforts have been performed ever since to strengthen the public procurement system. For instance, the creation of the Public Contracting Agency in 2011 aimed, among others, to improve SECOP data quality and structure [33].

All agencies must register their public procurement information in SECOP. The SECOP system consists of three platforms: SECOP I, SECOP II and *Tienda Virtual del Estado Colombiano*, the e-Store of Colombian Government for the Call-Off of the framework agreements. The former is a publicity platform, while the latter two are transactional platforms. Since the implementation of SECOP II in the second semester of 2015, Colombia is aiming at a transition from SECOP I to SECOP II database. However, at present SECOP I is by far the most utilized platform by public agencies. More than 20 thousand contracts are registered in Tienda Virtual del Estado Colombiano, and almost 200 thousand contracts are registered in SECOP II. On the other hand, SECOP I contains more than 6 million registries. As SECOP I presents by far the highest number of registries, we choose SECOP I (from now on referred as SECOP) as our study database.

Colombian public administration may chose contractors by either a competitive or a non-competitive procurement methods [25]. Non-competitive procurement methods only includes direct contracting, which represents between 33% and 45% of total public procurement spending [33]. Direct contracting includes contracts between public agencies, called inter-administrative contracts, but for terms of this paper we will call them *inter-administrative agreements* (or IAA). In IAA public agencies associate to provide public goods or services [6], and they represent between 40% and 65% of direct contracting spending [33].

The complete execution of an IAA requires two contracting stages, and in each stage distinct contracts are generated:

Stage 1 An agreement is signed between two public agencies. A national public agency transfers resources to a public agency at the sub-national level to provide goods or services in a region.

Stage 2 A public agency at the sub-national level contracts a third party actor to perform the object of the agreement established in Stage 1. More than one contract can be associated to the same agreement.

The agreements and contracts in both stages are manually reported by different agencies, and there exists no requirement to establish a correspondence between the processes in the first and second stages.

The independence between stages in SECOP hinders the identification and audit of the whole contract chain. This limitation of SECOP is just an example of the problems that difficult the exploitation of the information content of open-data for decision-making. Colombian open government data is still difficult to access and interpret, therefore its usage by decision-makers is discouraged [20]. However several governmental and non-governmental initiatives have been developed to strengthen the usage of data by decision-making actors in Colombia.

National government has developed initiatives like Terridata [10], the principal descriptive data repository in Colombia; the Economic Transparency Portal [19], a platform that reports the incomes, spending and contracting in the nation; and Royalties Map [28], a visual representation of the nation royalties paid for the exploitation and exportation of oil, gas and minerals.

Also, third party organizations have created data-driven platforms to promote citizenship usage of open government data. Transparency for Colombia has developed tools such as the Corruption Monitor, which visualizes, qualifies and analyses corruption in Colombia [30]. Other initiatives that aim at favoring the appropriation of the information content of open-data are being developed by organizations such as Datasketch and Somos Más.

Therefore, we believe the goal we set in this work of reconstructing public contracting traceability is a relevant objective, which would result in an increased utility of the SECOP database and in advantages for the whole community.

III. SECOP data

SECOP is a platform where public agencies must publish all documents regarding their contracting activity. It allows

public agencies and private sector to have a formal regulation on contracting processes [7]. SECOP contains more than 6 million registries since 2011, and data is exposed through several formats such as open APIs.

SECOP dataset is composed of 62 columns, where each row is a process registered in the platform. The most relevant columns for our framework and their definition are described in Table I.

TABLE I: Description of the SECOP dataset columns used in this work.

Variable name	Description
UID	Value to identify uniquely each record.
Execution year	Year when the contract was signed (if it was signed).
Public agency name	Name of the public agency responsible for the process.
Process type	Type of procurement method of the process.
Cause of other forms of direct contracting	Description of the cause of the process if it was developed in the "direct contracting" procurement method.
Detailed description of the contract	Description of the good or service acquired in the process.
Contractor name	Name of the contractor chosen in the selection.
Value of the contract	Value of the executed contract.
Link process in SECOP	Link to access the detailed documents of the process.

The complete execution of an IAA is divided in two contracting stages (see Section II). These two stages are independent in SECOP, meaning there exist no identifier to establish a correspondence between them. The process is illustrated in Figure 1.

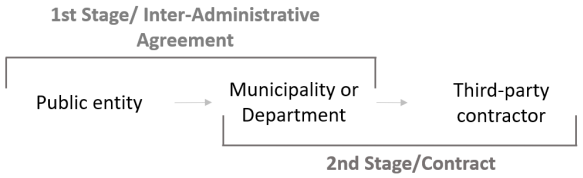


Fig. 1: Stages of the contract chain of an IAA and its respective contracts. The two stages are independent in SECOP.

IV. Methodology

In order to create an automated tool for tracing the contract chain associated with an IAA, and as there exists no standardized identifier for tracking an IAA in SECOP, we manually evaluated the dataset to find fields with common information in all the contracting chain stages. We identified the detailed description of the contract field — or *purpose of the contract* from now on– as an approximate ID for manually trace a particular IAA. As the field is not equal, but similar, in both contracting stages, we used two approximate string matching techniques to match IAA with its respective second-stage contract: Levenhstein distance and Cosine similarity. Therefore, to trace an IAA

Fig. 2: Example of traceability of an inter-administrative agreement between public agency ABC and Municipality of Palmira. The purpose of the contracts are not equal, however we can assume they are part of the same tracing process.

throughout its stages we used the purpose of the contract as our variable for reconstructing the contract chain for the IAA.

In Figure 2 we provide an example of the stages in a traceability process between an IAA of public agency ABC and the contracts of the Municipality of Palmira. The IAA in the first stage was concluded by ABC and the Municipality of Palmira. The second stage contract that presents the highest string similarity is chosen to complete the traceability. The differences between the first and second purpose of the contract are subtle: 'Dept' is an acronym for 'Department' and 'Valle' refers to 'Valle del Cauca'.

As the purpose of the contract field is a text box filled by different actors, most of the times the coincidences between strings are not exact. Therefore we implement an approximate string matching method for mining the SECOP database and trace the contracts associated with IAA. We designed a modular methodology that allows an execution by public agency. We established three phases for our algorithm: preprocessing of the datasets, construction of the traceability of IAA and validation of the model. The methodology is described below:

First phase:

1) Perform an API call to SECOP to obtain the public agency IAA database.
2) Complete an standardization process of the municipalities and departments names in the database obtained in Step 1. A standardized list with all the N municipalities and departments that concluded an IAA with the public agency is obtained.

3) For every municipality/department found in Step 2, identify duplicated IAA performing an approximate string matching algorithm. The contracts with a similarity score above a *duplication threshold* are considered duplicated.
4) Preprocess all the purpose of the contract strings of the database obtained in Step 1. Stop words, punctuation, special characters among others are discarded of modified.

Second phase: Perform an iteration in the list N of municipalities/departments obtained in Step 2. In the nth iteration:

5) Subset the first stage database to keep the M IAA between the public agency and the nth municipality/department.
6) Perform an API call to SECOP to obtain a second database of every ith contract of the *n*th municipality/department.
7) Preprocess all the entries in purpose of the contract of the database obtained in Step 5.
8) For every *m*th contract in data set obtained in Step 4,
   a) Perform an approximate string match algorithm between the *m*th and the *i*th purpose of the contract, of the datasets of Steps 4 and 6 respectively. A similarity metric between 0 and 1 is obtained from each comparison.
   b) Record the similarity score into a comparison matrix.
9) Identify the k contracts with the highest similarity scores with the mth IAA. The algorithm establishes the traceability of the mth IAA with the highest scored contract, however the remaining contracts are also reported.

Third phase: Perform a manual validation of the IAA contract chain dataset obtained in Stage 2.

10) Manually validate if the contract assigned by the algorithm to the *p*th IAA corresponds to the actual contract chain traceability. Additionally, validate if the second contract found also corresponds to the traceability.
11) Based on the precision metrics obtained in the validation step, adjust the algorithm.

As a result, in the second phase we obtained a traceability dataset containing all the public entity IAA and its associated contracts. We performed a validation process to the results to refine our algorithm. We point out that in Step 9 the traceability is established with the highest scored contract, however we report also the remaining $k-1$ contracts. This due to the fact that in some cases an IAA is associated with more than one contract, and also for validation purposes. An illustration of the entire methodology can be found in Figure 3, and a visual representation of the second phase is shown in Figure 4.

To test our methodology we compared two approximate string matching methods: Levenshtein distance and Cosine Similarity with a TF-IDF vectorization. Results are shown in Section V.

V. Approximate string matching

In natural language processing we shift from the general area of exact string matching to the area of inexact, approximate matching. "Approximate" means that some string differences are acceptable in valid matches [14]. Traditional editing methods such as *Levenshtein distance* focus on transforming one string into the other by a series of edit operations on individual characters. Furthermore approaches like cosine similarity measure similitude between two strings based on text-based vectors [17]. In this section we provide a short literature review regarding these two approximate string matching techniques.

A. Levenshtein distance

Levenshtein distance (LD) is an approximate string matching metric used to measure the distance between two strings based on their differences. Such measure is defined as the minimum number of edits required in the form of additions, deletions, or substitutions, to change or convert one string to the other one [26]. The LD between two strings u and v, with lengths $|u|$ and $|v|$ is given by $ld_{|u|,|v|}(i,j)$ and can be formally defined by the following recursive expression:

$$ld_{u,v}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0 \\ \min \begin{cases} ld_{u,v}(i-1,j)+1 \\ ld_{u,v}(i,j-1)+1 \\ ld_{u,v}(i-1,j-1)+1_{(u_i \neq v_j)} \end{cases} & \text{otherwise} \end{cases}$$

where $ld_{u,v}(i,j)$ is the distance between the first i characters of u and the first j characters of v and $1_{(a_i \neq b_j)}$ is the indicator function equal to 0 when $a_i = b_j$ and equal to 1 otherwise. Note that the formula covers all three operations of deletion, insertion and addition between the two strings.

Figure 5 shows a simplified example of the Levenshtein distance. If we evaluate the distance between the words "intention" and "execution", the minimum number of edits to transform one word to another would be 8: 1 deletion, 1 addition and 3 substitutions (as each substitution implies one deletion and one addition).

The standard Levenshtein edit distance is computed by a dynamic algorithm with time complexity $O(|u| \cdot |v|)$. Such complexity would imply prohibitively long computation times when working on large datasets [2]. Therefore the simplicity of the Levenshtein distance implementation algorithm could be overshadowed by its processing time.

We used the Python library `fuzzywuzzy` from SeatGeek [27], which returns a similarity ratio of two strings between 0 and 1 based on Levenshtein distance.

B. Term frequency Inverse document frequency

Term frequency - Inverse document frequency (TF-IDF) is a metric aimed to reflect how important is a word in a document. It is composed by Term Frequency and Inverse Document Frequency. Term Frequency (TF) describes how

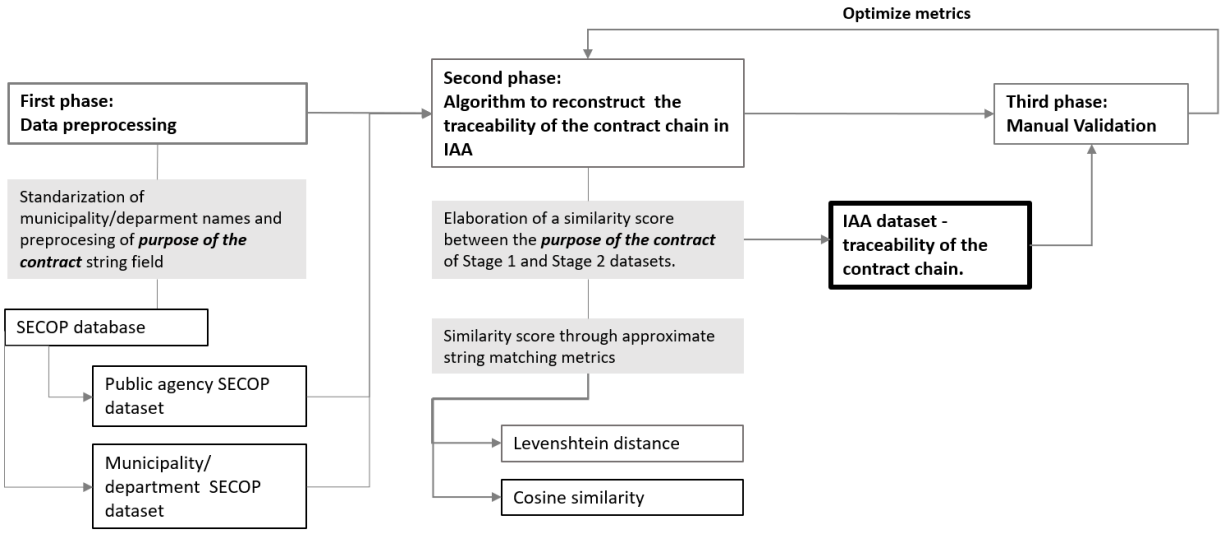


Fig. 3: Graphical representation of the proposed methodology.

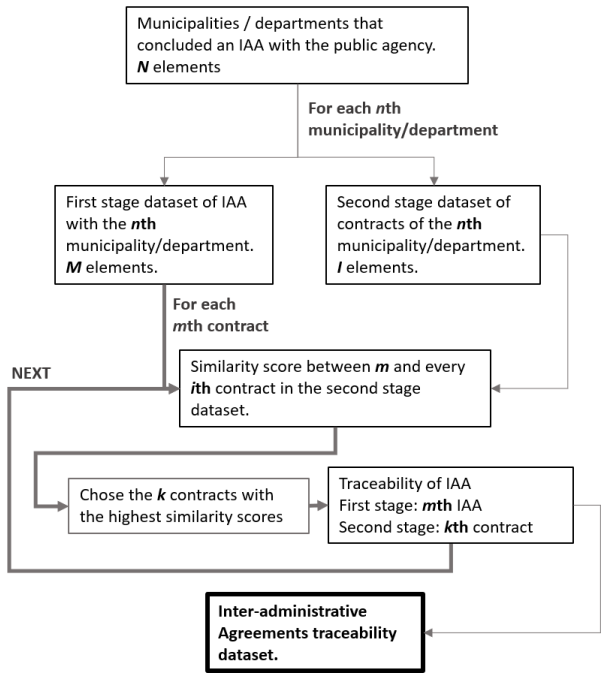


Fig. 4: Graphical representation of the second phase in our methodology between the first and second stage datasets.

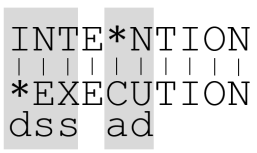


Fig. 5: Example of the calculation of the Levenshtein distance between two strings.

concentrated are the occurrences of a given word in a series of documents [23]. TF of the term i in document j is defined as follows:

$$TF_{i,j} = \frac{f_{i,j}}{\max_k f_{k,j}}$$

where $f_{i,u}$ is defined as the frequency of the term i in the document u, divided by the maximum number of occurrences of any term in the same document. Inverse Document Frequency (IDF) diminishes the weight of very frequent terms (e.g. stop words) and increases the weight of words that do not occur frequently. If term i appears in n_i of N total documents then $IDF_i = \log_2(N/n_i)$.

The TF-IDF score for the term i in the document j is defined by $w_{i,j} = TF_{i,j} \times IDF_i$. The terms with the highest TF-IDF score are often the terms that best characterize the topic of the document [23]. The associated term vector of TF-IDF scores for a document j is $\mathbf{u}_j = [w_{1,j}, w_{2,j}, ..., w_{N,j}]$. The process of representing a document as a term vector is also known as *vectorization* [21].

Consider a document containing 100 words in which the word "execution" appears 3 times. The TF for execution is then $(3/100) = 0.03$. Assuming we have 10 million documents and the word "execution" appears in one thousand of these, the IDF is calculated as $log(10,000,000/1,000) = 4$. Thus, the TF-IDF weight is the product of these quantities: $0.03 \cdot 4 = 0.12$.

For implementing TD-IDF we used `Scikit-learn` Python library, where the vectorization function returns a sparse CSR (Compressed Sparse Row format) matrix as most tokens in the corpus will not appear in the documents [21].

1) N-grams

In vectorization, each document is tokenized to construct the associated term vector. A very common vectorization process for documents is the *bag-of-words* approach. In

this approach, we look at the histogram of the words within the text, i.e. considering each word count as an individual token occurrence frequency in the term vector [12]. That is each position in the term vector represents a word in the document. This method completely ignores the relative position information of the words in the document [21], and therefore the context of the position of the words. A common method to allow bag-of-words to capture more context from the content is called *bag of n-grams*, were each token (or gram) is a n-group sequence of contiguous items -words or letters [15]. Using an n-group sequence of letters allows the vectorization to be more flexible with misspellings rather than with the document context [18].

C. Cosine similarity

Cosine similarity (CS) is a similarity metric between two non-zero vectors that measures the cosine of the angle between them. Cosine similarity between two documents j and k is defined as the cosine of the angle between their term vectors **u** and **v** [26]. Term vectors having similar orientation will have scores closer to 1 (cos 0) indicating the vectors have an almost perfect similarity. An example of a term vector measure (and the one used in this paper) is TF-IDF. Cosine similarity is obtained by the following formula:

$$cs(u,v) = \cos(\theta) = \frac{\mathbf{u}\cdot\mathbf{v}}{\|\mathbf{u}\|_2\|\mathbf{v}\|_2} = 1 - \frac{\sum_{i=1}^{n} u_i v_i}{\sqrt{\sum_{i=1}^{n} u_i^2}\sqrt{\sum_{i=1}^{n} v_i^2}}$$

where $cs(u,v)$ denotes the Cosine distance between the vectors **u** and **v**. The euclidean norm is used to calculate the CS between two vectors. Cosine similarity implementing TF-IDF as the vectorization method is computationally efficient due to high sparsity of most vectors [4] and provides a lower computational complexity than Levenshtein distance for large datasets. We used a cosine similarity open-source code provided by the ING Wholesale Banking Advanced Analytics team [29] to obtain the similarity string metric. The algorithm multiplies the sparse matrix **u** and **v**, and returning only the K highest matches for each document.

Figure 6 shows an simplified example of CS with a TD-IDF vectorization approach, choosing a tokenization of 3-grams by characters. For illustration purposes, we show a two-dimensional graphic. If we vectorize the documents "execution of the contract" and "execution of first contract", the angle between both vectors is $15°$, and their similarity score is 96.59%.

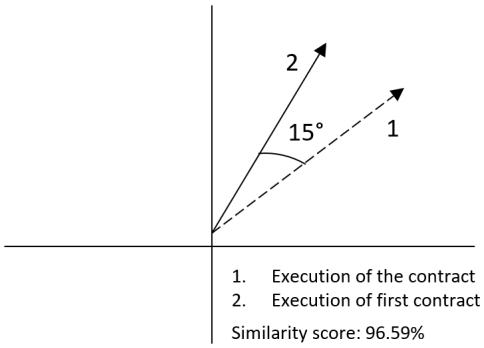


Fig. 6: Example of the calculation of the Cosine similarity metric between two strings.

VI. Results

To test our methodology, we evaluated the IAA dataset of the public agency *Instituto Nacional de Vías* (INVIAS), a public agency in charge of allocating, regulating and supervising contracts for highway and roads construction and maintenance. We selected INVIAS as our study case because the IAA with the highest contracting values are performed by this public agency. Furthermore INVIAS is responsible for the road infrastructure in the country, which affects region competitiveness and life standards.

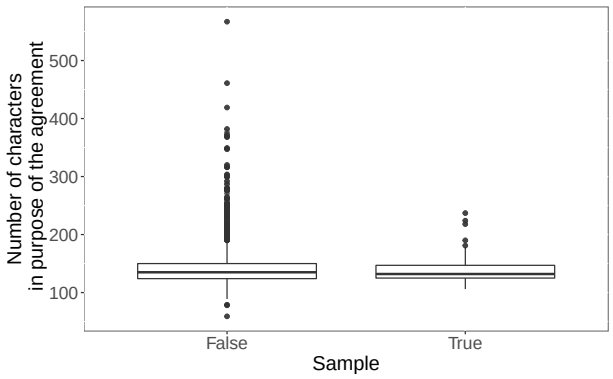
In our study we set a contracting time window from 2012 to 2017. As mentioned earlier in the document, SECOP database is available from 2009, however data before 2012 is considered of low-quality, and therefore is not included in the study. We took a random sample of 100 contracts from a total of 3026 IAA to manually validate and perform speed tests to both approximate string matching algorithms (from now on referred as algorithms). We tested the random sampling according to the number of characters in each purpose of the contract. LD and CS algorithms have a computational complexity proportional to the product of their lengths [22, 32]. We generated a box-plot and a one-way ANOVA to determine if there is a significant difference in the number of characters of the description of the contract among the sample and the population. The results are shown in Figure 7 and Table II. With a p-value of 0.13, we guaranty a random sampling of the population. We then performed manual validation and speed tests to our sample data set, to determine the algorithm that best fits our context.

TABLE II: One-way ANOVA of the sample data versus the population of the number of characters on the purpose of the contract.

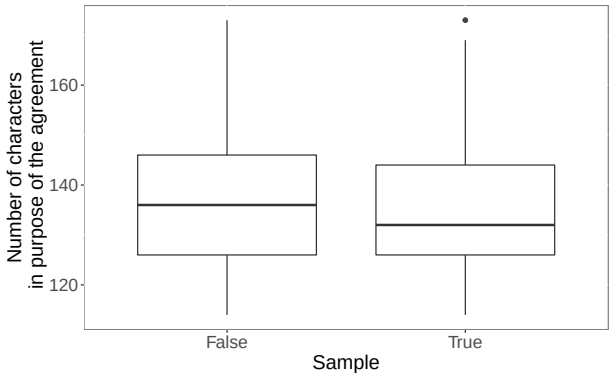
	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Sample	1	2519.22	2519.22	2.23	0.1355
Residuals	4212	4758824.48	1129.83		

A. Validation

To understand how each method performs in detecting duplicates and tracing IAA, we compared each result of the sample to hand-coded tracing analysis of the 100 sample IAA. We identified duplicates in our first stage database, and then read each INVIAS IAA and manually assigned them the corresponding contracts, based on its content. These results were then compared to the outcomes of both algorithms. In this case we report the first and the second highest rated contracts (i.e. the two most similar contracts) for each IAA, as the manual tracing for INVIAS only had a



(a) Box-plot with outliers.



(b) Box-plot without outliers.

Fig. 7: Boxplot of the "purpose of the contract" variable length in the sample data set versus the population, according to the number of characters in each purpose of the contract.

maximum of two contracts associated for an IAA. However we emphasize that our methodology allows a modification of the number of reported contracts, as stated in Section IV. In Table III we provide an example of an IAA, the algorithm outcome and similarity score for each associated contract. We also report if the contract matches our manual traceability. Vectorization in CS algorithm is performed by tri-grams, the *de facto* standard in language modelling research [13].

To evaluate when and whether these algorithms fail to construct the contract chain of IAA we manually checked each agreement, testing if one or both of the highest-rated contracts match the manual-checked contracts. Additionally we reported the score of the two matching contracts for each IAA and their ID to compare the behaviour of the two algorithms.

1) Accuracy performance

The results of the accuracy tests are shown in Table IV. We obtained satisfactory levels of accuracy for both algorithms. The table shows than LD always identifies duplicates, while CS does not identify one duplication. In IAA traceability CS is more accurate than LD. For instance, LD incurs in a 6% error when constructing the contract chain (neither the first nor the second highest-rated contract corresponds to the IAA), while CS always

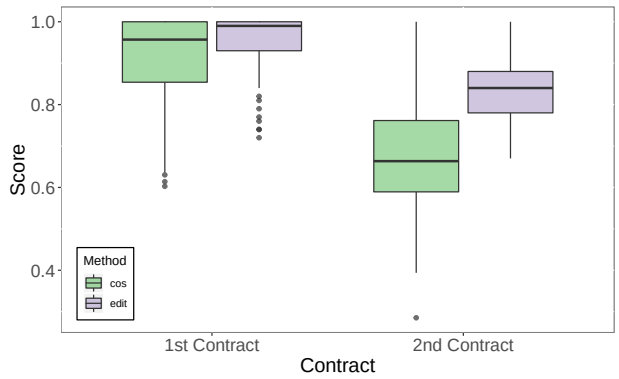


Fig. 8: Box-plot of differences in scores between CS and LD algorithms when choosing first and second contracts.

identifies the contract, either as the first or second highest-rated one. Regarding the highest-rated contract, for CS in 92% of the cases the first contract is part of the IAA, while in LD this percentage only reaches 85%. The second contract identified by CS and LD coincides with the actual IAA in 11% of the times, where in 8% and 9% of the cases respectively only the second contract is part of the IAA. When evaluating the 3 agreements where both the first and the second contracts are part of the traceability the score difference between the contracts differs from 5% to 20%, so a similarity threshold to establish if both contracts are associated with the same agreement is discarded.

The algorithms also manage to identify contracts that cover 2 or more IAA. An example is described below:

- **Purpose of the contract:** Execution of IAA A and IAA B.
  - IAA A: Execution of road X.
  - IAA B: Execution of road Y.

In our sample 8 IAA were covered by 3 contracts. For CS, in 7 cases the algorithm identifies the contract as the highest-rated and in the remaining case the second contract corresponded to the IAA. For LD, 5 of the 8 contracts corresponded to the first contract, while 3 of them were not identified by the algorithm.

2) Measuring score differences

To evaluate the differences between scores in both algorithms, we generated a series of graphs that compare the rating between methods. These are seen in Figures 8 and 9. In Table IV we see that the mean similarity score differs significantly between algorithms by 10% and 20% respectively, where LD always has the highest mean scores. The figures show that not only the mean score varies, also the scores distribution for each algorithm changes. For both first and second highest-rated contracts, LD scores are higher than CS ones. We conclude that CS rating is more strict than LD, as the similarity scores are greater in the latter.

Results show that in each case CS is more accurate than LD, although both algorithms present high validation metrics. Despite the accuracy results surpass 90% in CS, we recommend a manual validation of each IAA outcome, as

TABLE III: Example of highest rated contracts for an IAA with LD and CS algoritm, along with their similarity scores and correspondance to the IAA.

Purpose of the IAA	Contract ID	Purpose of the contracts	LD score	CS score	Correspondance to IAA
IMPROVEMENT AND MAINTENANCE OF THE ROUTE PUELENJE - EL CHARCO CODE 16126 MUNICIPALITY OF POPAYAN DEPARTMENT OF CAUCA	15-1-146400-0	IMPROVEMENT AND MAINTENANCE OF THE ROUTE PUELENJE - EL CHARCO CODE 16126 MUNICIPALITY OF POPAYAN DEPARTMENT OF CAUCA	0.98	0.94	Yes
	14-11-2702358	IMPROVEMENT MAINTENANCE AND CONSERVATION OF THE CORREGIMIENTO OF CALIBIO IN THE MUNICIPALITY OF POPAYAN DEPARTMENT OF CAUCA	0.76	0.72	No

TABLE IV: Metrics for the validation of CS and LD algorithms, with a random sample of 100 contracts.

	Cosine similarity	Levenshtein Dis.
Duplicated	0.99	1
First corresponds	0.89	0.83
Second corresponds	0.08	0.09
None corresponds	0	0.06
Both correspond	0.03	0.02
Mean score first contract	0.84	0.95
Mean score second contract	0.63	0.83

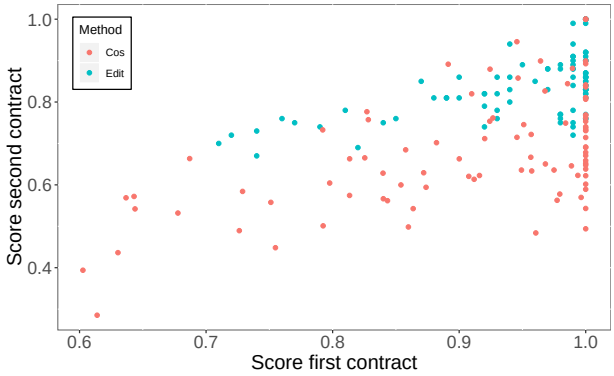


Fig. 9: Scatter plot of first and second contract scores between CS and LD algorithms.

different scenarios arise when determining the traceability of the IAA (only the first contract corresponds to the IAA, both contracts correspond, etc.).

B. Analysis of performance

As SECOP database contains more than 6 million entries, computational speed is crucial to scale our methodology. We performed efficiency tests to our sample data profiling the time performance of each algorithm with the same IAA dataset. Results are shown in Table V. CS is approximately 2.5 times faster than LD in our sample dataset. In CS, vectorization of the purpose of the contract covers 61.41% of CS total execution time, while matrix multiplication takes 37.20%, and abstraction of the two highest rated contracts from the resulting sparse matrix to a dataframe takes 1.38% of the time.

TABLE V: Running time (s) of Cosine Similarity and Levenshtein Distance algorithms for sample data set.

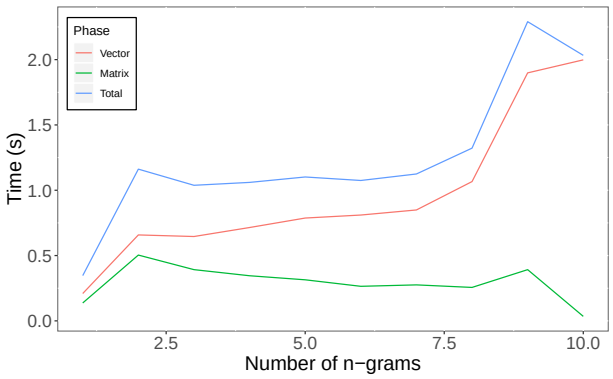
Algorithm	Total time	Average time per run	Percentage of total time
Cosine similarity	**103,322**	**1,086**	**-**
Vectorization	63.447	0.667	61.41
Matrix	38.429	0.404	37.20
Table	1.446	0.015	1.38
Levenshtein distance	**255.62**	**2.75**	**-**

As vectorization and matrix multiplication take 98% of CS completion time, we perform a sensitivity analysis for the number of n-grams chosen to vectorize each document. We performed the analysis from 2 to 10 n-grams, and included vectorization by words (bag-of-words approach). Results are shown in Figure 10.

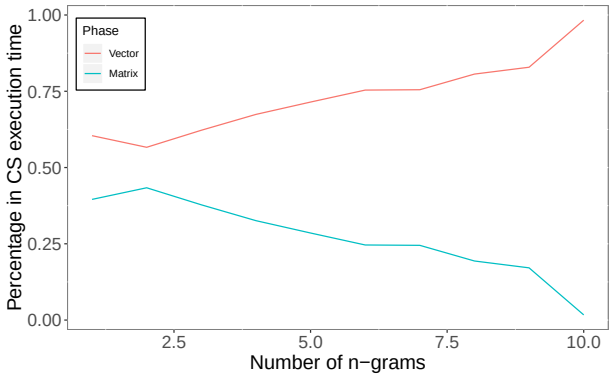
As seen in Figure 10(a), we found that the higher the n-gram the greater the average vectorization time, the lower the average matrix multiplication time and the higher the execution time overall. The lowest average execution time (64.5 s) is achieved with tri-gram vectorization. We also computed the time percentage of vectorization and matrix multiplication in the total CS execution time, shown in Figure 10(b). As n-gram increases, vectorization covers a higher percentage of the total CS time. The increase in execution time relates to the increment of the algorithm complexity [13] as the increment of n-grams produces a more sparse matrix, which increases the space and time computational complexity [15]. These analysis confirm our election of tri-grams vectorization to execute CS.

The number of contracts to be analyzed also affects the speed execution. Figure 11 illustrates the running time behaviour of CS and LD as the number of contracts increases (the number of IAA to match remains equal). Both CS and LD grow linearly, however the slope in LD is much higher than in CD. For example for 7800 contracts CS is executed in 1.90 s, while with LD it takes 7.69 s. In this case LD is completed in roughly four times the CS running time. Time gap between CS and LD increases as the number of contracts grows.

Figure 12 illustrates the relationship between the number of municipality contracts and the running time of vectorization and matrix multiplication in CS. Evaluating CS times, we see that until approximately 2000 contracts vec-



(a) Average execution time.



(b) Percentage of total execution time.

Fig. 10: Relationship between vectorization and matrix multiplication times and number of n-grams chosen for vectorization of words in CS algorithm.

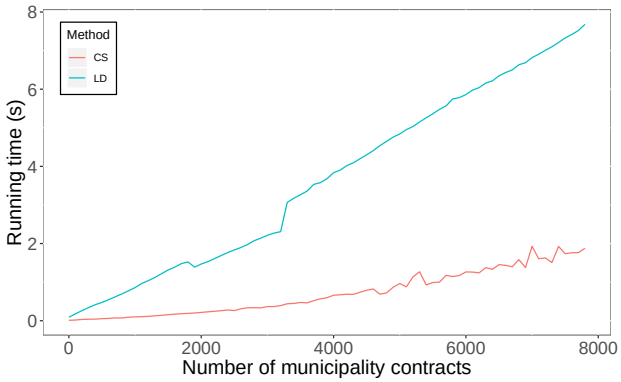


Fig. 11: Relationship between number of municipality contracts and running time of CS and LD algorithms.

torization is the highest running time, after which matrix multiplication oversteps vectorization execution time. We conclude that this behaviour is due to the fact that the matrix sparsity increases in a higher rate than the number of tri-grams, and as mentioned earlier the computational complexity to compute the matrix multiplication increases [13]. Furthermore, matrix running time presents an exponential growth, while vectorization running time presents a linear growth.

The benefits of using CS in our context are evident as

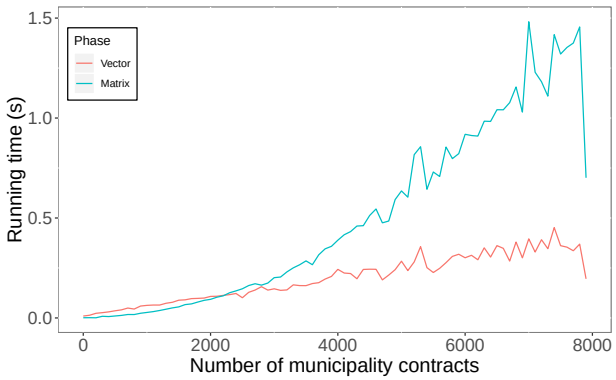


Fig. 12: Relationship between number of municipality contracts and running time of vectorization and matrix multiplication in CS algorithm.

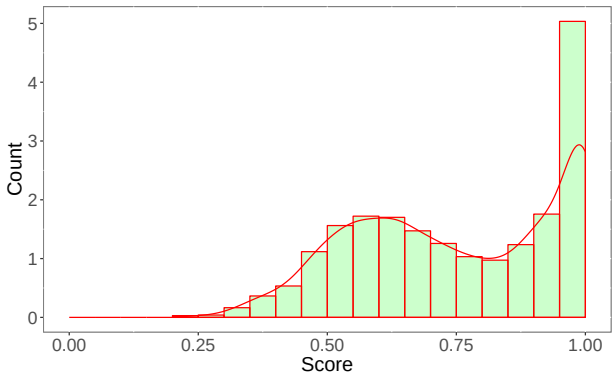


Fig. 13: Histogram of scores of INVIAS IAA dataset obtained with CS algorithm.

they pay off in terms of accuracy and time execution. In Section VI.A we demonstrate that CS accuracy metrics are better than with LD, and time speed advantages of CS is shown in Section VI.B. We proceed to scale our algorithm with a TD-IDF vectorization, based on tri-grams, and CS as our similarity measure to the 3026 contracts in INVIAS dataset.

Figure 13 and 14 illustrate the histogram of the scores and the times in the execution of the algorithm for the entire IAA dataset. The total execution time for the entire dataset was 19138.52 seconds (5.33 hours), where in average data preprocessing covered 25.5% of the total time and the running algorithm time covered 74.5%.

VII. Conclusions and Further Work

Colombian government has developed open government data initiatives to improve transparency and trust in public administration agencies. However, the citizenship and private sector involvement is minimum. Although there exist valuable open databases like SECOP, the difficult access and interpretation of the data discourages the its usage. Aiming at improving citizenship involvement and audit in public decision-making processes, governmental and non-governmental actors have begun to implement initiatives in order to encourage open data usage.

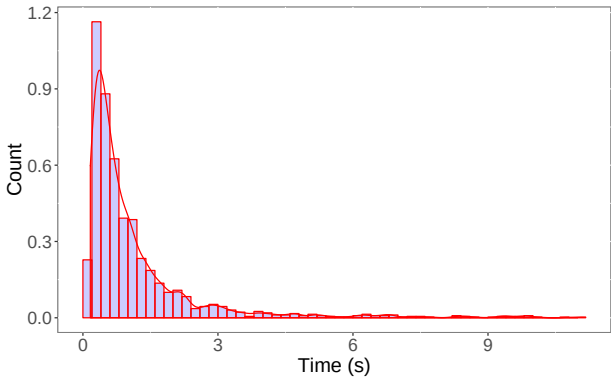


Fig. 14: Histogram of running time of INVIAS IAA dataset with CS algorithm. (excluding the 5% and 95% quantiles for illustration purposes)

In this paper, we develop a methodology to reconstruct the contract chain of inter-administrative agreements by comparing two approximate string matching techniques: Levenhstein distance and cosine similarity. Analyzing the results we find that cosine similarity is the algorithm that best fits our context, therefore we propose a traceability methodology implementing cosine similarity. Though a manual traceability of the agreements can be performed in SECOP, our algorithm provides a tool to achieve macroscopic analysis on inter-administrative agreements. Furthermore, the resulting dataset of our methodology allows an easier follow-up and audit of inter-administrative agreements. A future implementation of an open dashboard can be developed with our resulting dataset, aimed to increment the participation of the citizenship, the private sector and other decision-making actors.

Although the algorithm itself presents satisfactory results in the reconstruction of the contracting chain, a latter manual revision of the results is needed to guaranty the accuracy of the traceability. Furthermore, the manual validation will be useful to constantly improve the methodology and especially the algorithm.

Our methodology can be scaled to the other public agencies in Colombia. In a further research we plan to execute a future validation with other public agencies to straighten the effectiveness of our methodology. Finally, with the reconstruction of the contracting chain further analysis can expand the scope on contracting processes. An study of social networks between contractors and State actors such as mayors, governors and senators can give a bigger perspective in the public contracting landscape of Colombia. Likewise our results can be assembled with benchmarks such as a risk index in public procurement. These can be used to identify unusual or risky contracts or actors, aiming at detecting unusual processes or even corruption networks between state actors and the private sector.

REFERENCES

[1] *2018 Edelman Trust Barometer Global Report*. Tech. rep. URL: https://cms.edelman.com/sites/default/files/2018-01/2018%20Edelman%20Trust%20Barometer%20Global%20Report.pdf.

[2] Alexandr Andoni and Robert Krauthgamer. "The Computational Hardness of Estimating Edit Distance". In: *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*. IEEE, Oct. 2007, pp. 724–734. ISBN: 0-7695-3010-9. DOI: 10.1109/FOCS.2007.60. URL: http://ieeexplore.ieee.org/document/4389540/.

[3] David Banisar. "Freedom of Information Around the World 2006: A Global Survey of Access to Government Information Laws". In: *SSRN Electronic Journal* (Sept. 2006). ISSN: 1556-5068. DOI: 10.2139/ssrn.1707336. URL: http://www.ssrn.com/abstract=1707336.

[4] Mikhail Bilenko. "Learnable Similarity Functions and Their Applications to Record Linkage and Clustering". PhD thesis. Department of Computer Sciences, University of Texas at Austin, 2003, p. 136. URL: http://www.cs.utexas.edu/users/ai-lab/?bilenko:phd06.

[5] Agusti Cerrillo-i-Martínez. "The regulation of diffusion of public sector information via electronic means: Lessons from the Spanish regulation". In: *Government Information Quarterly* 28.2 (Apr. 2011), pp. 188–199. DOI: 10.1016/J.GIQ.2010.05.009. URL: https://www.sciencedirect.com/science/article/pii/S0740624X10001309.

[6] Colombia Compra Eficiente. "Estatuto General de Contratación que rige el Sistema de Compra pública". In: (2018). DOI: 10.1787/9789264202177-10-en. URL: http://dx.doi.org/10.1787/9789264202177-10-en.

[7] Colombia Compra Eficiente. *SECOP I — Colombia Compra Eficiente*. URL: https://www.colombiacompra.gov.co/secop/secop-i.

[8] Comisión Económica para América Latina y el Caribe. *Gobierno abierto*. URL: https://www.cepal.org/es/temas/gobierno-abierto.

[9] European Commission. *European Governance - A White Paper*. Tech. rep. Brussels: European Commission, 2001, p. 35. URL: https://ec.europa.eu/europeaid/sites/devco/files/communication-white-paper-governance-com2001428-20010725_en.pdf.

[10] Departamento Nacional de Planeacion. *TerriData*. 2018. URL: https://terridata.dnp.gov.co/#/.

[11] *Gobierno Abierto en América Latina*. Estudios de la OCDE sobre Gobernanza Pública. OECD Publishing, Feb. 2015. ISBN: 9789264225770. DOI: 10.1787/9789264225787-es. URL: http://www.oecd-ilibrary.org/governance/gobierno-abierto-en-america-latina_9789264225787-es.

[12] Yoav Goldberg. *Neural network methods for natural language processing*, p. 287. ISBN: 1627052984.

[13] Joshua Goodman. “A Bit of Progress in Language Modeling”. In: (Aug. 2001). URL: http://arxiv.org/abs/cs/0108005.

[14] Dan. Gusfield. *Algorithms on strings, trees, and sequences : computer science and computational biology*. Cambridge University Press, 1997, p. 534. ISBN: 0521585198.

[15] Dan Jurafsky and James H. Martin. *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition*. Dorling Kindersley Pvt, Ltd, 2014, p. 940. ISBN: 9789332518414.

[16] Vasiliki Karageorgou Lecturer in European Administrative law and European Environmental Law. *Transparency principle as an evolving principle of EU law: Regulative contours and implications*. Tech. rep. URL: www.connex-network.org/eurogov..

[17] Christopher D. Manning, Prabhakar. Raghavan, and Hinrich. Schutze. *Introduction to information retrieval*. Cambridge University Press, 2008, p. 482. ISBN: 0521865719.

[18] Christopher D. Manning and Hinrich. Schuuze. *Foundations of statistical natural language processing*. MIT Press, 1999, p. 680. ISBN: 0262133601.

[19] Ministerio de Hacienda. *Portal de Transparencia Económica*. URL: http://www.pte.gov.co/WebsitePTE/.

[20] Diaz-Cruz Nicolas. *100 Dias del gobierno Duque. Estamos construyendo pais?* Conversatorio. Bogota, 2018.

[21] Fabian Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12.Oct (2011), pp. 2825–2830. ISSN: ISSN 1533-7928. URL: http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html.

[22] Giovanni Pighizzini. “How Hard Is Computing the Edit Distance?” In: *Information and Computation* 165 (2001), pp. 1–13. DOI: 10.1006/inco.2000.2914. URL: https://ac.els-cdn.com/S0890540100929146/1-s2.0-S0890540100929146-main.pdf?_tid=dcdd7fde-d2f5-44a0-9401-fe4c7e8efa6e&acdnat=1542752571_007de7ab5274d8bc98300397c5afe2e5.

[23] Anand. Rajaraman and Jeffrey D. Ullman. *Mining of massive datasets*. Cambridge University Press, 2012, p. 315. ISBN: 1107015359.

[24] Jeannine E. Relly and Meghna Sabharwal. “Perceptions of transparency of government policymaking: A cross-national study”. In: *Government Information Quarterly* 26.1 (Jan. 2009), pp. 148–157. ISSN: 0740-624X. DOI: 10.1016/J.GIQ.2008.04.002. URL: https://www.sciencedirect.com/science/article/pii/S0740624X08000877.

[25] Republica de Colombia. *Ley 1150 de 2007*. 2007.

[26] Dipanjan Sarkar. *Text Analytics with Python*. 2016. ISBN: 978-1-4842-2387-1. DOI: 10.1007/978-1-4842-2388-8.

[27] SeatGeek. *FuzzyWuzzy*. https://github.com/seatgeek/fuzzywuzzy. 2018.

[28] Sistema General de Regalias. *Mapa Regalias*. 2018. URL: http://maparegalias.sgr.gov.co/#/.

[29] Zhe Sun. *Boosting selection of the most similar entities in large scale datasets*. 2017. URL: https://medium.com/wbaa/https-medium-com-ingwbaa-boosting-selection-of-the-most-similar-entities-in-large-scale-datasets-450b3242e618.

[30] Transparencia por Colombia. *Monitor Ciudadano de la Corrupción*. URL: http://www.monitorciudadano.co/.

[31] United Nations. *UN E-Government Knowledgbase*. 2018. URL: https://publicadministration.un.org/egovkb/en-us/Data/Region-Information/id/14-Americas---South-America.

[32] Damir Vandic, Flavius Frasincar, and Frederik Hogenboom. *Scaling Pair-Wise Similarity-Based Algorithms in Tagging Spaces*. Tech. rep. URL: http://damirvandic.com/wp-content/papercite-data/pdf/icwe_2012.pdf.

[33] Leonardo Villar et al. *Lucha Integral Contra la Corrupcion en Colombia: Reflexiones y Propuestas*. Ed. by Leonardo Villar and Daphne Alvarez. Bogota, Colombia: Fedesarrollo, 2018, p. 374. ISBN: 978-958-56558-4-3. URL: www.fedesarrollo.org.co.