ISSN 2007-9737

Tokenizer Adapted for the Nasa Yuwe Language

Luz Marina Sierra Martínez[1], Carlos Alberto Cobos Lozada[2], Juan Carlos Corrales[1]

[1] University of Cauca, Telematics Engineering Group (GIT), Popayán, Colombia

[2] University of Cauca, Information Technology Research Group (GTI), Popayán, Colombia

{lsierra, ccobos, jcorral}@unicauca.edu.co

Abstract. In Colombia, ethnic and cultural diversity is conceived by the government to be a social right. Such diversity finds expression, among other ways, in a large number of indigenous languages, which have been kept alive for centuries. However, efforts toward conservation and preservation of these languages have generally fallen short. This is the case for the Nasa Yuwe language, spoken by the Nasa, or Páez, indigenous community, the status of which is endangered. Given such a predicament, the use of technology has been found to provide a strategic opportunity for adaptation, ownership, and development of Nasa Yuwe within the social and cultural environment of the Nasa people. The technology includes the use of computational techniques, which allow the exchange of information by means of IR activities. These encourage different, new possibilities for the Nasa people to be able to interact in Nasa Yuwe. It has therefore become necessary to adapt the stages of the IR process to this language. The current paper specifically presents a process for adapting a tokenizer to texts written in Nasa Yuwe. This involves making use of the precision-recall curve as an evaluation and comparison measure. The results presented allow appreciation of all stages in the process of adapting the standard tokenizer to produce the Nasa version, of the Nasa tokenizer and its results over texts written in Nasa Yuwe, and of the analysis of the precision-recall curve baseline in contrast to that of the Nasa tokenizer.

Keywords. Nasa indigenous community, Nasa Yuwe language, tokenizer for Nasa Yuwe, information retrieval for texts written in Nasa Yuwe.

1 Introduction

Nasa Yuwe is one of 64 official languages of the Republic of Colombia. It is spoken by the Nasa, or Páez, indigenous community [1]. The language has recently been classified as independent and currently in the process of description. It is still not entirely clear in what way many aspects of its grammar, syntax, morphology, semantics, and of other linguistic layers are to be seen in order to analyze this language in its written variant [4]. This is due to the fact that Nasa Yuwe has always been a language of oral tradition. Only in 1964 an alphabet was first proposed for writing purposes. But only in year 2000 a unified alphabet became available. However, still there do not exist clear and specific rules for writing, i.e. writing of some words and structuring of sentences depend on the spoken variant of the language. The language is highly influenced by cultural, social, and political factors related to the Nasa community.

The main objective of this study is to demonstrate that it is possible to adapt an information retrieval task to process texts written in Nasa Yuwe. The adaptation of a standard tokenizer that takes Nasa Yuwe language characteristics into account is proposed. This will include the process followed, and the results obtained, using the tokenizer adapted for Nasa Yuwe, thus representing a major advance in the process of building an Information Retrieval (IR) system for texts written in Nasa Yuwe. It is expected to offer a tool that allows Nasa people to interact in a computational environment in their own language, encouraging writing and thereby contributing to the visibility of the language by means of computational tools.

The rest of the paper is organized as follows. Section 2 presents a background on IR for texts,

the tokenizer, basic information about the Nasa Yuwe language, and a brief description of a test collection for Nasa language IR systems. Section 3 describes the process of adapting the tokenizer for Nasa Yuwe. Section 4 presents an evaluation of the results. Finally, Section 5 offers conclusions and outlines future work intentions.

2 Background

2.1 Information Retrieval (IR)

Information retrieval has received much attention of the research community given the importance of providing the users with organized and relevant information about a desired topic [5]. The purpose of an Information Retrieval System (IRS) is to find those documents that meet the user information need, expressed in the form of a query [6, p. 221]. The "ideal" response of an IRS consists only of documents relevant to the query [7]. Hence the primary aims of IRS are [8, p. 17]: to maximize the probability that the query recovers relevant elements, and to produce the best ranking possible according to the relevance probability estimates [3, p. 446].

Some of the important references that have made it possible to analyze strategies used for adapting each IR phase are as follows.

In 2013, Zhang and Zhan [9] presented the construction of a retrieval system based on Lucene. The architecture and components of Lucene used for the construction of Compass Framework are described therein.

In 2012, ARGOsearch [10] was presented, which is an IRS that improves the quality of search in a Library Management System (LMS), supported on a general and extensible architecture based on relevance criteria that orders the results of user queries by considering the similarity of texts, user profile, and statistical information collected by the LMS.

In 2011, Cui, Chen, and Li [11] presented the construction of an IRS for Chinese based on the Lucene model and it is applied to the retrieval of research data of a forestry information system, finding the system to be fast and efficient.

In 2004, Hollink et al. [12] presented research done on eight European languages (Dutch, English, Finnish, French, German, Italian, Spanish and Swedish), which describes the IR tasks and how they were applied to these languages, obtaining good results.

2.2 Tokenizer

Tokenization is based on the recognition of lexical units of a text. A document (text) is introduced as input and a list of lexical units, or tokens, is obtained as output [13]. It is one of the first tasks carried out when performing the pre-processing of text [14]. In most cases, tokens match the corresponding words. However, they comprise the so-called lexical units from the text and are known as tokens. Likewise, programs responsible for carrying out this lexical decomposition are called tokenizers [13].

A tokenizer has two important properties: (1) It is the only component that requires to be used in any application of natural language processing that involves analysis tasks; and (2) it is always used at the beginning of the analysis process [15].

Some important studies that reveal the different uses and strategies for text tokenization are as follows.

In 2015, Hammo, Yagi, et al. [16] described how an IR process can be used for Arabic, studying the semantic changes of the language in relation to the frequency of use of a word over time and how it changes its meaning over time. The tokenizer implemented for this processing provides two types of output: tokens that correspond to lexical units whose characters are recognizable, and tokens that need more morphological analysis, while tokens of one character in length were not taken into account.

In 2011, Jamil et al. [17] presented a system that initiates the retrieval process with query transformation using tokenization, stemming (Fatimah) and stop-word removal (Bahasa Melayu morphological rules). The IRS was tested with 30 documents and was able to operate effectively.

In 2008, Bijankhan et al. [18] discussed linguistic aspects taken into account in building a Corpus in Persian and reducing problems when making a suitable tokenization process specific to Persian.

In 2006, Jiang and Zhai [19] presented the generalization of various tokenization strategies in

Table 1. Attributes of the Nasa Yuwe collection. Source: [25]

Attributes	Value
Collection size	113 KB
Document formats	Text (UTF-8)
No. of documents	97
Terms by document average	103
Corpus size in numbers of words	9955
Average document size	1.5 KB
No. of queries	8

a set of heuristics organized to be applied to IR in biomedicine, concluding that the removal of non-functional characters is safe and different types of tokenization can be applied.

2.3 Nasa Yuwe Language

Nasa Yuwe is the most important ethnic language in Colombia. It is spoken in various Nasa reservations and settlements across a number of municipalities in the Departments of Caquetá, Putumayo, Meta, Tolima, Valle del Cauca, Cauca, and Huila [20]. The Nasa population is currently estimated to total around 200,000 people, of which 75% are active speakers of the Nasa language [2]. Nasa Yuwe is in danger of extinction. It has been losing ground due to a range of cultural, social, geographical, and historical factors [2].

The alphabet, unified only as recently as 2000, consists of 32 vowels (oral, interrupted oral, aspirated oral, long nasal oral, interrupted nasal, and aspirated nasal) and 61 consonants (basic, aspirated, palatalized, nasal, fricative, approximant, among others). The different variants are associated with the geographical area in which the reserves of each community are located. An approximate guide identifies the different variants of Nasa as [21]: Northern region (Munchique – Tigres, Huellas, Toribío, San Francisco, and Tacueyó); Central region (on one side, Jambaló and Pitayó, and on the other side, Caldono, Pioyá, and Pueblo Nuevo); southern region (Novirao and Paniquitá); and the interior region.

The main works to describe the language are as follows. “Paéz or Nasa Yuwe Grammar” [22] describes the phonology, morphology, and syntax of Paéz or Nasa Yuwe. “A Nasa Yuwe—Spanish Dictionary” [23] contains vocabulary for Nasa editors and gives a description of Nasa Yuwe based on Spanish grammar. “A look at the Nasa Yuwe of Novirao” [24] provides a phonological and morphological description of Nasa Yuwe in Novirao and draws a comparison with other variants. Finally, “The Paéz language—an overview of its grammar” [21] presents a more detailed description of the language.

2.4 Test Collection for the Nasa Yuwe Language

A test collection is the most widely used method for evaluating information systems. As such, in the process of building an IR system for Nasa Yuwe it was necessary to have a test collection for such a purpose. The first test collection for Nasa Yuwe [25] was therefore built. Its description is shown in Table 1; it is published at the webpage with the URL http://www.ewa.edu.co/coleccion.

The documents that form this collection were selected following fieldwork. They are written in the unified alphabet and treat topics on the Nasa culture and worldview. The queries were performed by prioritizing the issues most in need of information in reference to the selected texts. The work was carried out with expert speakers and teachers of the language, and finally the queries were obtained. To make the classification of the relevant documents for each query, the judgment of experts for each document-query pair was used, since there were no previous judgements nor pre-selections from other IR systems. The experts used a scale of four levels of relevance in order to return a judgment as follows: very relevant, relevant, not very relevant, and not relevant. In the collection, 63% of the documents were relevant for one or more of the eight queries [17]. It was also established that Nasa Yuwe complies with Zipf’s law [17].

3 Process of Adapting the Tokenizer for Nasa Yuwe

In order to define the tokenizer for Nasa Yuwe, a standard tokenizer (available in Lucene Libray) was adapted to take into account the current

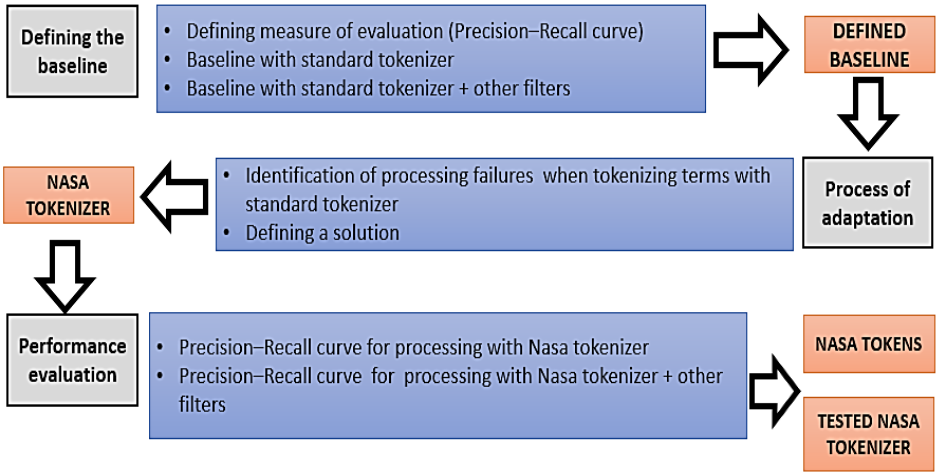

Fig. 1. Description of the process of adapting the tokenizer for Nasa Yuwe

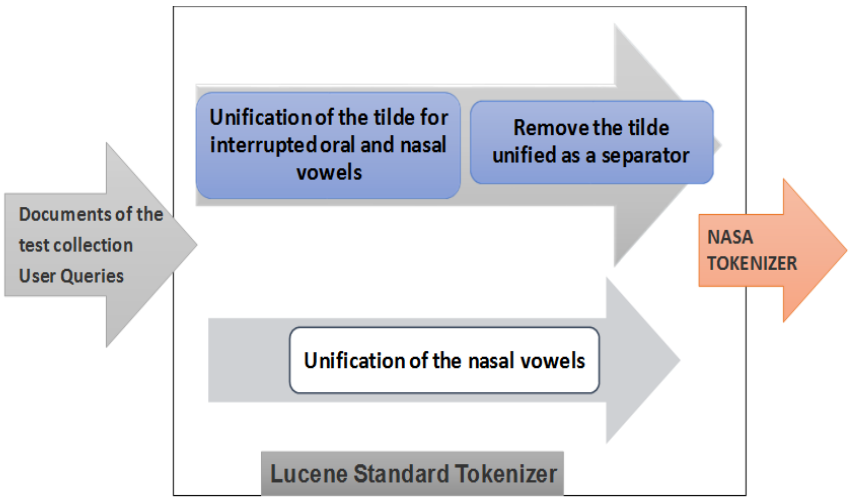

Fig. 2. The resulting Nasa tokenizer (NT)

conditions of this language. In Figure 1, a brief outline of the process used to obtain the Nasa tokenizer is shown.

3.1 Defining the Baseline

To begin, the Lucene standard tokenizer (ST) [26] was selected (version 2.9.4). It separates the words by punctuation characters, removing punctuation marks, but full stops that are not followed by a blank space are considered part of the token; it separates words joined together by hyphens (-) unless there is a number contained in the token, in which case they are not separated. It also identifies website addresses, emails addresses, and acronyms [27]. Secondly, the baseline was established using the Lucene standard tokenizer, making use of the queries and documents in the test collection [17]. Thirdly, the precision-recall curve was selected as a measure of evaluation, taking into account the result of each of the eight queries defined in the test collection.

Table 2. Tokenized text snippet

Nasa Word	Tokens	Nasa Word	Tokens
u'j	U J	luuçxwe'sxa	luuçxwe sxa
jĩ'ku'tx	jĩ ku tx	u'jweçtene'ta	u jweçtene ta
luuçxyu'	luuçxyu	kxuyyu'	kxuyyu
txãawẽ'sxçxane'ta	txãawẽ Sxçxane Ta	thẽ'sa	thẽ sa

Table 3. Inadequate separation of Nasa tokens

Nasa Word	Processing	No of Tokens	Corrects tokens
tu'ka	tu ka	2	1
a'tete	a tete	2	1

Table 4. Replacement of an interrupted vowel by an oral vowel

Nasa Word	Tokenization
Txanteya'	Txanteya
Sayu'	Sayu

Table 5. Nasal vowel as a separating character

Word Nasa	Tokenization
ẽente'	e ente
Mjĩsa	mji sa
kũhku'ka'wã'	ku hku'ka'wa

Table 6. Data about the errors

Description	Error 1	Error 2	Error 3
Total Occurrences	3936	2531	141
Number of documents that occurs	97	97	7
Maximum occurrences per document	218	182	46
# Words document	515	515	137
Minimum occurrences per document	4	1	1
# Words document	20	17	141
Percentage	59.6	38.3	2.1

Finally, with these elements it was possible to establish the baseline using different processing techniques on texts written in Nasa Yuwe, such as: (1) the Lucene standard tokenizer; (2) a Lucene standard filter used to remove, among other features, the 's word endings and the full stops in acronyms [27]; and (3) a converter of words to lowercase (LC) added to the Lucene standard tokenizer [27].

3.2 Process of Adaptation to Nasa Tokenizer

Once the processing of the test collection texts has been carried out with: (1) ST; (2) ST + Filter; and (3) ST + LC, the difficulties in processing text written in Nasa Yuwe were identified. The aim was then to resolve these difficulties, looking to obtain a Nasa tokenizer that correctly processes texts written in that language.

In Table 2, a list of words from a snippet of a Nasa document from the collection is presented along with the tokens obtained using the standard tokenizer. A high number of errors were seen to occur even in such small snippets. It was therefore necessary to focus attention on identifying the causes of the errors and detecting a solution. These are explained as follows.

1. Inadequate separation (division) of words. Table 3 presents two examples of this. The standard tokenizer on finding a word with an interrupted or nasal oral vowel separates it and replaces the interrupted vowel with an oral or nasal vowel depending on the situation.

   To resolve this difficulty, different types of tilde (´, `, ‘,’) utilized for writing interrupted oral and nasal vowels that could be found in the documents in the collection and in queries entered by the user were identified. All of these were unified into a single (’) in the Nasa tokenizer. That tilde was then removed in the standard Lucene tokenizer as a separating element for words in a text or sentence.
2. Replacement of an interrupted vowel located at the end of the word. Table 4 shows how the standard tokenizer changes the interrupted vowel at the end of the word for an oral vowel.

   To keep the interrupted oral and nasal vowels in the processed tokens, the same steps were taken as above.
3. Arbitrary change of nasal vowels for oral vowels. In Table 5, three Nasa words with nasal vowels are shown that are incorrectly separated by the standard tokenizer, creating a separation of tokens and the replacement by the corresponding oral vowel. This occurred only in some documents in the collection, not in them all.

   To resolve this arbitrary change, the Nasa texts were reviewed and it was found that the people from the Nasa communities use different ways of inserting these special characters and that one of these ways (for example, selecting the character from a page with the International Phonetic Alphabet). It is registered as two characters that being joined together are viewed as a single character.

The Nasa tokenizer therefore unifies these pairs of characters into the character to which they correspond. It was thus necessary to make an initial unification both in the documents in the collection and in the terms of the query. Table 6 presents some data about the errors previously described, highlighting the impact of each one and showing the importance of correcting them.

Figure 2 summarizes the additional steps that the Nasa tokenizer carries out compared to the standard tokenizer.

3.3 Performance Evaluation

With the Nasa tokenizer already implemented, we proceeded to carry out various comparisons for each case of the baseline contrasted with the Nasa tokenizer (NT), determining the quality of the test collection text tokenization and establishing the various conclusions.

4 Results for the Nasa Tokenizer

4.1 Baseline

Figure 3 shows the performance of the standard tokenizer, standard tokenizer with filter, and standard tokenizer with the converter of text to lowercase (LC).

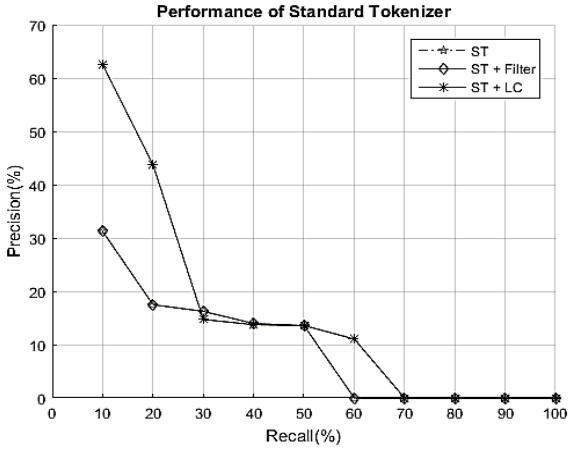

Fig. 3. Baseline for standard tokenizer for Nasa Yuwe

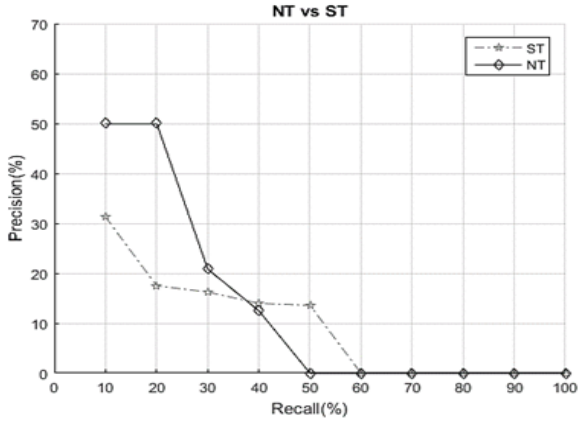

Fig. 4. Comparison between NT and ST

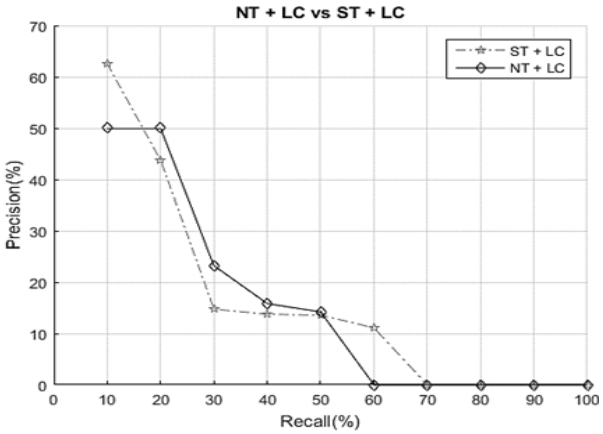

Fig. 5. Comparison between ST + LC and NT + LC

Table 7. Text snippet correctly tokenized

Word Nasa	Token	Word Nasa	Token
u'j	u'j	luuçxwe'sxa	luuçxwe'sxa
jĩ'ku'tx	jĩ'ku'tx	u'jweçtene'ta	u'jweçtene'ta
luuçxyu'	luuçxyu'	kxuyyu'	kxuyyu'
txãawẽ'sxçxane'ta	txãawẽ'sxçxane'ta	thẽ'sa	thẽ'sa

The filter removes endings but preserves Internet addresses, e-mail addresses, some nouns that contain numbers, etc. Figure 3 shows that:

– The performance of the standard tokenizer does not change with adding the filter, which therefore does not change the baseline of the standard tokenizer.
– With adding the converter of written text to lowercase, precision increased substantially for recall values of 10% up to 30%. This is because it improves the comparison of terms in the query with those in the documents. E.g., the word *Wala* is different from the word *wala*, but on applying the LC converter, the two words become indistinguishable.

4.2 Nasa Tokenizer (NT)

First, in the process followed in order to obtain the Nasa tokenizer, several precision-recall curves were analyzed by correcting each of the errors, each curve giving better precision values until Figure 3 was obtained. Figure 4 shows the precision-recall curve that presents the Nasa tokenizer compared with the standard tokenizer over the eight queries on the test collection. In this figure, it can be seen clearly that the Nasa tokenizer obtains better precision values in the first recall values, which is lower than 40%.

The filter was then applied to the Nasa tokenizer and no changes were obtained in the values for the precision-recall curve.

Next, the converter of text to lower case (LC) was applied to the Nasa tokenizer and compared with the standard tokenizer, as shown in Figure 5, where it can be seen that in general the precision-recall curve for NT + LC shows the best precision values, with the exception of the first value (a recall of 10%). The latter is understandable given that as the standard tokenizer separates terms incorrectly, the chances are increased of finding similarities in the words in the queries and the documents. For example, the words a'te, a'te', a'te's, a'tek, and a'tepa', on being incorrectly tokenized (a te, a te, a te s, a te pa) produce greater possibilities of coincidences than in the case of the Nasa tokenizer, where these words are different.

Table 7 provides a summary of the best results that can be obtained with the proposed tokenizer.

5 Conclusion and Future Work

It was possible to carry out the adaptation of one of the phases of the information retrieval process, such as tokenization. The adaptation proved successful in obtaining the correct Nasa tokens. The experience gained from carrying out this work is very valuable, since no similar background work exists for Nasa Yuwe in relation to adaptation of the tasks of lexical analysis (tokenizer). It thus becomes another important step in building an IRS for Nasa Yuwe.

As future work, our research group will continue the development of an IRS for Nasa Yuwe, available via the Web to be able to encourage reading and writing in the language, to the benefit of Nasa community members. In addition, there is keen awareness of the need to continue reviewing other activities and research that will improve the performance of the Nasa tokenizer.

Acknowledgements

We are grateful to the University of Cauca and its research groups GTI and GIT of the Computer Science and Telematics departments and

VicePresident of Research. We are especially grateful to Colin McLachlan for suggestions relating the English text.

References

1. **National Constitutional Assembly of the Republic of Colombia, Bank of the Republic (1990).** http://www.banrep.gov.co/regimen/resoluciones/cp91.pdf
2. **University of Cauca, CRIC—PEBI—General Language Commission. (2008)**. *Sociolinguistic study in preliminary phase.* Database—CRIC 01/2007 Nasa Yuwe and Namtrik language. Popayán
3. **Baeza-Yates, R. (2004).** Challenges in the Interaction of Information Retrieval and Natural Language Processing. *Computational Linguistics and Intelligent Text Processing*, Springer, Berlin Heidelberg, Vol 2945, pp 445–456. DOI: 10.1007/978-3-540-24630-5_55.
4. **Rojas, T.E. (2012).** *Esbozo Gramatical de la lengua Nasa (lengua Paéz).* El Lenguaje en Colombia. Tomo I: Realidad Lingüística de Colombia. UNICEF Bogotá.
5. **Manning, C., Raghavan, C., & Shütze, H. (2009).** *An Introduction to Information Retrieval.* Cambridge University Press.
6. **Larkey, L.S., Ballesteros, L., & Connell, M.E. (2007).** Light Stemming for Arabic Information Retrieval. *Arabic Computational Morphology Text, Speech and Language Technology,* Springer, Vol. 38, pp. 221–243.
7. **Tolosa, G.H. & Bordignon, F. (2005).** *Introducción a la Recuperación de Información.* Universidad Nacional de Luján, Argentina.
8. **Peters, C., Braschler, M., & Clough, P. (2012).** Within-Language Information Retrieval. *Multilingual Information Retrieval,* Springer Berlin Heidelberg, pp. 17–55. DOI: 10.1007/978-3-642-23008-0_2.
9. **Zhang, C. & Zhan, S. (2012).** Research and Implementation of Full-Text Retrieval. *Proceedings of the 2012 International Conference on Communication, Electronics and Automation Engineering.* Advances in Intelligent Systems and Computing, Springer Berlin Heidelberg, Vol 181, pp. 349–356. DOI: 10.1007/978-3-642-31698-2_50.
10. **Borges, E.N., Pereira, I.A., & Tomas, C. (2012).** ARGOSearch: an Information Retrieval System based on text similarity and extensible relevance criteria. *31st International Conference of the Chilean Computer Science Society (SCCC).* Valparaiso. DOI: 10.1109/SCCC.2012.23.
11. **Cui, Y., Chen, Y., & Li, J. (2011).** Research of Information Search Engine in Forestry Based on the Lucene. *Advances in Automation and Robotics, LNEE,* Springer Berlin Heidelberg, Vol 2, pp. 603–609. DOI: 10.1007/978-3-642-25646-2_78.
12. **Hollink, V., Kamps, J., Monz, C., & De Rijke, M. (2004).** Monolingual Document Retrieval for European Languages. *J. Information Retrieval,* Vol. 7, No. 1, pp. 33–52. DOI: 10.1023/B:INRT.0000009439.19151.4c.
13. **Yatso, V.A. (2011).** Methods and algorithms for automatic text analysis. *J. Automatic Documentation and Mathematical Linguistics,* Vol. 45, No. 5, pp. 224–231. DOI: 10.3103/S0005105511050062.
14. **Sproat, R. (2008).** Linguistic Processing for Speech Synthesis. **Benesty, J., Sondhi, M., & Huang, Y. (eds).** *Springer Handbook of Speech Processing,* Springer Berlin Heidelberg, pp. 457–470. DOI: 10.1007/978-3-540-49127-9_22.
15. **Klatt, S. & Bohnet, B. (2004).** You Don't Have to Think Twice if You Carefully Tokenize. *Natural Language Processing - IJCNLP*. LNCS, Springer Berlin Heidelberg, Vol. 3248, pp. 299–309. DOI: 10.1007/978-3-540-30211-7_32.
16. **Hammo, B., Yagi, S., Ismail, O., & AbuShariah, M. (2015).** Exploring and exploiting a historical corpus for Arabic. *Language Resources and Evaluation.* pp. 1–23. DOI: 10.1007/s10579-015-9304-9.
17. **Jamil, N., Jamaludin, N.A., Abdul Rahman, N., & Sabari, N. (2011).** Implementation of Vector-Space Online Document Retrieval System Using Open Source Technology. *Conference on Open Systems (ICOS).* Langkawi, IEEE. DOI: 10.1109/ICOS.2011.6079228.
18. **Bijankhan, M., Sheykhzadegan, J., Bahrani, M., & Ghayoomi, M. (2011).** Lessons from building a Persian written corpus Peykare. *Language Resources and Evaluation,* Vol. 45, No. 2, pp. 143–164. DOI: 10.1007/s10579-010-9132-x.
19. **Jiang, J. & Zhai, C. (2007).** An empirical study of tokenization strategies. *Information Retrieval,* Vol. 10, No. 4, pp. 341–363. DOI: 10.1007/s10791-007-9027-7.
20. **Instituto Colombiano de Cultura Hispánica, Geografía Humana de Colombia, Región Andina Central (2000).** Tomo IV Volumen II. Banco de la República, Bogotá.
21. **Rojas, T.E. (1998).** *Lengua Páez, Una visión de su gramática.* Ministerio de Cultura, Bogotá.

22. **Jung, I. (1984).** *Gramática del Páez o Nasa Yuwe. Descripción de una Lengua Indígena de Colombia.* LINOM GmbH.
23. **CRIC - Programa de Desarrollo Rural en la Región de Tierradentro Cxhab Wala -PT/CW. (2005).** *Diccionario Nasa Yuwe - Castellano*. Popayán.
24. **Rojas, T.C., Perdomo Dizú, A., & Corrales, M.H. (2009).** *Una Mirada al Nasa Yuwe de Novirao,* Sello Editorial Universidad del Cauca, Popayán.
25. **Sierra Martínez, L.M., Cobos Lozada, C.A., Corrales, J.C., & Rojas Curieux, T. (2015).** Building a Nasa Yuwe Test Collection. *Computational Linguistics and Intelligent Text Processing.* LNCS, Springer International Publishing, El Cairo, Egipt, Vol. 9041, pp. 112–123.
26. **Tan, K. (2015)** *Lucene tutorial.com*, http://www.lucenetutorial.com/lucene-in-5-minutes.html.
27. **Lucene Apache (2016).** https://lucene.apache.org/core/3_6_2/api/core/org/apache/lucene/search/Similarity.html.

Luz Marina Sierra M. received her Master's degree in telematics engineering from the University of Cauca (Colombia) in 2016 and the MBA from ICESI University (Colombia) and Tulane University (USA) in 2010. Currently, she is studying her PhD in Telematics Engineering at the University of Cauca (Colombia). She is a full-time professor of Computer Science at the University of Cauca (Colombia). Her research interests include information retrieval, metaheuristic algorithms, Project manager information technology, and Educational Informatics.

Carlos Alberto Cobos received his PhD in Computer Science from the Universidad National de Colombia in 2014. He is a full-time professor of Computer Science at the University of Cauca (Colombia) and director of the Information Technology Research Group (GTI). His research interests include information retrieval, metaheuristic algorithms, and data mining.

Juan Carlos Corrales received the Dipl.-Ing. and Master's degrees in telematics engineering from the University of Cauca, Colombia, in 1999 and 2004, respectively, and the PhD degree in sciences, specialty computer science, from the University of Versailles Saint-Quentin-en-Yvelines, France, in 2008. Presently, he is a full professor and leads the Telematics Engineering Group at the University of Cauca. His research interests focus on service composition and data analysis.

Article received on 16/01/2016; accepted 20/02/2016.
Corresponding author is Luz Marina Sierra M.