Renato Rocha Souza and Isidoro Gil-Leiva

Automatic Indexing of Scientific Texts: A Methodological Comparison

Abstract

We are aiming at establishing a comparison between two information retrieval systems: SISA and PyPLN, regarding their performance when indexing the same set of documents. To this end, we took a corpus of a hundred scientific articles on the field of Agriculture and have them processed by both tools. The index produced by each tool was stored in two different databases. Subsequently, seven queries with information needs were prepared, based on the document contents, in order to establish which set of documents would be relevant for each tool. With the result set, the index and precision indexes were calculated and it was possible to highlight each tool's strengths and weaknesses.

1 Introduction

Research on automating indexing began in the late fifties. Since then, there have been numerous and varied proposals to undertake the intellectual process that involves indexing. The terminology used in the literature to refer to the process of making indexing automatic is varied: we can find names as "Automated assisted indexing", "Automated indexing", "Automated supported indexing", "Automatic support to indexing ", "Computer aided indexing ", "Computer assisted indexing ", among others, whereas the most used is "Automatic indexing ". The definition of automatic indexing must be derived from three perspectives: a) Computer programs that assist in the process of storing indexing terms, once obtained intellectually. (Computer Aided Indexing during storage); b) Systems that analyze documents automatically, but the indexing terms proposed are validated and published - if necessary – by a professional (Semiautomatic Indexing); and c) programs without any further validation programs, i. e., the proposed terms are stored directly as descriptors of that document. (Automatic Indexing).

The methodologies used in automating indexing through the decades have changed until nowadays. In the early days, indexing documents was made almost exclusively from statistics based on terms frequency; but from the eighties on, they incorporated techniques as natural language processing to get the roots of words (stems), morphological taggers and parsers (POS taggers), among others. It is, though, usual that the proposals or prototypes submitted by researchers include a combination of both approaches, i. e, calculating the frequency and tools, more or less complex, for automatic processing of texts (Gil Leiva, 2008).

In spite of all this years of work and research, the use of automatic indexing software is still rare in libraries and documentations centers. Nevertheless, since manual indexing was found impossible for some activities in most of the digital information environments, given the massive amounts of documents to be processed, researchers seek alternatives to represent documents' subjects automatically; using statistical and/or rule based computational linguistic techniques. The oldest and most common process seek to determine documents' subjects solely through the analysis of words' frequencies, but that can lead to poor indexing and erroneous assumptions, as the context can be lost when the collocations are broken into single words. In the last decades, many other techniques were developed, either trying to capture corpus structure with statistical methods, as the TfIDf methodology (Spärck Jones, 1972); Multiword expressions (Silva & Souza, 2014); Latent Semantic Indexing (Deerwester et al., 1988); Latent Dirichlet Allocation (Blei et al., 2003); Word2vec (Mikolov et al., 2010); or aiming at the extraction of the deep semantic structure of the texts (i.e. Extraction of Noun Phrases, Souza & Raghavan, 2006). Also, the use of each technique presents some advantages and drawbacks over the others, as language dependencies (as the case of Noun Phrases), the need of huge computational structures to process the documents timely and the quality of the results. So far, there is no rules of thumb on the techniques and strategies, and it is very common to observe ensembles of these in automatic indexing systems.

In this paper, we are aiming at comparing two indexing systems, each of them using different sets of techniques for indexing documents: the first, named SISA was developed by Gil-Leiva (1999 and 2008); the other, named PyPLN, was developed by the Applied Mathematics School from Fundação Getulio Vargas.

2 The information retrieval systems

In this section, we will present the main characteristics of both SISA and PyPLN.

2.1 SISA

SISA is designed to be used as a semiautomatic system (users can edit the result of the process by adding terms not proposed by the system or browsing the embedded controlled vocabulary of the system to assign additional terms or as a fully automatic system without user intervention once the configuration set.

The system has been developed in JAVA, and different libraries have been used to:

- extract information from documents in PDF, TXT or XML;
- read the controlled vocabulary (SKOS);
- remove the roots of words.

On the other hand, it has been used as a MySQL database to store fonts, documents, results and a retrieval module can be used for system evaluation.

The SISA main features are as follows: It is a system designed for indexing journal articles on web platform implemented with ease of use through a web browser. It works with various file formats such as HTML, PDF, XML and plain text. It also processes documents in Spanish, Portuguese and English, using stopwords and controlled vocabularies in these languages. It makes use of stemming and is based on heuristic and statistical methods with a set of rules that mark the extraction parameters or weighting of terms.

SISA has used a stopword list in Portuguese composed of 586 words and a controlled vocabulary with 9,588 1,122 descriptors and non-descriptors. This vocabulary has only the relationship of synonymy (USE). The vocabulary used by SISA comes from Thesagro, a thesaurus prepared by the National Agricultural Library (BINAGRI) of the Ministry of Agriculture of Brazil. In SISA the following parts of the article are labeled: title, abstract, keywords, authors, headings, first paragraph, conclusions and references with tags such as # ITI # and # FTI #, # CRE # and # FRE #, to delimit the title, starts and ends for many articles parties. If the source texts in txt or PDF formats are not labeled they can be labeled when items are loaded into SISA. Finally, SISA has handled a set of 41 rules that can be grouped into a) positional heuristic rules: If a word is not an empty word, is in a particular combination of tags and appears in the controlled vocabulary, it is presented as descriptor; b) statistical rules: if a word is not a stopword and exceeds a certain frequency or, if a word exceeds a certain TFIDF, it is presented as a descriptor; and c) mixed rules: if a word is not empty word, is in one or more tags or appears above a certain threshold frequency, it is proposed as a descriptor.

Successive tasks for indexing an article with SISA are: label items, process (apply stemming apply, calculate and record TFIDF the place in which they appear words and phrases) and index them according to the configured rules.

SISA is installed on a Proliant server with 32GB RAM ML310E and a CentOS 7.0 operating system. It has been developed in JAVA and different libraries have been used to extract information from documents, read in SKOS format controlled and remove the roots of vocabulary words. On the other hand, it has been used Cascading Style Sheets for application design and MySQL as a database for storing fonts, documents, results and a retrieval module that can be used for system evaluation.

2.2 PyPLN

The PyPLN platform is a research project in active development. Its main goal is to make available a scalable computational platform for a variety of language-based analyses. Its main target audience is the academic community, where it can have a powerful impact by making sophisticated computational analyses doable without the requirement of programming skills on the part of the user. Among the many features already available, we can cite: Simplified access to corpora with interactive visualization tools, text extraction from TXT, RTF, HTML and PDF documents, encoding detection and conversion to utf-8, language detection, tokenization, full-text search across corpora, part-of-speech tagging, word and sentence level statistics, n-gram extraction and word concordance. Many more features are in development and should become available soon, such as: semantic annotation, sentiment and text polarity analysis, automatic social network information monitoring, stylistic analysis and the generation of Knowledge Organization Systems such as ontologies and thesauri. PyPLN aims for unrivaled ease of use, and wide availability, through its web interface and full support to Portuguese language. Besides being a free, uncomplicated research platform for language scholars capable of handling large corpora, PyPLN is also a free software platform for distributed text processing, which can be downloaded and installed by users on their own infrastructure. It was developed using the Python programming language and can be deployed in a single server or in a cluster of servers, for fast parallel processing of documents. It exposes a REST and a Python APIs (Application Program Interface) for ease of embedding its functionalities within other applications.

2 Materials and Methods

To carry out this experiment we have used the two indexing systems (SISA and PyPLN) described in the preceding paragraphs and a corpus of one hundred items in the field of agriculture, published in the Brazilian Journal of Fruticultura, between 2006 (vol. 28, No. 1) and 2007 (vol. 29 , No. 1).

SISA have used a Portuguese stopwords list composed of 586 words and a controlled vocabulary composed by 9,588 1,122 descriptors and non-descriptors. This vocabulary has only the relationship of synonymy (USE). The vocabulary used by SISA comes from Thesagro, thesaurus prepared by the National Agricultural Library (BINAGRI) of the Ministry of Agriculture of Brazil.

The main tasks for the performance of this test were as follows:

1. Build two databases of documents, in each of the tools;
2. Index a hundred documents using both SISA and PyPLN;
3. Choose seven examples of user information needs;
4. For each information need, establish the relevant documents;
5. Convert the information needs into seven search terms and query the database;
6. Apply tests to measure recall and precision of each information need and for each platform;
7. Use these measures the recall and precision to compare SISA and PyPLN.

To measure the rates of recall and precision, we have been used traditional formulas:

- Recall = Number of relevant items retrieved / Number of relevant items in the collection
- Precision = Number of relevant items retrieved / Number of items retrieved.

SISA is composed of three integrated modules which allow the following tasks: processing and indexing of documents; storing metadata items as title, data source magazine, abstract, keywords, descriptors A (descriptors assigned by SISA) and descriptors B (descriptors assigned from another indexation system); and a third information retrieval module. This retrieval module allows searches on the metadata of the stored items.

Once the documents were collected and stored, indexing was automatically triggered by SISA (without human action) for a hundred articles on Agriculture, and we proceeded to manually enter the descriptors also obtained automatically by PyPLN in the field descriptors_B. Thus, we stored SISA and PyPLN indexing results in the database. Indexing in PyPLN was made using a Part of Speech Tagger and an automatic Noun Phrase extractor at first. After extracting the Noun Phrases, the most frequent are considered for assigning descriptors. No sophisticated stopword removal was done in this experiment, because the system does not provide this functionality yet – though it can be easily done in an after processing fashion.

The retrieval module was used to perform information searches in both databases and to apply Recall and Precision formulas with the results obtained.

Fig. 1: SISA Interface

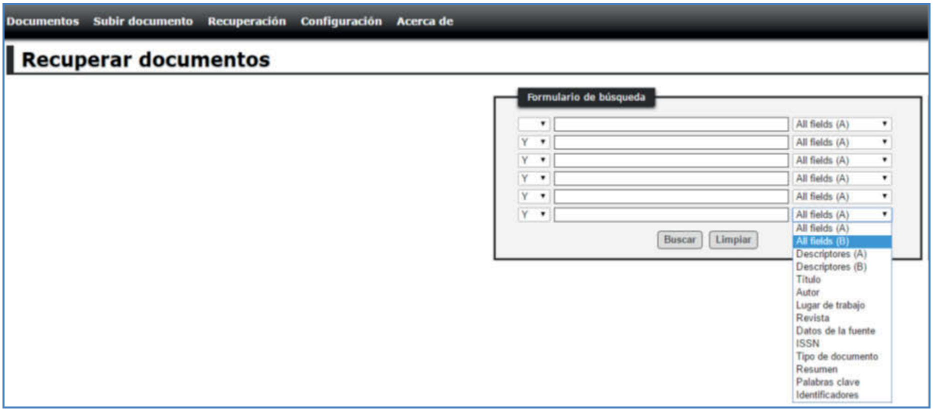
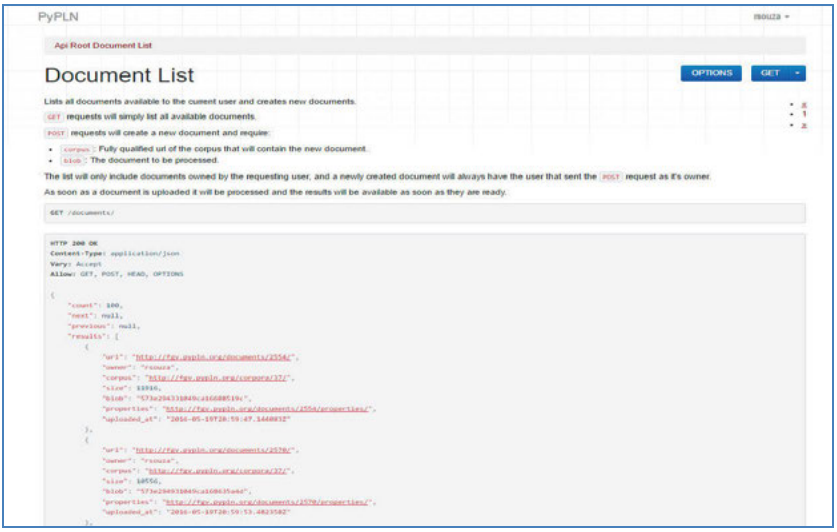


Fig. 2: PyPLN Interface



Document List

Lists all documents available to the current user and creates new documents.

GET requests will simply list all available documents.

POST requests will create a new document and require:

- corpus : Fully qualified url of the corpus that will contain the new document.
- blob : The document to be processed.

The list will only include documents owned by the requesting user, and a newly created document will always have the user that sent the POST request as it's owner.

As soon as a document is uploaded it will be processed and the results will be available as soon as they are ready.

```
GET /documents/
```

```
HTTP 200 OK
Content-Type: application/json
Vary: Accept
Allow: GET, POST, HEAD, OPTIONS

{
    "count": 100,
    "next": null,
    "previous": null,
    "results": [
        {
            "url": "http://fgv.pypln.org/documents/2554/",
            "owner": "rsouza",
            "corpus": "http://fgv.pypln.org/corpora/37/",
            "size": 11916,
            "blob": "573e204331049ca16680519c",
            "properties": "http://fgv.pypln.org/documents/2554/properties/",
            "uploaded_at": "2016-05-19T20:59:47.144083Z"
        },
        {
            "url": "http://fgv.pypln.org/documents/2570/",
            "owner": "rsouza",
            "corpus": "http://fgv.pypln.org/corpora/37/",
            "size": 10556,
            "blob": "573e204931049ca160635a4d",
            "properties": "http://fgv.pypln.org/documents/2570/properties/",
            "uploaded_at": "2016-05-19T20:59:53.482358Z"
        },
```

The queries ran against SISA have used all fields available, such as title, abstract, keywords proposed by the authors of papers and indexing terms obtained by the tool. Queries against PyPLN have used only the terms in the noun phrases automatically attributed by the platform. We also present in appendix 2 the index terms attributed to the documents.

3 Results and discussion

The following tables present the results from the indexing and retrieval process:

TAB. 1: Recall for SISA and PyPLN

SISA Recall			**PyPLN Recall**		
	Searched in all fields	Searched in Descriptors Field		Searched in all fields	Searched in Descriptors Field
Searched 1	0,85	0,71	Searched 1	0,71	0,14
Searched 2	0,75	0	Searched 2	0,75	0
Searched 3	1	1	Searched 3	0	0
Searched 4	1	1	Searched 4	1	0
Searched 5	1	1	Searched 5	1	0
Searched 6	1	0,75	Searched 6	1	0
Searched 7	0,83	0	Searched 7	0,83	0.16
Average	**0,91**	**0,59**		**0,75**	**0,04**

As we can see, recall is lower in PyPLN because it does not make any distinction between descriptors' position in the text, whilst SISA uses this information when indexing. The same occurs when we are comparing the precision measures. The fact that only the most frequent noun phrases were used in the PyPLN indexing process takes a toll in its results, making the results not as good as it would be expected:

TAB. 2: Precision for SISA and PyPLN

SISA Precision			**PyPLN Precision**		
	Searched in all fields	Searched in Descriptors Field		Searched in all fields	Searched in Descriptors Field
Searched 1	1	1	Searched 1	1	1
Searched 2	1	0	Searched 2	1	0
Searched 3	1	1	Searched 3	0	0
Searched 4	1	1	Searched 4	1	0
Searched 5	1	1	Searched 5	1	0
Searched 6	1	1	Searched 6	1	0
Searched 7	1	0	Searched 7	1	1
Average	**1**	**0,75**		**0,85**	**0,28**

Regarding the limitations identified in the operation of SISA and possible improvements, it can be noted that most of the effort and time spent on SISA has been to insert labels to documents. In future experiments, XML format should be prioritized for the scientific papers, since SISA is already implemented to automatically tag documents with certain structures. On the other hand, the controlled vocabulary is an important tool in the operation of SISA, therefore it's necessary to use a large vocabulary of preferred terms and non-preferred terms for enhancing the results. Although the controlled vocabulary used in this experiment has nearly eleven thousand terms it has been observed that there is room to incorporate new terms and to introduce a greater number of synonyms. Finally, it is necessary to continue working on other ways to combine rules SISA.

In the PyPLN side, speed (the whole processing took only three minutes) and the absence of human interaction is key for numbering its advantages. In addition, the use of high frequency Noun Phrases can add a bit of semantics. The lack of stopwords and of any TfIDf weighting procedure, though, has set a penalty in the results. By design, it does not discriminate of the parts of the document in which the extracted words reside. Incorporating these features can truly enhance the performance of the platform.

4 Conclusions

This paper aimed at comparing two automatic indexing platforms; SISA and PYPLN. The results has shown advantages from both of them, with clearly better results presented by SISA, although PyPLN took less time to process the documents. The researchers are planning to incorporate the best features of both tools in new versions of their software, to achieve even better results.

References

Blei, David M., Andrew Y. Ng & Michael I. Jordan (2003). Latent dirichlet allocation. *The Journal of machine Learning research*, 3: 993-1022.

Deerwester, Scott et al (1988). Improving Information Retrieval with Latent Semantic Indexing. In Proceedings of the 51st Annual Meeting of the American Society for Information Science 25. Pp. 36–40.

Gil Leiva, Isidoro (1999). *La automatización de la indización*. Gijón: Trea.

Gil Leiva, Isidoro (2008). *Manual de indización. Teoría y práctica*. Gijón: Trea.

Mesquita, L.A., Souza , R.R., & Porto, R.M.A.B. (2014). Noun phrases in automatic indexing: A structural analysis of the distribution of relevant terms in doctoral theses. *Advances in Knowledge Organization*, 14: 327-34.

Mikolov, Tomas, et al. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301-3781.

Silva, Edson Marchetti & Souza, Renato Rocha (2014). Fundamentos em processamento de linguagem natural: uma proposta para extração de bigramas. *Encontros Bibli*, 19 : 1.

Souza, Renato Rocha & Raghavan, Koti S. (2006). A methodology for noun phrase-based automatic indexing. *Knowledge Organization*, 33 (1): 45-56.

Spärck Jones, Karen(1972).A Statistical Interpretation of Term Specificity and Its Application in Retrieval. *Journal of Documentation*, 28: 11–21.