# Lexicon Construction for Information Systems

**Miriam Sayão[1,2]** **Gustavo R. de Carvalho[2]**

[1]Faculdade de Informática - Pontifícia Universidade Católica do Rio Grande do Sul
Porto Alegre - RS - Brasil
[2]Departamento de Informática - Pontifícia Universidade Católica do Rio de Janeiro
Rio de Janeiro - RJ - Brasil
miriam.sayao@pucrs.br guga@les.inf.puc-rio.br

**Abstract**

Requirements engineering activities may include reading documents which impact in some way the system being developed. Examples of such documents are organization standards, legislation, interview summaries, and also requirements documents. During that process, one of the most important tasks is the construction of a lexicon for the application, because the symbols of the domain are registered on it. The lexicon is a basis to promote understanding among customers, users and software professionals. In this article we propose a strategy for automatic symbols identification to compose an application lexicon, focusing on relevant actors and resources for the elicitation process. The proposed strategy evaluates documents handled in the requirements engineering process. Those documents are written in natural language, and the strategy aims to extract expressions that identify actors and resources to compose a lexicon for the application. Results from two case studies illustrate the application of our strategy.

**Keywords:** requirements engineering, lexicon for information systems, noun phrases.

## 1. Introduction

Requirements-based engineering basically comprises elicitation, modeling and analysis. During elicitation the requirements engineer employs several sources aimed to obtaining essential information for a good understanding of the problem. Traditional information sources include customers and system users, but a careful reading of the documents is also usual in this phase. Company roles, related laws, and other systems documentation are examples of documents that must be evaluated. Documents generated during elicitation are also frequently consulted: interview reports and meeting proceedings, memoranda and messages, and the own requirements documents are part of this documentation. These documents are written in natural language.

During elicitation, terms belonging to the problem domain are used by customers and users, and are also present in the handled documents. Considering that professionals playing different roles and with skills participate on this process, technical terms and those with a different meaning from the usual one may lead to different interpretations. The need for an application lexicon is justified by the all stakeholders sharing the same understanding of terms in the domain.

The lexicon may take different formats, such as a glossary in its simplest form, or ontology, in its most elaborate form. The lexicon is not just a requirement of a quality process, but it also constitutes a reference source for stakeholders. The strategy presented here for an automatic symbols

identification to compose a lexicon is oriented to the Language Extended Lexicon (LEL), proposed by Leite [8]. Symbols included in LEL correspond to actors, objects, verb phrases or states, and will be detailed in section 2. Other approaches to the lexicon, however, can also benefit from this strategy; since symbols used in LEL correspond to entities used in different proposals.

The need for a broad lexicon is greater in distributed development environments. Because stakeholders are geographically distant, have cultural differences and are subject to communication difficulties. The lexicon represents the shared knowledge, reducing difficulties in understanding the terminology of the problem domain. Even in countries that share the same language, cultural differences may also be diminished with the use of a lexicon. This makes it possible for requirements process, which are admittedly intense in communicating, to be less subject to problems derived from imperfections in communication, contributing positively to requirements process activities.

Elicitation activities are an integral part of requirements definition and management. We are developing a strategy for requirements definition and management in distributed software development environments which is based on software agents [15]. This work reports the initial results obtained with basic techniques used to construct software agents, particularly the Lexicon Analyzer Agent and the Lexicon Builder Agent. The former is responsible for functions related to text processing, and the Lexicon Builder Agent is responsible for lexicon maintenance, aiming to update the organization knowledge basis.

This paper is organized as follows: in Section 2 we present the concepts involved in the glossary and lexicon constructions. Section 3 details our process for automatic lexicon construction, according to a process perspective. In Section 4 we present two case studies to demonstrate our strategy. Section 5 compares our work to those in the literature and in Section 6 we present the final conclusions.

## 2. The lexicon and the requirements process

Our challenge is the identification of the symbols that will be included in the application's lexicon, contributing to the construction of the organization domain vocabulary. Software process development, e.g. RUP (Rational Unified Process), define glossaries as an artifact to be generated in the requirements process. A glossary can be understood as a simplified form of a lexicon, structured in a linear form, containing terms and their definitions. The terms to be inserted in a glossary are those used by the stakeholders to make references to application characteristics, aiming to improve comprehension among them. The glossary must be built at the business modeling phase during the construction of the domain model. On the other hand, the Language Extended Lexicon, or LEL, is a more elaborated form of glossary, presenting more information than simply the definition of a term. Next we will briefly introduce the LEL, as proposed by Leite [7] [8].

### 2.1 LEL - Language Extended Lexicon

Terms registered in LEL are typed, and this is the first difference with respect to a glossary. In LEL, terms represent problem domain characteristic symbols, and they correspond to one of four types: subject, object, verb phrase or state. Symbols of LEL have notion and denotation (or behavioral response). The notion of a symbol is what defines it, and the denotation represents the impacts that the symbol causes or receives in the domain. Table 1 [7] shows the four symbols types and their corresponding behavioral responses.

**Table 1 – LEL symbols, notion and behavioral responses [7]**

| Symbol type | Notion | Behavioral response |
|---|---|---|
| Subject | who is the subject | executed actions |
| Verbal phrase | who acts, when it happens and procedure | environmental impacts and resulting states |
| Object | define the object and identifies other related objects | actions that are applied to the object |
| State | what it represents and actions that led to it | identify other states that can be reached from the actual state |

Subjects correspond to active entities, actors with relevant roles in the application. A subject may be an actor, a software component or another system with which interactions will have to occur. Verb phrases describe actions or functionalities to be

performed by the subjects, with some impact on the operational environment. Objects are passive entities used or required by an action or set of actions; and states are characterized by significant attributes that contains values at different moments during system execution. Table 2 presents an example of a lexicon entry for an academic library system.

**Table 2 – LEL entry symbol example**

| **Language Extended Lexicon – Academic Library System** |
|---|
| **User** |
| Symbol type: subject |
| Notion: person who uses the library; may be a student, a teacher or a university employee |
| Behavioral Responses: |
| user is registered in the system |
| user is removed from the system |
| user checks books from the library |
| user returns previously checked books |
| user renews deadlines for book return |

## 3. Extracting symbols for lexicon construction

In this article we present the proposed process for the extraction of subjects and objects from organization documents, using an approach based on noun phrases. Noun phrases are defined as (a) a grammatical class with the syntactic role of a subject, a direct object and, if preceded by a preposition, an adnominal adjunct or indirect object [13] [16]; (b) a concept or an entity (abstract or concrete) identified by proper names or noun phrases; they can represent roles as well [9].

In software engineering, Booch et al. affirms that an actor represents a role that may be played by a human, a hardware device or some other software system [1]. In technical documents, actors are registered through identifiers that include proper names, professions and roles. In linguistic terms, we can associate actors to noun phrases. Resources correspond to objects that occupy a space in the real or virtual world and are used or generated by an action. Resources and objects can be identified by nouns and consequently by noun phrases as well. Figure 1 illustrates the process of symbol extraction, which is based on noun phrase extraction.

### 3.1 Detailing the symbol extraction process

The strategy proposed is schematized using a process perspective, as shown in Figure 1. The process can be divided into four activities or sub-processes. Initially, the document is prepared to be handled by a Part of Speech tagger (POS tagger). After the insertion of tags, noun phrases are extracted, which are related to pre-defined patterns. These noun phrases will go through a selection process, which will consider only those that attend to relevance criteria. The final stage extracts, for each of the noun phrases, concordances that can be used as notion and behavioral responses. Each one of these sub-processes will be detailed.

#### 3.1.1 Preparing for processing

It is mandatory to prepare the documents, because the tools which we have used work only with *txt* type files. The documents can be in different formats, such as *pdf* (Portable Document Format), *doc* (Microsoft Word document) or *rtf* (Rich Text Format). Documents may even include figures and tables, which will not be processed by text processing tools. This process initially removes figures, transforms tables into text, removes possible formatting tags and generates *txt* files.

After converting files to the *txt* type, the text is tokenized, that is, split word by word, including punctuation marks, placing one token per line.

#### 3.1.2 Extracting noun phrases

In Portuguese, noun phrases possess a well defined structure. Perini [13] affirms that noun phrases possess two basic structures: the structure to the left of the head is comprised of positions that can be occupied by determiners, possessives, quantifiers and other word classes. The structure to the right is comprised of modifiers, which in turn can be open categories or even other noun phrases. In this work, we do not use the structure to the left of the head, since it is not usual for technical documents or those using technical language to work with more sophisticated phrasal constructions, as usually happens in literature. The patterns that we establish for the selection of actors/subjects and objects/resources will be presented in sections 3.2 and 3.3.

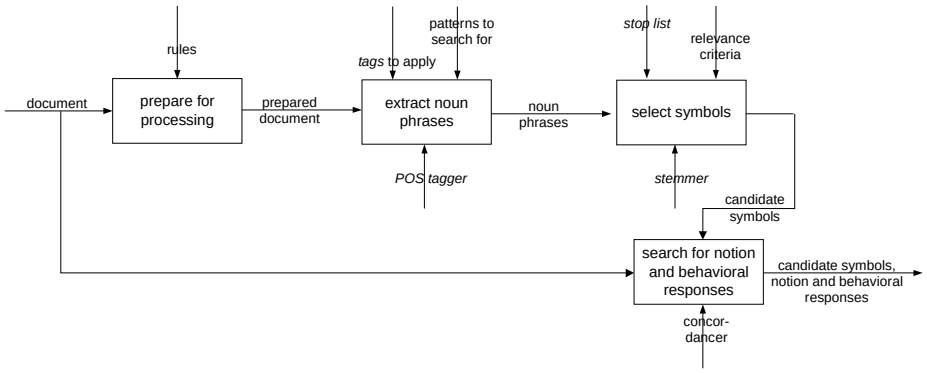


**Figure 1 – Symbol extraction process**

The process for extracting noun phrases is based on a morphosyntactic POS tagger, which analyzes the pre-processed text and associates a tag to each token. Afterwards, the noun phrases are extracted and the frequencies of each are obtained, according to pre-established patterns. There are some POS *taggers* for Portuguese, and for our work we choose QTAG, detailed in [10]. Although other *taggers* possess a slightly higher precision[1], we prioritize QTAG because the tagged texts are in a format similar to XML, which facilitates the exchange of information between different applications.

The QTAG is a probabilistic *tagger*, originally developed for the English language. The precision score of tags is around 96.3% [10]. The new version of this *tagger*, in the Java programming language, can be used to process texts in various other languages, through table construction generated by training in specific corpora. For the Portuguese language, this *tagger* was trained with a corpus of approximately 500,000 words, by researchers T. Sardinha and R. Lima-Lopes, associated to Lael/PUCSP. According to experimental data, its precision score is 93% [14]. The set of tags for Portuguese texts can be obtained at http://www2.lael.pucsp.br/corpora/etiquetagem.

### 3.1.3 Selecting symbols

The process of symbol selection from noun phrase candidates disregards those which are on a *stoplist*. This *stoplist* is composed of terms from the problem domain or from the general language that are not relevant to the lexicon. The following stage consists of *stemming* the noun phrases extracted. This is necessary since, in our experiments, we need to group together terms in the plural, singular, feminine and masculine forms. For example, in our case studies, noun phrases like *gestor de escal**a*** e *gestor de escal**as*** (singular and plural forms) had been extracted and counted separately, the same occurring with *cessionári**a*** e *cessionári**o*** (feminine and masculine forms). We use the stemmer originally developed by V. Orengo [11] and modified by M. A. L. Dias [4]. Noun phrase candidates are grouped and counted after the stemming process, and we adopt the singular masculine for representation, when needed.

The last activity of the selection process discards noun phrases whose frequency is less than four, a common procedure in terminology extraction [3]. In our experiments, with documents of different sizes, this procedure was adequate for our purposes. However, we are planning to carry out new experiments considering different values in order to confirm or modify our choice.

### 3.1.4 Searching for notion and behavioral responses

The concordance search works with each one of the chosen symbols, scanning the document in order to identify the presence of the symbol. When this happens, the paragraph is extracted, since it may correspond to the definition or the behavioral responses of the symbol. In sentence identification, the *stem* of the symbol being searched is used; the extracted paragraphs will be grouped together with the candidate symbol, and utilized afterwards by the requirements engineer for insertion into the lexicon of the application. In the case studies, we observed that the identification of notion (or definition) of the symbols is sensitive to the context of the application. This stage of our process is still in a

[1] *Projeto Lácio-Web, http://www.nilc.icmc.usp.br/lacioweb/ferramentas.htm. Accessed in 28.07.06*

phase of refinement and will not be detailed in this article.

### 3.2 Subject or Actor Extraction: patterns used

In the Portuguese language, functions or roles played by people or entities are identified by nouns with specific suffixes. Some examples with the ending *-ente*: gerente ('manager'), presidente ('president'); ending *-or*: gestor ('manager'), diretor ('director'), trabalhador ('worker'); ending *-ário*: usuário ('user'), funcionário ('employee'). In the extraction of noun phrases corresponding to actors/subjects, we use a set of 82 patterns, since for each ending we consider the singular, the plural, feminine and masculine genders, in practice multiplying each ending by four. Some examples of patterns utilized are:

| | |
|---|---|
| N*ente PRP* N* | N*entes PRP* N* |
| N*enta PRP* N* | N*entas PRP* N* |
| N*or PRP* N* | N*ores PRP* N* |
| N*ora PRP* N* | N*oras PRP* N* |
| N*eiro PRP* N* | N*eiros PRP* N* |

In these patterns, * indicates any number of characters. For example, the pattern **N*ente PRP N*** should extract phrases composed of a noun, preposition and noun, being that the first noun should have the ending **ente**. The patterns were defined empirically, after the evaluation of tagged texts.

### 3.3 Object or Resource Extraction: patterns utilized

The extraction of resources/objects used more generic patterns than those used in the stage of actor/subject extraction. Nine patterns were used, also defined empirically after the evaluation of tagged documents.

| | |
|---|---|
| N* PRP* N* | N* N* |
| N* PRN* N* | N* CPR* N* N* |
| N* PRP* N* PRP* N* | |

## 4. Case studies

Our proposal was validated by two case studies, detailed here. To support the proposed process, we used a prototype which consists of programs written in Java, and a library for the treatment of regular expressions and patterns.

### 4.1 Case study: application in the financial area

This first case study used public documents related to the system SELIC, from Banco Central do Brasil ('Brasilian Central Bank'), available at http://www.bcb.gov.br/htms/spb/. The documents follow a standard structure for the SELIC system. The total number of words from the portion of documents evaluated is 5,461. Table 3 presents partial results obtained after the application of the proposed process.

### 4.2 Case study 2: application in the area of human resources

This case study used a requirements document for an information system in the area of human resources. The document follows standards internal to the organization, modeling requirements with use cases and supplementary specifications. The total number of use cases in this document is 60 and there are 5 supplementary specifications, related to documentation, training and usability. The total number of words in this second document is 6,665. Table 4 presents part of the set of subjects and objects selected.

### 4.3 Preliminary results evaluation

The percentages of *recall* and *precision* of the preliminary results are summarized in table 5. The evaluations were realized by independent teams, one of them composed of specialists from Banco Central (case study 1) and the other of software engineers (case study 2). Although a small number of case studies were carried out, the percentages of *recall* and *precision* show the appropriateness of the proposal for the semi-automatic construction of a lexicon of applications. We are initiating new experiments with the application of this strategy to documents from different application domains, aiming to obtain a significant set of results for analysis. On the other hand, some aspects of the proposal are under revision, seeking to improve the results already obtained.

The extraction algorithm for noun phrases is under revision in order to avoid that those phrases which attend to more than one pattern are retrieved twofold. Some modifications in the patterns used are being experimented, aiming to obtain better percentages of *recall* and *precision* with respect to the retrieved objects.

**Table 3 – Symbols of the type subject and object in the Selic system**

| Subject | Pattern | Freq. |
|---|---|---|
| cedente | N*ente | 17 |
| clientes | N*entes | 17 |
| cessionário | N*ário | 15 |
| correspondente | N*ente | 10 |
| instituição financeira | N* ADJ* | 9 |
| usuário | N*ário | 9 |
| cedente da operação | N*ente CPR* N* | 6 |
| cessionário da operação | N*ário CPR* N* | 6 |
| destinatário | N*ário | 6 |
| emissor | N*or | 6 |
| instituição liquidante | N*ão N* | 6 |

| Object | Pattern | Freq. |
|---|---|---|
| operação compromissada | N* PART* | 35 |
| retorno de operação | N* PRP* N* | 17 |
| conta reservas | N* N* | 16 |
| pu de retorno | N* PRP* N* | 16 |
| número de operação | N* PRP* N* | 10 |
| reservas bancárias | N* N* | 10 |
| situação da operação | N* CPR* N* | 9 |
| conta da instituição | N* CPR* N* | 7 |
| operação 1054/s | N* N* | 7 |
| pu da operação | N* CPR* N* | 7 |
| endereço eletrônico | N* ADJ* | 6 |

**Table 4 – Symbols of the type subject and object in the work schedule management system**

| Subject | Pattern | Freq. |
|---|---|---|
| gestor | N*or, N*ores | 101 |
| usuário | N*ário | 88 |
| gestor de escalas | N*or PRP* N* | 68 |
| usuário final | N*ário N* | 58 |
| funcionário | N*ário, N*ários | 39 |
| administrador | N*or | 28 |
| colaborador | N*or, N*ores | 22 |
| administrador central | N*or N* | 12 |
| leitor | N*or | 8 |
| empresa | N*esa | 7 |
| administrador local | N*or N* | 7 |

| Object | Pattern | Freq. |
|---|---|---|
| banco de dados | N* PRP* N* | 11 |
| cálculo de estimativas | N* PRP* N* | 10 |
| sistema operacional | N* N* | 8 |
| postos de serviços | N* PRP* N* | 7 |
| parâmetros de pesquisa | N* PRP* N* | 7 |
| solicitações de processamento | N* PRP* N* | 6 |
| lista de postos de serviços | N* PRP* N* PRP* N* | 6 |
| sistema administrador | N* N* | 5 |
| relatório de ocorrencias | N* PRP* N* | 5 |
| log das ocorrencias | N* CPR* N* | 5 |
| habilita relatório | N* N* | 5 |

**Table 5 – Values of recall and precision obtained in the case studies**

| | ec1: subj | ec1: obj | ec2: subj | ec2: obj |
|---|---|---|---|---|
| **recall** | 92% | 100% | 78% | 78% |
| **precision** | 75% | 78% | 84% | 56% |

We also observed some problems resulting from the insertion of incorrect tags, mainly with respect to objects, since they require more general patterns than those used with subjects. This is the case with the object *habilita relatório* ('enable report'), presented in table 4, where the word *habilita* was incorrectly tagged as a Noun, when the correct tag is Verb. A more detailed analysis of the problems is scheduled to be performed after the adjustments already cited and after a larger set of experiments is performed.

## 5. Related studies

In the context of natural language processing and information retrieval, the use of noun phrases for information retrieval is found in many studies. Parreiras [12] proposes this use in scientific texts indexation and Pérez [6] uses them to obtain concepts that will compose conceptual maps. In the context of the requirements process, studies targeting the exploitation of natural language aspects are still not very common, and we selected the studies described in [5] and [2], in search for different frameworks dealing with noun phrases in requirements documents.

The CM-Builder, described in [5], is a *case* tool which evaluates requirements documents written in natural language (the English language) searching for semantic relations in requirements sentences and aiming to build an initial model of classes in UML; the classes are derived from noun phrases. The result includes classes, attributes and relationships, grouped in class models and stored in files in a format adequate for subsequent handling by tools normally used in development project phase.

The authors emphasize that the results should be seen as a support for the software or requirements engineer, needing also some refinement to effectively contribute to an initial model of classes.

The framework proposed by Boyd et al. [2] is in the

line of research which uses a controlled subset of natural language to register the requirements, with the objective of reducing ambiguity. That work investigates the syntactic and semantic expressiveness of a subset of the English language for use in requirements documents, focusing specifically on verbs. To correct errors in the entities identification (entities retrieved through noun phrases) by the POS tagger, the authors used a dictionary generated from noun phrases in an existing glossary.

Our work is similar to the one described in [5], but our objectives are different: we aim to support the requirements process, while the aforementioned study aimed to support the system project phase. Other than requirements documents, we also use information sources that can be handled or generated in the requirements process – it is sufficient that these documents are written in natural language. While the framework described in [2] uses a restricted subset of natural language, our framework does not restrict the use of the Portuguese language in the documents handled.

Our goal is to identify relevant actors/subjects and resources/objects, in such a way to support the construction or updating of the application lexicon. The process created is also useful in the construction of a lexicon for the organization domain, since in organizations using distributed development of software, difficulties derived from different linguistic capacities, different cultures and *delays* in communication highlight the increased need for a broad lexicon.

Our framework also uses a *stoplist* consisting of terms not relevant to the considered domain, allowing adjustments in the terms to be extracted and generating more precise results in the future evaluation of documents from the same domain. As emphasized by Harmain and Gaizauskas [5], we believe that the obtained results do not preclude the revision and evaluation by specialists of the domain. The subjects and objects extracted through the proposed process constitute an important aid for the construction or update of the application lexicon.

## 6. Conclusions and future work

In this article we presented a process for the automatic identification of symbols to build a lexicon for the application domain. The extraction is based on noun phrases identification, through the use of empirically identified patterns in handled documents in the requirements process. The required steps of the process begin with *tokenization* of the document, and then there is tagging to identify the word classes. Noun phrases that attend to predefined patterns are extracted and consolidated. Symbols that match simple frequency criteria are selected to compose the application lexicon, and for each of these symbols, the contexts surrounding them are extracted as well, so that the requirements engineer may obtain the definition and behavioral responses of the symbol.

We carried out two case studies, one of them in a requirements document made available by a company that uses distributed software development; the participants of the requirements process are geographically separated, which increases the importance of the lexicon in order to share the same understanding with respect to the symbols characteristic of the application domain. The other case study used documents extracted from a user's manual, showing that the strategy is applicable to different types of documents handled in the requirements process. The strategy can be used for other purposes, for example, to search for relevant actors and to define candidate classes for the project process.

The following stages of this work involve the realization of new experiments, aiming to evaluate the application of the proposal in different types of documents and different domains. After this, the process of obtaining notion and behavioral responses from extracted symbols will be refined. The final stage is related to the structuring of the whole process in the form of services of the agents Lexicon Builder and Lexicon Analyzer, referred to in the introduction of this article.

This work has also evidenced that the maturity already reached by methods and techniques in the area of Natural Language Processing enables its application in the process of software development. The current availability of a greater number of tools that work with the Portuguese language contributes to this, which was possible to verify in this work.

## Acknowledgements

We thank Tlantic Sistemas de Informação, for providing the requirements documents and to the anonymous reviewers, whose suggestions and questions contributed to the improvement of this

work. We also thank Will Lowe, for the Yoshikoder converter.

## References

[1] G. Booch ; J. Rumbaugh & I. Jacobson. "UML: guia do usuário". Rio de Janeiro: Campus, 2000. 472 p. ISBN 8535205624

[2] S. Boyd; D. Zowghi & A. Farroukh. "Measuring the expressiveness of a Constrained Natural Language: an Empirical Study". In: 13th IEEE International Conference on Requirements Engineering (RE´05). Proceedings.

[3] B. Daille. "Study and Implementation of Combined Techniques for Automatic Extraction of Terminology". In: Klavans, J., Resnik, P. The Balancing ACT- Combining Symbolic and Statistical Approaches to Language, The MIT Press, 1996. pp. 49-66.

[4] M. A. L. Dias. "Extração Automática de Palavras-Chave na Língua Portuguesa Aplicada a Dissertações e Teses da Área das Engenharias". Master Thesis. Campinas: FEEC-UNICAMP, 2004.

[5] H. M. Harmain& R. Gaizauskas. "CM-Builder: An Automated NL-based CASE Tool". In: 15th IEEE International Conference on Automated Software Engineering (ASE'2000), 2000. Proceedings. pp. 45-53.

[6] C. C. C. Pérez; C. Gasperin, & R. Vieira. "Extração semi-automática de conhecimento a partir de textos". In: IV Encontro Nacional de Inteligência Artificial (ENIA 2003), Campinas, 2003. Anais da SBC, 2003, v. 7, pp.193-202.

[7] J. C. S. P. Leite; A. P. Franco. "O uso de hipertexto na elicitação de linguagens da aplicação". In: 4º Simpósio Brasileiro de Engenharia de Software, 1990. Proceedings. pp.124-133.

[8] J.C.S.P. Leite & A. P. M. Franco. "A Strategy for Conceptual Model Acquisition". In: First IEEE International Symposium on Requirements Engineering, San Diego, Ca, IEEE Computer Society Press, 1993. Proceedings. pp. 243-246

[9] Y. G. Liberato. "A estrutura do SN em português: uma abordagem cognitiva". Doctoral Thesis, 1997. UFMG, Departamento de Lingüística, Belo Horizonte.

[10] O. Mason & D. Tufis. "Probabilistic Tagging in a Multi-lingual Environment: Making an English Tagger Understand Romanian". In: Third European TELRI Seminar, Montecatini, Italy, 1997. Proceedings.

[11] V. M. Orengo & C. Huyck. "A Stemming Algorithm for Portuguese Language". In: 8th Symposium on String Processing and Information Retrieval (SPIRE 2001), Laguna de San Raphael , Chile, (2001). Proceedings. pp. 186-193.

[12] F. Parreiras. "O uso de sintagmas nominais como fonte de descritores para textos de periódicos científicos". Escola de Ciência da Informação. Belo Horizonte, 2003. http://www.fernando.parreiras.nom.br/publicacoes/sn.pdf.

[13] M. A. Perini. "Gramática descritiva do português". 3ªed. - São Paulo: Ática, 1998. 380p. ISBN 8508055501

[14] A. P. B. Sardinha. "Lingüística de Corpus". São Paulo: Ed. Manole, 2004. 410 p.

[15] M. Sayão & J. C. S. P. Leite. "Uso de Agentes no Processo de Requisitos em Ambientes Distribuídos de Desenvolvimento". In: Workshop de Engenharia de Requisitos, Lisboa, Portugal, 2005. Proceedings.

[16] R. Vieira & V. L. S. Lima. (2001). "Lingüística Computacional: Princípios e Aplicações". In: As Tecnologias da Informação e a questão social: anais. Carlos Eduardo Ferreira (Ed.) Fortaleza, SBC. ISBN 85-88442-03-5 (v.2). pp 47-88.