Semantic Indexing of Technical Documentation

Samaneh CHAGHERI[1], Cyril DUMOULIN[2]

1. Université de LYON, CNRS, LIRIS UMR 5205-INSA de Lyon
7, avenue Jean Capelle
69621 Villeurbanne Cedex France
E-mail: samaneh.chagheri@insa-lyon.fr

2. CONTINEW
27, rue Lucien Langénieux
42300 Roanne France
E-mail : rd@continew.fr

Abstract. This research takes place in an industrial context: the CONTINEW Company. This company ensures the storage and security of critical data and technical documentation. Consequently, it is necessary to organize these documents in order to retrieve quickly critical information. The management of this increasing volume of documents requires document classification which is based on indexing techniques. So, how much relevant the indexing phase is, more relevant the classification will be. The technical documentation is by nature strongly structured. For example, the logical structure describes the role and the nature of the document elements (introduction, title, section, and so one…) and the logical links between them (A chapter is composed of section and so one). Such structure facilitates document presentation and improves the indexing precision. The classical information retrieval systems use neither the logical structure, nor the concept contained in the textual content of documents. The document semantic is described by concepts belonging to a semantic resource. In this context, we propose a new semantic indexing model which exploits both the logical structures and the semantic contents of documents.

Keywords: Technical documentation, logical structure, semantic structure, semantic indexing, semantic resource.

1 Introduction

Technical documentation is the generic term for documentation with regard to a product's data and information stored for different purposes. Different purposes are user manual, product specification, manufacturing, product presentation; etc.These documents have an important role in industrial production. Indeed without such documents, the products can neither be manufactured nor used according to their complexity. Technical documents are strongly structured. Structure here means logical structure like title, chapters, sections, figures, paragraphs, etc.

The logical structure describes the role of each element of a document. Each element type corresponds to a logical unit in the document, like title or chapter. All these logical elements are organized as a tree to represent the inclusion link between elements. These documents are often available in electronic formats as "PDF" or "Word". So, by using extensibility of XML, it is possible to represent simultaneously the content and the logical structure of documents.

The continuous growth in structured documents stored in companies has caused different efforts in developing retrieval systems based on document structure. These systems exploit the available structural information in documents, as marked up in XML, in order to implement a more accurate retrieval strategy and return document structural elements instead of complete documents. However, much of the information is contained in the text fields not just in tag labels.

Moreover, most of the terms have more than one meaning, and a meaning can be expressed by different terms. Therefore, the notion of concept in the documents has been increasingly important and has attracted many researchers to identify them. Semantic resources and ontologies have an important role in conceptual indexing.

Traditional IR systems consider neither the structure nor the semantic of document content. They mostly focus on indexing the document content as a bag of words, without considering the concept and the structural element in which words appear. Therefore, devising a method which exploits both the structure and semantic of documents is promising.

This method has to use an external semantic resource in order to extract the concepts. In the other hand, it should consider the logical element in which the identified concept appears.

In this article, we have proposed a method which is an extension of the vector model of Salton (Salton G., 1968) adjusting the calculation of the TF-IDF by considering the structural element instead of whole document. We suggest using a semantic resource to model the semantic of document content. This indexing allows a search based on context (document structure) and semantic (the concepts of external semantic resource).

The rest of the article is organized as follows. We present in section 2 the different approaches proposed in the literature to conceptual indexing and indexing the structural documents. Our proposal on semantic indexing the structured documents is written in section 3. Section 4 presents the conclusion and further works.

2 Related Works

The appearance of XML documents has caused a lot of researches on adapting information retrieval techniques to structured documents. Taking into account the logical structure of documents affects the document representation.

Wilkinson was the first to propose an information retrieval system based on document structure (Wilkinson R., 1994). In his system, Documents are split in section and the query is compared to each section. Document relevancy depends on different aspects: the frequency of terms in document content, frequency of term in a section content and section type. He applies the *tf*idf* formula to section of document instead of the whole document.

S.Myaeng (Myaeng S., 1998) proposes an information retrieval model derived from the inference net model. He has improved the effectiveness of document retrieval in his model compared to traditional systems based on whole document without structure. In this net, the document is represented by the hierarchy of nodes where the leaves contain the textual part. His approach is based on the Bayesian network for computing the probability of term occurrence in logical elements of document.

Yossi Mass in INEX03 (Mass Y., 2003) describes a method for component ranking in XML documents by creating separate indices for the most informative logical element type in the collection of documents. In (Mass Y., 2004) they have improved their approach by proposing document pivot to compensate the problem of the data outside the scope of the logical element. The document pivot scales scores of logical elements by the scores of their containing articles. Their method is based on the vector space model and *tf*idf* formula.

In M.Lalmas (Lalmas M., 2000) document is represented by a tree with nodes, edges and leaves which represent respectively logical elements, compositional relationship and raw data in textual part of document. His approach is based on evidential reasoning which can be applied to model the representation of content only, structure only and both content and structure of document. In (Kazai G., 2002) they use best entry points (BEPs) which correspond to document logical elements from where users can browse to access relevant document elements. This approach represents a document element as an aggregation of the contents of all its logical sub elements.

K.Sauvagnat (Pinel-Sauvagnat K., 2006) has combined the different factors for calculating the weight of terms in structured document. She has used *tf* for taking into account the local importance of the term, *idf* and *ief* for global importance of term in collection and document. And ief^d estimates the semi-global importance of term in the collection of structural elements in the document.

In (Schileder T., 2002), T.Schlieder adopts the similarity measure of the vector space model, incorporates the document structure, and supports structured queries. He extends the term to structural term which includes structure of query and document. The notion of term frequency and inversed document frequency are adapted to logical element of documents.

Besides the structural indexing, recently many researchers have focused on conceptual IR systems by using semantic resources (Salton G., 1968). E.Voorhee (Voorhees E., 1993) uses WordNet to disambiguate terms by considering the hyponymy relation between synsets of WordNet. His experimentations are applied to nouns synsets only. .

N.Guarino in (Guarino N., 1999) presents OntoSeek a research system specifically targeted to on-line yellow pages and product catalogs by using a semantic resource like WordNet for supporting content matching. In this system a basic formalism of conceptual graph is used.

Khan in (Khan L., 2004) proposes a concept-based model using domain-dependent ontologies. He uses an automatic disambiguation algorithm which prunes irrelevant concepts. Only relevant concepts are associated to documents and thus they participate in query generation.

In (Zargayouna H., 2004) the computation of term weights is influenced by the context (the indexing unit) in which they appear. The computation of weight based on the tf-*idf* method is applied to tags. Thus, the author proposes the *tf- itdf* formula

(Term Frequency - Inverse Tag and Document Frequency), which estimates the discriminatory power of a term t for a tag b in a document d. This work uses the concept and document structure together.

3 General Overview of our Model

In this article we have proposed indexing the structured documents by considering the logical structure and the semantic of the document simultaneously.

By using the XML format of document, we present the document as a tree in which the nodes are logical elements like chapter, title, paragraph, etc… and the leaves are the textual content of each structure element.

On textual part of each logical element we do some pre-processing analyses to extract the words and their lemma. Then we extract candidate terms which can be single or multi words. Each term corresponds to an entry in the lexical database WordNet. Then we compute the term's weight by expending TF*IDF at each document structural level instead of document level. Thus we have a vector of weighted terms for each logical element type.

By using an external semantic resource we identify terms related concepts. Due to polysemic term that can appear inside a document, we add a disambiguisation phase in order to select the best concept for each term in our text. For this reason we calculate the semantic distance between concepts. At the end of this process we have a vector of weighted concepts for each logical element type.

At the end for each document we build an index for each logical element type for example index of title, index of paragraph… Then we merge these vectors to one unique vector to use it for the purpose of classifying the documents and comparing their index.

3.1 Identifying the Candidate Terms

The first phase consists of a series of basic linguistic analysis on the text to prepare it for the concept extraction phase. We use some Natural Language Processing tools like tokenisers, lemmatisers and others parsers to obtain word lemma instead of derived forms.

After pre-processing the text, we have to identify the candidate terms in document which are representative of its content. We project the document onto the lexical database WordNet (Miller G., 1995) which is able to identify more than 90% of terms in the text (Baziz M., 2005), by mapping WordNet synsets and document, we extract the terms in documents which correspond at least to an entry in WordNet.

For extracting the compound terms, we define a window size containing the successive words. In this window we search the longest combination of words which matches with an entry in the semantic resource as Baziz has done in (Baziz M., 2005).

After identifying the candidate terms we compute their weights according to their frequency in logical elements in which they appear.

3.2 Logical Structure Modeling

XML documents indicate the logical structuring of document: like title, chapter, paragraph and etc. These structural elements are used in order to retrieve document’s terms and compute their weight according to their positions inside the document.

Take the following example. We have a document in XML format:

```
<article lang="en">
<title> Technical Introduction</title>
<sect1 id="section 1">
        <title> Introduction</title>
        <para> Dust and foreign material...</para>
</sect1>
<sect1 id="section 2”>
        <title> Cleaners</title>
        <sect2>
                <title> Card Cleaners</title>
                <para> The cleaning surfaces are individually ...</para>
        </sect2>
        <sect2>
                <title> Stick Cleaner</title>
                <para> This is a tool to clean...</para>
               <para>  It is an appropriate...</para>
         </sect2>
</sect1>
</article>
```

Example 1: XML format of a document

We consider the structured document as a tree in which, the nodes are logical elements of document like chapter, section or etc. The arcs of this tree represent the inclusion relation between logical elements, for example in a document we have a section and some subsections under this section. The leaf nodes are the textual part of logical elements; indeed they contain the document content. Each element has a label which defines its type, for example section or paragraph. The tree schema of the example document has been shown in Figure 1.

To simplify the computation we aggregate the nodes with the same label as shown in Figure 2. For example if there are 2 section titles in the document we consider a single element called section title which has two leaf nodes containing text. In the rest of this article, a term in a logical element means its occurrence in all the leaves of the same logical element node.

Indeed, we do language processing and term identification in each logical element instead of whole document. We have to consider the path of node in our segment separation. For example in Figure 1 we have the label “title” in two different levels, one is the title of document, and the other is the title of section which are different.

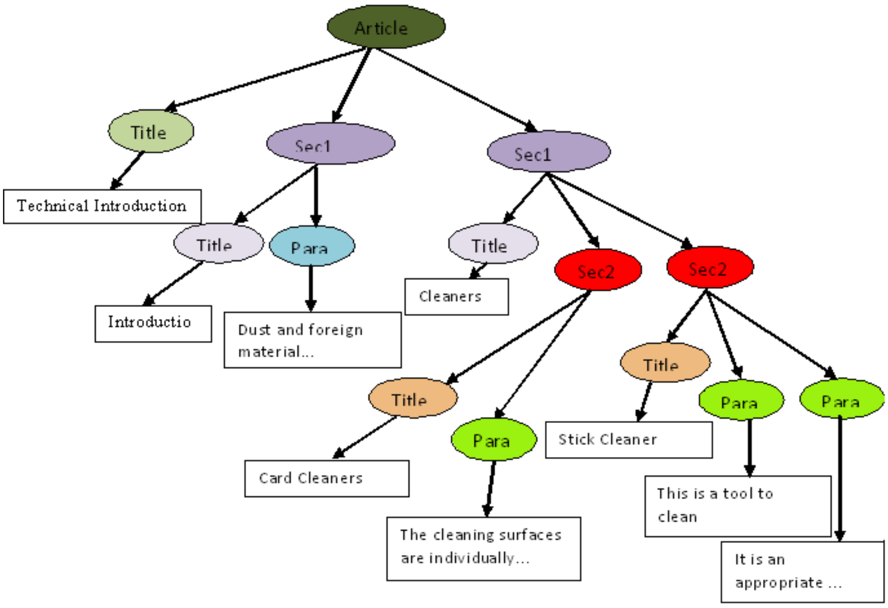

Figure 1: The tree schema of document

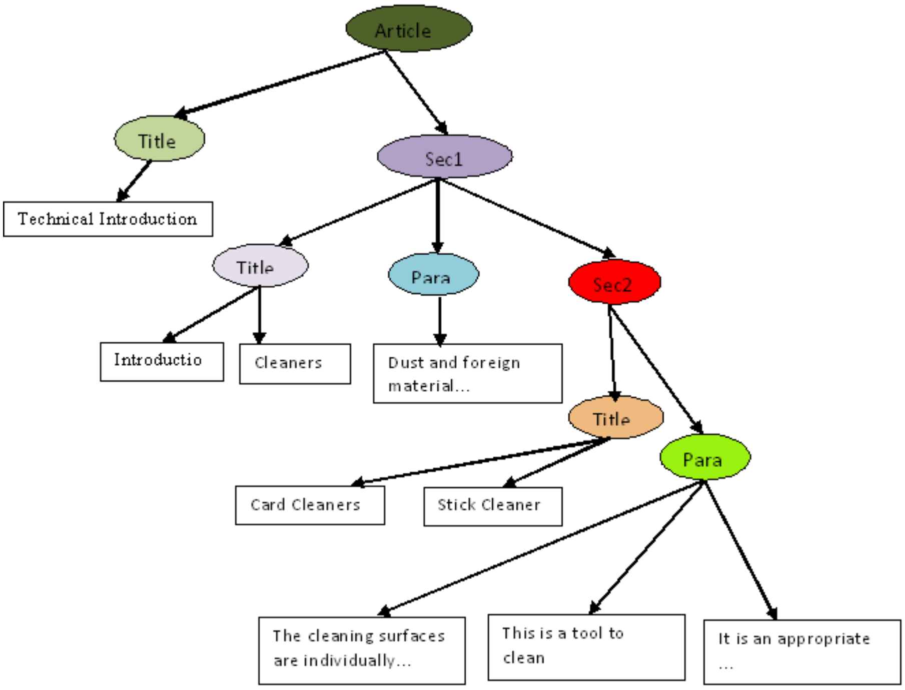

Figure 2: Final aggregated tree of document

After separating the logical elements and doing some basic processes on textual content as mentioned previously, we compute term's weight according to its frequency in its logical element.

The document index is composed of several vectors of weighted terms: one by element type.

$$d_d = \bigcup_{e=1..x} (w_{1,e,d}t_1, \ldots w_{i,e,d}t_i, \ldots, w_{n,e,d}t_n)$$

Where:
d_d is the document d
x is the number of logical element types found in the corpus of document.
n is the number of terms found in the corpus of documents.
We have extended the tf.idf as bellow:

$$\mathrm{w}_{\mathrm{i,e,d}} = \mathrm{tf}_{\mathrm{i,e,d}} * \mathrm{ief}_{\mathrm{d,e}}$$

Where:

$W_{i,e,d}$ is the weight of the term t_i for the logical element type e inside the document $d_{d.}$

$$tf_{i,e,d} = \frac{\text{frequency of ti in the logical element e of document d}}{\text{nb of leaf nodes of the logical element e in the document d}}$$

$$ief_{d,e} = \log \frac{D_e}{d_{e,i}}$$

D_e = total number of documents in the corpus which have the same logical element e
$d_{e,i}$ = number of documents which have the same element e containing the term i

The size of the retrieved logical elements should be considered for the term frequency. Thus we devise the term frequency by the number of leaves of the logical element e to prevent a bias towards longer documents (which may have a higher term frequency regardless of the actual importance of that term in the document). For example, consider that the document d_j has 6 logical elements “section.title”, and the overall frequency of term t_i “computer science” in this logical element is 6. That means that t_i could appear in each title of section in the document d_j. Its $tf_{i,e,j}$ = 6/6=1.

Consider that the document d_k has 1 logical element “section.title”, and the overall frequency of “computer science” in this logical element is 1. Thus Its $tf_{i,e,k} = tf_{i,e,j}$ =1

Indeed, the $ief_{d,e}$ globalizes the appearance of the logical element e in the corpus. A term which has appeared in a same logical element of all documents cannot be significant and its weight in that element will be 0.

Take the example in Tableau 1: content of logical elements of documents; we compute the weight of term “introduction” in all logical elements of Doc1. This term has appeared in document title and section title. We consider that there are 2 documents in our corpus:

$$w(\text{"introduction"}, \text{document.title}, \text{Doc 1}) = 1 * \log\left(\frac{2}{1}\right) = 0.3$$

$$w(\text{"introduction"}, \text{section.title}, \text{Doc 1}) = 1 * \log\left(\frac{2}{2}\right) = 0$$

“Introduction” in section.title cannot be at all informative, thus *ief* will be zero. But in the document.title is a significant term and we do not miss this term.

Tableau 1: content of logical elements of documents

Logical element	Doc 1	Doc 2
Document.Title	Technical Introduction	User Guide
Section.Title	Introduction, product lines, cleaners	Introduction, Configuration, installation

Now we merge all the vectors of weighted terms describing a logical element in one single vector. Thus the document index will be:

$$d_d = (w_{1,d}t_1, \dots, w_{i,d}t_i, \dots, w_{N,d}t_N)$$

d_d is the document d

x is the number of logical element types found in the corpus of documents.

N is the number of candidate terms found in the corpus of documents.

In this vector the weight of term i in whole of document d according to its weights $w_{i,e,d}$ in each logical element in d will be as follows:

$$w_{i,d} = \prod_{e=1}^{x} w_{i,e,d} * ilf_{d,e}$$

$$\mathrm{ilf}_{\mathrm{d,e}} = \log \frac{\mathrm{L_d}}{\mathrm{l_{d,e}}}$$

$\mathrm{L_d} =$ total number of levels from root to last node in the tree of document d

$\mathrm{l_{d,e}} =$ the level of the logical element e in the tree of document d.

X= the number of logical element types containing $t_{i.}$

$\mathrm{ilf_{d,e}}$ is a factor for integrating the importance of each type of logical element in the document. For example the title of document can be more representative than any paragraph. For example once a term like “user guide” appears in a title of document it has a higher coefficient than when it appears in the section of that document. The distance of title from the root is equal to 1, which is the nearest level to the root.

So, in our example the weight of the term “introduction” in the whole document will be:

$$\mathrm{w("introduction", Doc\ 1)} = \left(0.3 * \log\left(\frac{4}{1}\right)\right) + \left(0 * \log\left(\frac{4}{2}\right)\right) = 0.18$$

4 is the number of levels in the example document.

3.3 Concept detection

The traditional IR systems do not consider the semantic of the document content. They just consider the whole document as a bag of words. But as we know each term has different meaning and can explain different concepts. In the other hand, a concept can be expressed by different terms. This multi meaning leads to ambiguity in text. For extracting the concepts of documents external semantic resources have been vastly used.

In previous sections of article, we extracted the terms and computed their weight by considering the structure of document, but we have not yet taken in to account the notion of concept in document. For considering the document concept, we propose using a semantic resource (which adapts to the domain of our corpus) to extract the document semantic, in order to obtain an index which is more relevant.

For finding the concepts, we do mapping between the candidate terms and the concepts in semantic resource. Because of the ambiguity which exists in terms by their nature, we will obtain a set of concepts for each candidate term. So, we call this mapping function SM for semantic mapping.

$$\mathrm{SM(t_i)} = \{\mathrm{C_1}, \dots, \mathrm{C_k}, \dots, \mathrm{C_P}\}$$

Where t_i is a candidate term in the document and P is the number of concepts which are associated to this term according to the semantic resource. Also each concept can be expressed by other terms in the document.

We defined the inverse function called LM for Lexical Mapping

$$\mathrm{LM}(\mathrm{C_k}) = \{\mathrm{t_1}, \dots, \mathrm{t_i}, \dots, \mathrm{t_Q}\}$$

Where C_k is a candidate concept for the document and Q is the number of terms label of this concept in the semantic resource which exist in the document.

To overcome the ambiguity we have to choose for each term t_i the best concept in SM which represents the document semantic. Therefore, we assign a score for each concept of SM (t_i), and then we choose the concept with the highest score. We use the C_score define in (BAZIZ M., 2005).

C_k^i is the k^{th} concept associated to the term t_i in the semantic resource

$$C_{score}(C_k^i) = \sum_{l \in [1..F], l \neq i, j \in [1..P]} dis_{i,l}(C_k^i, C_j^l)$$

Where F is the number of terms in the document and P the number of concepts associated to term t_l.

$dis_{i,l}(C_k^i, C_j^l)$ is a function which measures the similarity between the concepts in a semantic resource. There are different works for measuring this similarity. We propose the formula of Leacock (Leacocke C., 1998) in which he measures the shortest path between concepts

$$dis_{i,l}(C_k^i, C_j^l) = \max\,[-\log\,(\mathrm{lenght}(\mathrm{C_k}, \mathrm{C_j})/2.\,\mathrm{G})]$$

The shortest Length between two concepts corresponds at the nodes intermediate between these concepts, and G is the longest distance between the root and the leaf in the semantic resource.

After calculating the score of all competitor concepts for each term, we choose the concept which has the greatest score between these concepts:

$$\text{Selected\_Concept}\,(t_i) = \max\,(C_{score}(C_k^i))$$

Thus after identifying the candidate concepts of document, the document index will follow this formula:

$$d_d = (wc_{1,d}C_1, \dots, wc_{k,d}C_k, \dots, wc_{M,d}C_M)$$

Where $wc_{k,d}$ is the weight of the concept C_k for the document d and M is the number of candidate concepts in the document d.

$$wc_{k,d} = \sum_{t_i \in LM(C_k)} w_{i,d}$$

Where $w_{i,d}$ is the weight of term t_i which appears in the document d. C_k is selected concept of t_i., and t_i belongs to $\mathrm{LM}(\mathrm{C_k})$.

As the most works show, indexing based on concepts in addition to the terms of document offers the best result comparing by indexing just by concepts.

Thus the final index looks like:

$$d_d = (wc_{1,d}C_1, \dots, wc_{k,d}C_k, \dots, wc_{M,d}C_M,\ w_{1,d}t_1, \dots, w_{i,d}t_i, \dots, w_{N,d}t_N)$$

4 Conclusion

In this article we have proposed a model for conceptual indexing the structured documents based on an semantic resource. We separate the document in logical elements as a tree. Then by doing some basic analysis on the text and projecting it onto WordNet we identify the candidate terms in each logical element. We proposed a weight computation formula according to structure in which the terms appear. Then by using a semantic resource we obtain the concept of document and enrich our index.

The corpus which we are working on it is the corpus of the structured documents of Continew company. One limitation of our approach in structure aspect is converting the whole corpus in XML format for achieving the tags of structure. The other limitation in semantic aspect is constructing a domain ontology which corresponds to our domain and corpus and contains all related concepts and instances. In future work we are going to evaluate WordNet utilization for term extraction. And then we are going to construct an ontology of the corpus domain and use it to achieve the document concept. At the end by using the index we have to classify the documents in predefined classes.

5 References

BAZIZ M. (2005). *Indexation Conceptuelle Guidée par Ontologie pour la Recherche d'Information.* Thèse Informatique, INSTITUT DE RECHERCHE EN INFORMATIQUE DE TOULOUSE, Toulouse.

Baziz M., B. M.-G. (2005). Evaluating a Conceptual Indexing Method by Utilizing WordNet.

Guarino N., M. C. (1999). OntoSeek: Using Large Linguistic Ontologies for Accessing On-Line Yellow Pages and Product Catalogs.

Kazai G., L. M. (2002). Focussed Structured Doccument Retrieval. 241-247.

Khan L. (2004). Retrieval effectiveness of an ontology-based model for information selection. (Springer, Ed.) (13), 71–85.

Lalmas M. (2000). Uniform representation of content and structure for structured document retrieval. *20th SGES International Conference on Knowledge Based Systems and Applied Artificial Intelligence* .

Leacocke C., M. G. (1998). Using corpus statistics and WordNet relations for sens identification. (C. Linguist, Ed.) 147-165.

Mass Y., M. M. (2004). Component ranking and automatic query refinement for XML retrieval. *INEX 2004* , 134–140.

Mass Y., M. M. (2003). Retrieving the most relevant XML Component. *the Second Workshop of the Initiative for The Evaluation of XML Retrieval (INEX)* , 53-58.

Miller G. (1995). *Wordnet: A lexical database.* New York: Communication of the ACM, 38(11).

Myaeng S., J. D. (1998). A flexible model for retrieval of SGML documents. 138-145.

Pinel-Sauvagnat K., B. M. (2006). Propositions pour la pondération des termes et l'évaluation de la pertinence des éléments en recherche d'information structurée. (Cépaués, Ed.) 77-98.

Salton G. (1968). Search and retrieval experiments in real-time information retrieval. (C. University, Ed.) 1082-1093.

Schileder T., M. H. (2002). Querying and ranking XML documents. *Journal of the American Society for Information Science and Techno-logy* , 489–503.

Voorhees E. (1993). Using WordNet to Disambiguate Word Senses for Text Retrieval.

Wilkinson R. (1994). Effective retrieval of structured documents. (S.-V. N. York, Ed.) 311 – 317.

Zargayouna H. (2004). Contexte et sémantique pour une indexation de documents semi-structurés. *CORIA 04* , 161–178.