Editorial

SEMANTIC WEB and intelligent text processing technologies involved in it are crucial for today's information infrastructure. Brilliant success of information retrieval and machine translation, as well as rapid progress in opinion mining and sentiment analysis for decision making and recommender systems, make these topics, as well as their underlying technology and research, especially important for the readers of this journal. It is my pleasure to present to the readers an issue of Polibits featuring a thematic section on Semantic Web and Intelligent Text Processing.

The thematic section contains nine papers written by authors from seven countries: France, India, Mexico, Romania, Spain, Switzerland, and USA. In addition, the issue contains a regular paper.

The first two papers are directly connected with treatment of information in webpages.

López, Silva, and Insa (Spain) address a very important problem of automatic separation of useful text on a webpage from noise, mainly advertisement, which nowadays floods nearly any webpage. They describe a simple yet powerful idea of using the HTML tree structure to identify large blocks of text, which very probably represent useful text and not advertisements and other noise.

Schilder, Kondadadi, and Kadiyska (USA) explain how to extract tabular data from pieces of text that look like tables for humans but are not structured well enough to allow trivial technique for recognition of the structure. They use geometric approach based on spatial proximity of the pieces of text representing table cells in two dimensions, starting from a corner of the table and iteratively pulling the cells adjacent to those already identified.

The next two papers deal with string alignment problems: when you have two different strings and you want to identify what they have in common.

Dănăilă, Dinu, Niculae, and Șulea (Romania) present a detailed survey of a number of different string comparison measures and study their performance in an important task: identifying different expressions that refer to the same thing, such as different spellings of the name of the same person or product, or different ways to express the same address. This is an important task because the presence of huge amount of such duplicates in large databases prevents us from correct analysis and handling of those databases.

Nicolas Béchet and Marc Csernel (France) use string alignment technique for comparison of different versions of the same, or nearly the same, text in Sanskrit. A particular difficulty of comparing Sanskrit documents is that text in Sanskrit is written without spaces between words, and some parts of text can be freely moved without changing the meaning of the text. I believe their technique will be interested not only for historians and philologists but also for those who deal with genetic sequences: DNA structures exhibit similar properties.

The next two papers deal with very important extralinguistic phenomena that are present in huge quantities in the (semantic) Web: emotions and references to locations.

Loza-Pacheco, Torres-Ruiz and Guzmán-Lugo (Mexico) identify the location to which a map, a photo, or a toponym (name of a place) belongs. They use knowledge-based techniques and ontologies to reason about spatial relationships between objects and thus their names and thus to bring order and meaning in geographically-related databases. While the paper is written in Spanish, it provides a English abstract.

Das and Bandyopadhyay (India) extend analysis of emotions expressed by the authors of blog messages to a new language, in this case Bengali, which is, according to different accounts, the fourth or fifth world's most spoken language, accounting for approximately 200 million speakers. Analysis of emotions and sentiments expressed in blogs and social networks is currently probably the hottest topic in natural language processing and web-related studies.

The next three papers deal with semantics of natural language.

Martins (Switzerland) discusses issues related to an extension of the Universal Networking Language (UNL, see www.cicling.org/2005/UNL-book for an introduction) to a knowledge representation language called XUNL, and draws some important conclusions and guidelines about the desired structure and properties of such semantic language.

Castro-Sánchez and Sidorov (Mexico) present a novel method for building formal syntactico-semantic structures that describe the roles of nouns that they play in a situation expressed by a verb, and how these constructions are expressed in natural language texts. They build such a formal lexical resource out of existing dictionaries oriented to human readers, which do not allow direct use of the information they contain by computer programs.

Dinu (Romania) closes the thematic section on Semantic Web and Intelligent Text Processing with a paper devoted to particular issues in formal semantics, such as scope taking and quantification, expressed in precise mathematical form. These issues, that have been in the core of formal semantics research during decades if not centuries, are discussed in the paper in the context of a specific semantic theory called continuation semantics.

Apart from the thematic section, this issue of Polibits includes a regular paper unrelated to the thematic section but directly related with the topic of the journal.

Jiménez, Sossa, Cuevas, and Gómez apply well-known artificial intelligence technique called particle swarm optimization to an important practical technical problem: interferometry, which is a task of non-destructive optical measuring of dimensions of a physical object with very high precision comparable with the wavelength of light. They introduce the reader to the practical task with a clear explanation of the problem, and then explain how the artificial intelligence technique is used to solve this practical problem.

I would like to thank the Editorial Board of Polibits for inviting me to serve as a guest editor of the journal and express my hope that the papers selected for this issue will prove to be interesting and useful for all readers working in, or interested in, Computer Science in general, Artificial Intelligence, and specifically Text Processing and Semantic Web.

Dr. Niladri Chatterjee

Associate Professor,
Indian Institute of Technology Delhi,
New Delhi, India

Guest Editor

Content Extraction based on Hierarchical Relations in DOM Structures

Sergio López, Josep Silva, and David Insa

***Abstract*—This article introduces a new approach for content extraction that exploits the hierarchical inter-relations of the elements in a webpage. Content extraction is a technique used to extract from a webpage the main textual content. This is useful in order to filter out the advertisements and all the additional information that is not part of the main content. The main idea behind our approach is to use the DOM tree as an explicit representation of the inter-relations of the elements in a webpage. Using the information contained in the DOM tree we can identify blocks of content and we can easily determine what of the blocks contains more text. Thanks to this information, the technique achieves a considerable recall and precision. Using the DOM structure for content extraction gives us the benefits of other approaches based on the syntax of the webpage (such as characters, words and tags), but it also gives us a very precise information regarding the related components in a block, thus, producing very cohesive blocks.**

***Index Terms*—Content Extraction, Block Detection, DOM**

I. Introduction

CONTENT Extraction is one of the major areas of interest in the Web for both the scientific and industrial communities. This interest is due to the useful applications of this discipline. Essentially, content extraction is the process of determining what parts of a webpage contain the main textual content, thus ignoring additional context such as menus, status bars, advertisements, sponsored information, etc. Content extraction is a particular case of a more general discipline called *Block Detection* that tries to isolate every information block in a webpage. For instance, observe the blocks that form the webpage in Figure 1, and in particular, the main block delimited with a dashed line. Note that inside the main block there are other blocks that should be discarded.

It has been measured that almost 40-50% of the components of a webpage can be considered irrelevant [1]. Therefore, determining the main block of a webpage is very useful for indexers and text analyzers to increase their performance by only processing relevant information. Other interesting applications are the extraction of the main content of a webpage to be suitably displayed in a small device such as a PDA or a mobile phone; and the extraction of the relevant content to make the webpage more accessible for visually impaired or blind.

Manuscript received on October 21, 2011, accepted for publication on December 9, 2011.

The authors are with the Departamento de Sistemas Informáticos y Computación, Universitat Politècnica de València, E-46022 Valencia, Spain (e-mail:{slopez,jsilva,dinsa}@dsic.upv.es).

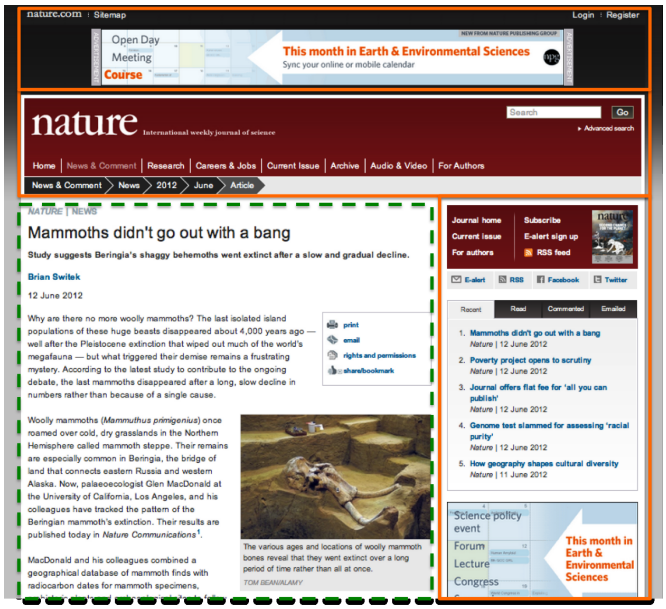


Fig. 1. Blocks of a webpage from the Nature website

Our technique combines ideas from other works such as [2], [3], and it also uses additional information that is explicit in the DOM tree of webpages, and that allows the technique to produce very accurate results.

In summary, the main advantages of our technique are the following:

- It does make no assumptions about the particular structure of webpages.
- It only needs to process a single webpage (no templates, neither other webpages of the same website are needed).
- No preprocessing stages are needed. The technique can work online.
- It is fully language independent (it can work with pages written in English, German, etc.).
- The particular text formatting of the webpage does not influence the performance of the technique.

The rest of the paper has been structured as follows: In Section II we discuss the state of the art and show some problems of current techniques that can be solved with our approach. In Section III we recall the DOM model and provide some useful notation. Then, we present our algorithms and explain the technique with examples in Section IV. In Section V we give some details about the implementation and

show the results obtained with a collection of benchmarks. Finally, Section VI concludes.

II. Related Work

Many different techniques have been proposed to solve the problem of content extraction. Some of them are based on the assumption that the webpage has a particular structure (e.g., based on table markup-tags) [4], that the main content text is continuous [5], that the system knows a priori the format of the webpage [4], or even that the whole website to which the webpage belongs is based on the use of some template that is repeated [6]. This allows the system to analyze several webpages and try to deduce the template of the webpage in order to discard menus and other repeated blocks.

The main problem of these approaches is a big loss of generality. In general, they require to previously know or parse the webpages, or they require the webpage to have a particular structure. This is very inconvenient because modern webpages are mainly based on $<$*div*$>$ tags that do not require to be hierarchically organized (as in the table-based design). Moreover, nowadays, many webpages are automatically and dynamically generated and thus it is often impossible to analyze the webpages a priori.

There are, however, other approaches that are able to work online (i.e., with any webpage) and in real-time (i.e., without the need to preprocess the webpages or know their structure). One of these approaches is the technique presented in [2]. This technique uses a *content code vector* (CCV) that represents all characters in a document determining whether they are content or code. With the CCV, they compute a *content code ratio* to identify the amount of code and content that surrounds the elements of the CCV. Finally, with this information, they can determine what parts of the document contain the main content. Another powerful approach also based on the labeling of the characters of a document has been presented in [3]. This work is based on the use of *tag ratios* (TR). Given a webpage, the TR is computed for each line with the number of non-HTML-tag characters divided by the number of HTML-tags. The main problem of the approaches based on characters or lines such as these two, or words such as [7], is the fact of completely ignoring the structure of the webpage. Using characters or words as independent information units and ignoring their interrelations produces an important loss of information that is present and explicit in the webpage, and that makes the system to fail in many situations.

Example 2.1: Consider the portion of a source code extracted from a Fox News webpage shown in Fig. 2.1.
The tag ratios associated to this webpage are shown in Figure 3. Observe that the initial part of the footer (which is not part of the main content) is classified as part of the main content because it starts with a high tag ratio. Unfortunately, this method does not take into account the information provided by tags, and thus, it fails to infer that the footer text belongs to a different $< div >$ than the other text classified as relevant.

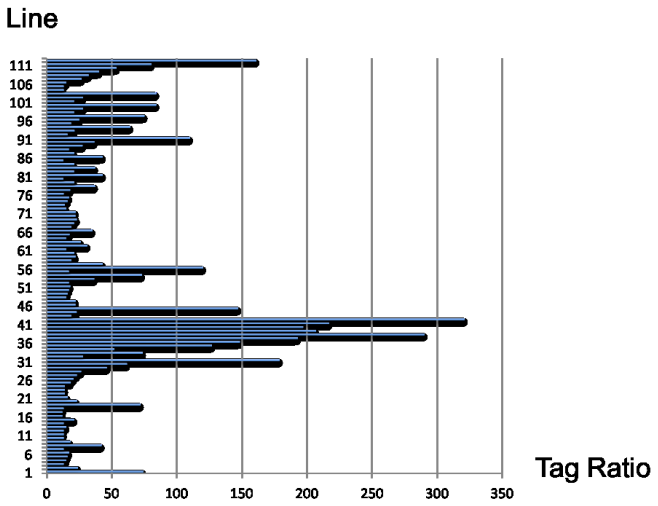


Fig. 3. Tag ratios associated with the code in Example 2.1

The distribution of the code between the lines of a webpage is not necessarily the one expected by the user. The format of the HTML code can be completely unbalanced (i.e., without tabulations, spaces or even carriage returns), specially when it is generated by a non-human directed system. As a common example, the reader can see the source code of the main Google's webpage. At the time of writing these lines, all the code of the webpage is distributed in only a few lines. In this kind of webpages tag ratios are useless.

In this work, we solve this situation by using a ratio similar to the tag ratio but based on the DOM structure of the webpage. This makes our approach keep the good properties of the tag ratios approach, but it also solves the problems shown in the previous example because the technique combines the computed ratios with the information of the DOM tree. In particular, because the DOM tree is independent of the distribution of the code between the lines of the HTML webpage, our technique is able to work with any webpage independently of how the webpage was generated or formatted.

It should be clear that—as it happens in the other approaches—the technique could fail to detect the main block if other block (e.g., the footer) contains more text density that the real main block. But our technique easily distinguishes between different blocks (thanks to the DOM information), and it does not mix information from different blocks as in Example 2.1.

Although essentially different to our work, there exist other techniques that make use of the DOM structure, and thus, they could exploit the same information than our technique. The most similar approach is the one presented in [8]. This approach presents a proxy server that implements a set of filters for HTML documents. These filters include HTML cleaning (e.g., removing images, scripts, etc.), HTML refactoring (e.g., removing empty tables), and deleting advertisements (e.g., with a blacklist of URLs that allow

```
<body>
 (...)
 <div id="article-section" class="hnews hentry item">
  <h1 id="article-title" class="entry-title">Alleged
victim tells court Sandusky (...)</h1>
  <div class="article-text">
   <title="Abstract">Sandusky trial continues after
yesterday‘s testimony from alleged victim | Fox News</title>
   <p>Sandusky has displayed no visible emotion (...)</p>
   <p>The man identified as "Victim 10" by (...)</p>
   <p>The man, now 25, had been in foster care (...)</p>
  </div>
 </div>
 <div id="section-footer">
  <p class="published">Published June 13, 2012</p>
  <p class="summary">This material may not be published,
broadcast, rewritten, or redistributed. FOX News Network.
All rights reserved. All market data delayed 20 minutes</p>
 </div>
 (...)
</body>
```

Fig. 2. Code extracted from a Fox News webpage

them to remove external publicity content). Some of these transformations are used by our technique, but the objective is different, we do not want to clean, improve or transform the original webpage; our goal is to detect the main content and remove all the other components of the webpage. Also the implementation is different, our tool is not based on a proxy server; it is implemented in the client side, and thus it is independent of any external resource.

There are some approaches specialized for a particular content such as tables that are somehow related to our work. They do not focus on block detection but in content extraction from tables [9], or in wrappers induction [10], [11]. Other related approaches are based on statistical models [12], [13] and machine learning techniques [14], [15] and they use densitometric features such as link density and text features such as number of words starting with an uppercase letter [16].

III. The DOM tree

The Document Object Model (DOM) [17] is an API that provides programmers with a standard set of objects for the representation of HTML and XML documents. Our technique is based on the use of DOM as the model for representing webpages. Given a webpage, it is completely automatic to produce its associated DOM structure and vice-versa. In fact, current browsers automatically produce the DOM structure of all loaded webpages before they are processed.

The DOM structure of a given webpage is a tree where all the elements of the webpage are represented (included scripts and CSS styles) hierarchically. This means that a table that contains another table is represented with a node with a successor that represents the internal table. Essentially, nodes in the DOM tree can be of two types: tag nodes, and text nodes.[1] Tag nodes represent the HTML tags of a HTML document and they contain all the information associated with the tags (e.g., its attributes). Text nodes are always leaves in the DOM tree because they cannot contain other nodes. This is an important property of DOM trees that we exploit in our algorithms.

Definition 3.1 (DOM Tree): Given an HTML document D, the DOM tree $t = (N, E)$ of D is a pair with a finite set of nodes N that contain either HTML tags (including their attributes) or text; and a finite set of edges E such that $(n \rightarrow n') \in E$, with $n, n' \in N$ if and only if the tag or text associated with n' is inside the tag associated with n in D. The reflexive and transitive closure of E is represented with E^*.

For the purpose of this work, it does not matter how the DOM tree is built. In practice, the DOM's API provides mechanisms to add nodes and attributes, and provides methods to explore and extract information from the tree.

Example 3.2: Consider again the source code from Example 2.1. A portion of the associated DOM tree is depicted in Figure 4. For the time being the reader can ignore the different colors and borders of nodes.

IV. Content extraction using DOM trees

In this section we formalize our technique for content extraction. The technique is based on the notion of *chars-nodes ratio* (CNR), which shows the relation between text content and tags content of each node in the DOM tree.

[1]We make this assumption for simplicity of presentation. In the current DOM model, there are 12 types of nodes, including the type text.

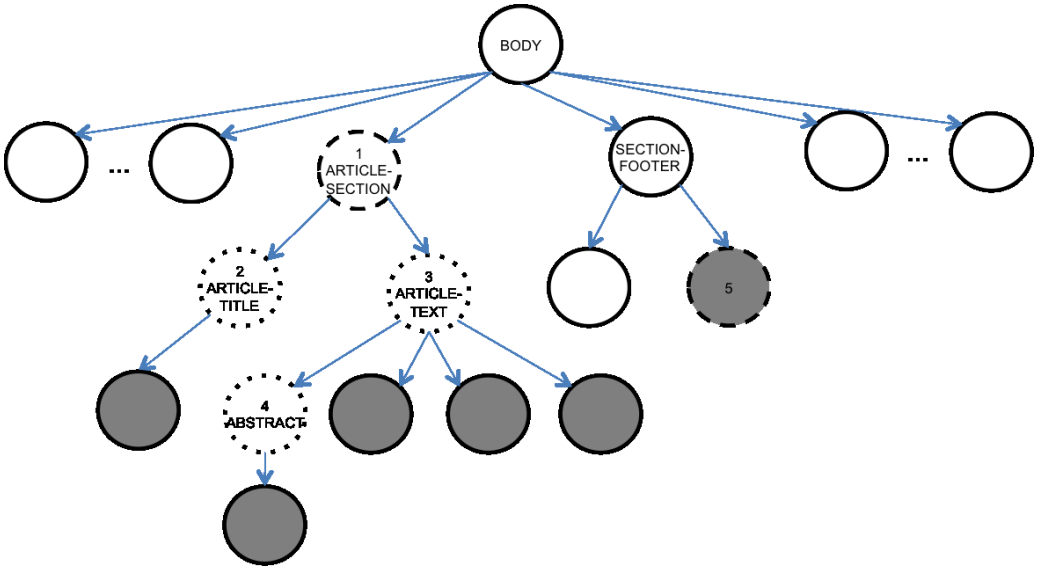


Fig. 4. DOM representation of the Fox News webpage

Definition 4.1 (chars-nodes ratio): Given a DOM tree (N, E), a node $n \in N$ and the set of nodes $M \subseteq N$ that form the subtree rooted at n ($M = \{n' \in N \mid (n \rightarrow n') \in E^*\}$), the *chars-nodes ratio* of n is $chars/weight$; where $chars$ is the number of characters in the text nodes of M, and $weight = |M|$.

The interesting property of this definition, is that it considers nodes as blocks where the internal information is grouped and indivisible using the DOM structure. Therefore, the CNR of an internal node, takes into account all the text and tags included in its descendants. Note also that the CNR of a node n, $\mathrm{CNR}(n)$, with a single child n_1 is always smaller than $\mathrm{CNR}(n_1)$ because n can not contain text. However, if n has several children $n_1 ... n_c$, then $\mathrm{CNR}(n)$ can be greater than $\mathrm{CNR}(n_1)$ depending on the amount of text in the other children. This is very useful, because it allows us to detect blocks of relevant content, even if some nodes without text belong to the block.

Now, we are in a position to describe our method for content extraction. (i) We first compute the CNR for each node in the DOM tree. Then, (ii) we select those nodes with a higher CNR and, starting from them, we traverse the DOM tree bottom-up to find the best container nodes (e.g., tables, divs, etc.) that, roughly, contain as more relevant text as possible and less nodes as possible. Each of these container nodes represents an HTML block. Finally, (iii) we choose the block with more relevant content. All three steps can be done with a cost linear with the size of the DOM tree.

The first step is computed with a cost $\mathcal{O}(|N|)$. With a single traversal of the tree, it ignores irrelevant code that should not be counted as text (such as Javascript), and it computes the CNRs. Even though, the computation of the CNR seems to be trivial because the DOM model's API has a method to obtain the text content of a node, this method cannot discriminate between different kinds of text contents (e.g., plain text, scripts, CSS...). Moreover, there does not exist a method to calculate the number of descendants of a given node; therefore, the computation of CNRs is done with a cumulative and recursive process that explores the DOM tree counting the text and descendants of each node. This process also allows us to detect irrelevant nodes that we call "nonContentNode". They are, for instance, nodes without text (e.g., *img*), nodes mainly used for menus (e.g., *nav* and *a*) and irrelevant nodes (e.g., *script*, *video* and *svg*). This is an important advantage over other techniques that rely on the analysis of single characters or lines. These techniques cannot ignore the noisy code if they do not perform a pre-processing stage to delete these tags.

Algorithm 1 recursively obtains the CNR of each node starting at the root node of the DOM tree. At each node it adds three new attributes to the node with the computed weight (*weight*), the number of characters it contains (*textLength*), and the CNR (*CNR*). The number of characters is computed ignoring special characters such as spaces or line breaks. This makes the algorithm independent of the formatting of the webpage (e.g., those webpages that organize the code using several spaces do not influence the CNRs).

The algorithm distinguishes between three kinds of nodes, namely *textNode* which is a kind of DOM node that contains plain text and that is always a leaf, thus, it has weight 1; *nonContentNode* that represents irrelevant nodes with a CNR of 0; and the rest of nodes that represent all kinds of tags. All methods (such as *addAttribute*) and attributes (such as *innerText*) used in the algorithm are standard in the DOM model and have the usual meaning.

Once the CNRs are calculated, in the second step we select those nodes with a higher CNR. Then, we propagate these nodes bottom up to discover the blocks to which they belong,

Algorithm 1 Algorithm to compute chars-nodes ratios

Input: A DOM tree $T = (N, E)$ and the root node of T, $root \in N$
Output: A DOM tree $T' = (N', E)$

```
computeCNR(root)

function ComputeCNR(node n)
  case n.nodeType of
  "textNode":
       n.addAttribute('weight',1);
       n.addAttribute('textLength',n.innerText.length);
       n.addAttribute('CNR',n.innerText.length);
       return n;
  "nonContentNode":
       n.addAttribute('weight',1);
       n.addAttribute('textLength',0);
       n.addAttribute('CNR',0);
       return n;
  otherwise:
       descendants = 1;
       charCount = 0;
       for each child ∈ n.childNodes do
           newChild= ComputeCNR(child);
           charCount = charCount + newChild.textLength;
           descendants = descendants + newChild.weight;
       n.addAttribute('weight',descendants);
       n.addAttribute('textLength',charCount);
       n.addAttribute('CNR',charCount/descendants);
       return n;
```

and the block with more text is selected. This means that if some nodes not belonging to the main block are included in the selected nodes, they will be discarded in the next steps.

The computation of the container blocks is performed with Algorithm 2. Roughly, this algorithm takes the DOM tree and the set of nodes identified in the previous step, and it removes all the nodes in the set that are descendant of other nodes in the set (line (1)). Then, in lines (2) and (3), it proceeds bottom-up in the tree by discarding brother nodes and collecting their parent until a fix point is reached. This process produces a final set of nodes that represent blocks in the webpage. From all these nodes, we take the one that contains more text (in the subtree rooted at that node) as the final block.

Example 4.2: Consider again the HTML code from Example 2.1 and its associated DOM tree shown in Figure 4. Algorithm 1 computes the CNR associated to each node of the DOM tree. All the CNRs are shown in Figure 5.

After we have computed the CNRs we take the top rated nodes. Let us consider that the dark nodes in Figure 4 represent the top rated nodes. Then, we use Algorithm 2 to identify the most relevant blocks in the webpage. Initially, all the dark nodes are in the set of blocks. Then, because nodes 7 and 8 are brothers, in the first iteration, the algorithm removes nodes 7 and 8, and it adds node 6 to the set. In the second iteration, nodes 5 and 6 are removed, and node 2 is added.

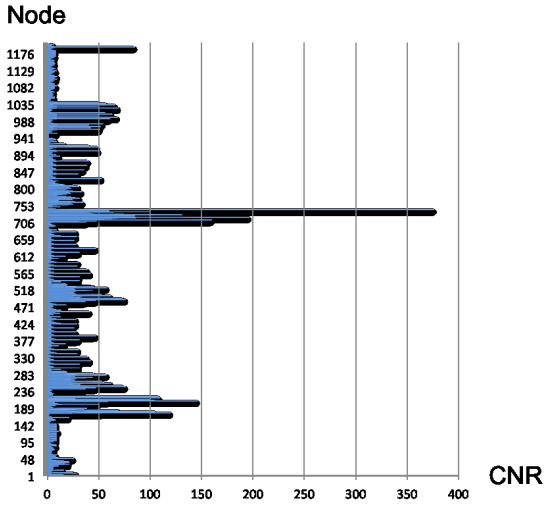


Fig. 5. Char-Nodes Ratios associated with the code in Example 2.1

Finally, in the third iteration, nodes 2 and 3 are removed, and node 1 is added. Therefore, the final set of nodes computed by Algorithm 2 only contains the dashed nodes (1 and 4). The node that contains more text is selected as the block with the main content of the webpage. Observe that due to the structure

Algorithm 2 Identifying main content blocks

Input: A DOM tree $T = (N, E)$ and a set of nodes $S \subset N$
Output: A set of nodes $blocks \subset N$
Initialization: $blocks = S$

(1) $blocks = blocks \backslash \{b \mid (b' \rightarrow b) \in E^*$ with $b, b' \in blocks\}$
(2) while $(\exists n \in N \ . \ (n \rightarrow b), (n \rightarrow b') \in E$ with $b, b' \in blocks)$
(3) $\quad blocks = (blocks \backslash \{b \mid (n \rightarrow b) \in E\}) \cup \{n\}$

return $blocks$

of the DOM tree, the final node is often a container tag. Note also that all the nodes of this container are part of the final block, even if they do not contain text. Therefore, the final block is a block with all the information of the initial webpage that was placed together by the designer including related but non-textual elements such as images. The information of other blocks is not mixed with the information of the main block due to the structure of the DOM tree. For instance, node 4 corresponds to the footer, and it contains a lot of text. However, although it is textually adjacent in the source code to some nodes included in the main block; it is outside the container selected as the main content block (node 1). Therefore, node 4 is finally discarded.

V. Implementation

We have implemented the technique presented in this paper and made it publicly available, including the source code. It was implemented as a Firefox's plugin that can be installed in the Firefox's browser as a toolbar. Then, it can filter any loaded webpage or produce information about the CNRs of the DOM tree.

The implementation allows the programmer to activate the transformations of the technique and to parameterize them in order to adjust the amount of blocks retrieved, and the thresholds used to detect these blocks. In order to determine the default configuration, it was initially tested with a collection of real webpages that allowed us to tune the parameters. Then, we conducted several experiments with real and online webpages to provide a measure of the average performance regarding recall, precision and the F1 measure (see, e.g., [18] for a discussion on these metrics).

For the experiments, we selected from the top-most 500 visited webpages (see http://www.alexa.com/topsites) a collection of domains with different layouts and page structures in order to study the performance of the technique in different contexts (e.g., company's websites, news articles, forums, etc.). Then, we randomly selected the final evaluation set. We determined the actual content of each webpage by downloading it and manually selecting the main content text. The DOM tree of the selected text was then produced and used for comparison evaluation later.

Table I summarizes the results of the performed experiments. The first column contains the URLs of the evaluated webpages. For each benchmark, column `DOM nodes` shows the number of nodes of the whole DOM tree associated to this benchmark; column `Main block` shows the number of nodes that were identified by the tool as the main block; column `Recall` shows the number of relevant nodes retrieved divided by the total number of relevant nodes (i.e., in the main block); column `Precision` shows the number of relevant nodes retrieved divided by the total number of retrieved nodes; Finally, column `F1` shows the F1 metric that is computed as $(2 * P * R)/(P + R)$ being P the precision and R the recall.

Experiments reveal that in many cases, the retrieved block is exactly the relevant block (F1=100%), and in general, the recall is 100%. This means that the retrieved block often contains all the relevant information. The average recall is 94.39 and the average precision is 74.08. These are really good measures. For instance, with the same webpages, the best previous technique (using tag ratios [3]) produces an average recall of 92.72 and an average precision of 71.93.

Observe one important property of the experiments: in all cases, either the recall, the precision, or both, are 100%. This phenomenon did not happen by a chance, it is a direct consequence of the way in which the technique selects blocks. Let us consider a DOM tree where node n is the actual relevant block. Our technique explores the DOM tree bottom-up to find this node, and only four cases are possible: (1) If we detect node n as the main block, then both recall and precision are 100%. (2) If we choose a node that is a descendant of n, then precision is 100%. (3) If we choose a node that is an ancestor of n, then recall is 100%. Finally, (4) if we select a node that is not an ancestor neither a descendant of n then both recall and precision would be 0%. This case is very rare because this would mean that there exists a non-relevant block that contains more text than the relevant block. This never happened in all the experiments we did.

We could take advantage of this interesting characteristic of our technique. We could parameterize the technique to ensure that we have a 100% recall, or to ensure that we have a 100% precision depending on the applications where it is used. This can be easily done by making Algorithm 2 to be more restrictive (i.e., selecting blocks closer to the leaves, thus, ensuring 100% precision), or more relaxed (i.e., selecting blocks closer to the root, thus, ensuring 100% recall).

TABLE I
BENCHMARK RESULTS

Benchmark	DOM nodes	Main block	Recall	Precision	F1
www.wikipedia.org	870 nodes	712 nodes	100 %	100 %	100 %
www.facebook.com	744 nodes	293 nodes	28.6 %	100 %	44.47 %
www.nytimes.com	742 nodes	217 nodes	100 %	49.7 %	66.39 %
www.engadget.com	2897 nodes	1345 nodes	100 %	100 %	100 %
us.gizmodo.com	2205 nodes	1375 nodes	100 %	84 %	91.3 %
googleblog.blogspot.com	1138 nodes	743 nodes	100 %	100 %	100 %
www.bbc.co.uk	401 nodes	111 nodes	100 %	4.98 %	9.49 %
www.vidaextra.com	1144 nodes	602 nodes	100 %	100 %	100 %
www.gizmologia.com	926 nodes	415 nodes	100 %	100 %	100 %
www.elpais.com	3017 nodes	120 nodes	100 %	100 %	100 %
www.elmundo.es	1722 nodes	416 nodes	100 %	100 %	100 %
www.ox.ac.uk	279 nodes	30 nodes	100 %	28 %	43.75 %
www.thefreedictionary.com	1170 nodes	509 nodes	100 %	100 %	100 %
www.nlm.nih.gov	320 nodes	156 nodes	100 %	56.52 %	71.56 %
www.scielosp.org	563 nodes	458 nodes	100 %	100 %	100 %
www.wordreference.com	269 nodes	95 nodes	100 %	57.23 %	72.79 %
en.citizendium.org	1645 nodes	1478 nodes	100 %	100 %	100 %
knol.google.com	601 nodes	219 nodes	100 %	100 %	100 %
www.healthopedia.com	557 nodes	21 nodes	100 %	21 %	34.7 %
www.filmaffinity.com	1198 nodes	153 nodes	100 %	100 %	100 %
www.umm.edu	290 nodes	30 nodes	100 %	22.22 %	36.42 %
www.microsiervos.com	604 nodes	382 nodes	100 %	68.83 %	81.54 %
abcnews.go.com	907 nodes	102 nodes	100 %	44.16 %	61.27 %
www.latimes.com	1056 nodes	22 nodes	100 %	100 %	100 %
www.philly.com	378 nodes	30 nodes	100 %	100 %	100 %
www.blogdecine.com	1567 nodes	24 nodes	100 %	8.33 %	15.38 %
www.cnn.com	597 nodes	248 nodes	100 %	67.21 %	80.39 %
www.lashorasperdidas.com	87 nodes	30 nodes	100 %	100 %	100 %
www.cbc.ca	847 nodes	138 nodes	100 %	100 %	100 %
www.appleweblog.com	1013 nodes	475 nodes	5.9 %	100 %	11.15 %
www.applesfera.com	1215 nodes	721 nodes	7.49 %	100 %	13.94 %
www.guardian.co.uk	1111 nodes	59 nodes	100 %	100 %	100 %
www.news.cnet.com	2023 nodes	169 nodes	100 %	71.01 %	83.05 %
www.venturebeat.com	263 nodes	107 nodes	100 %	100 %	100 %
www.computerworld.com	558 nodes	62 nodes	100 %	100 %	100 %
www.usatoday.com	1118 nodes	523 nodes	100 %	100 %	100 %
www.cbssports.com	1450 nodes	232 nodes	100 %	67.05 %	80.28 %
www.nationalfootballpost.com	565 nodes	23 nodes	100 %	9.62 %	17.55 %
ncaabasketball.fanhouse.com	885 nodes	78 nodes	100 %	40.20 %	57.35 %
www.sportingnews.com	1394 nodes	79 nodes	100 %	72.48 %	84.05 %
www.hoopsworld.com	629 nodes	112 nodes	100 %	100 %	100 %
profootballtalk.nbcsports.com	394 nodes	28 nodes	100 %	45.17 %	62.23 %
www.thehollywoodgossip.com	362 nodes	44 nodes	100 %	100 %	100 %
www.rollingstone.com	993 nodes	29 nodes	100 %	20.42 %	33.92 %
popwatch.ew.com	919 nodes	93 nodes	100 %	100 %	100 %
www.people.com	923 nodes	56 nodes	100 %	32 %	48.49 %
www.cinemablend.com	495 nodes	59 nodes	100 %	37.34 %	54.38 %

All the information related to the experiments, the source code of the benchmarks, the source code of the tool and other material can be found at http://users.dsic.upv.es/~jsilva/CNR.

VI. CONCLUSIONS

Content extraction is useful not only for the final user, but also for many systems and tools such as indexers as a preliminary stage. It extracts the relevant part of a webpage allowing us to ignore the rest of content that can become useless, irrelevant, or even worst, noisy. In this work, we have presented a new technique for content extraction uses the DOM structure of the webpage to identify the blocks that groups those nodes with a higher proportion of text.

The DOM structure not only allows us to improve the detection of blocks, but it also allows us to discard those parts of the webpage that have a large amount of textual information but belong to other HTML containers. Our implementation and experiments have shown the usefulness of the presented technique.

The technique could be used not only for content extraction, but also for blocks detection. It could detect all blocks in a webpage by applying the presented algorithms iteratively to detect one block after the other. In this way, we could detect the most relevant block; then, remove from the DOM tree all its nodes, and detect the next relevant block in the remaining DOM tree. This process would identify all blocks in relevant order. Another interesting open line of research is using the technique to detect the menus of a webpage. A preliminary study showed that instead of using a ratio characters/nodes, we could use a ratio hyperlinks/nodes to discover big concentrations of links in the DOM tree. If we

collect those concentrations of links where the links contain less characters, we will find the menus of the webpage.

Acknowledgments

This work was partially supported by the Spanish *Ministerio de Ciencia e Innovación* under the grant TIN2008-06622-C03-02 and by the *Generalitat Valenciana* under the grant PROMETEO/2011/052. David Insa was partially supported by the *Ministerio de Educación* under grant FPU AP2010-4415.

References

[1] D. Gibson, K. Punera, and A. Tomkins, "The volume and evolution of web page templates," in *Proceedings of the 14th International Conference on World Wide Web (WWW'05)*, Chiba, Japan, 2005, pp. 830–839.

[2] T. Gottron, "Content code blurring: A new approach to content extraction," in *Proceedings of the 5th International Workshop on Text-Based Information Retrieval (TIR'08)*, Turin, Italy, 2008, pp. 29–33.

[3] T. Weninger, W. Hsu, and J. Han, "CETR — content extraction via tag ratios," in *Proceedings of the 19th International Conference on World Wide Web (WWW'10)*, North Carolina, USA, 2010, pp. 971–980.

[4] X. Li and B. Liu, "Learning to classify text using positive and unlabeled data," in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'03)*, Acapulco, Mexico, 2003.

[5] J. Arias, K. Deschacht, and M. Moens, "Language independent content extraction from web pages," in *Proceedings of the 9th Dutch-Belgian Information Retrieval Workshop (DIR'09)*, The Netherlands, 2009, pp. 50–55.

[6] B. Krüpl, M. Herzog, and W. Gatterbauer, "Using visual cues for extraction of tabular data from arbitrary HTML documents," in *Proceedings of the 14th International Conference on World Wide Web (WWW'05)*, Chiba, Japan, 2005.

[7] F. Finn, N. Kushmerick, and B. Smyth, "Fact or fiction: Content classification for digital libraries," in *Proceedings of DELOS-NSF Workshop on Personalisation and Recommender Systems in Digital Libraries*, Dublin, 2001.

[8] S. Gupta, G. Kaiser, D. Neistadt, and P. Grimm, "DOM-based content extraction of HTML documents," in *Proceedings of the 12th International Conference on World Wide Web (WWW'03)*, North Budapest, Hungary, 2003, pp. 207–214.

[9] B. Dalvi, W. W. Cohen, and J. Callan, "Websets: Extracting sets of entities from the web using unsupervised information extraction," Carnegie Mellon School of Computer Science, Tech. Rep., 2011.

[10] N. Kushmerick, D. S. Weld, and R. Doorenbos, "Wrapper induction for information extraction," in *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI'97)*, 1997.

[11] W. W. Cohen, M. Hurst, and L. S. Jensen, "A flexible learning system for wrapping tables and lists in HTML documents," in *Proceedings of the international World Wide Web conference (WWW'02)*, 2002, pp. 232–241.

[12] C. Kohlschütter and W. Nejdl, "A densitometric approach to web page segmentation," in *Proceeding of the 17th ACM conference on Information and knowledge management (CIKM '08)*. New York, NY, USA: ACM, 2008, pp. 1173–1182.

[13] C. Kohlschütter, "A densitometric analysis of web template content," in *Proceedings of the 18th international World Wide Web conference (WWW'09)*. New York, NY, USA: ACM, 2009, pp. 1165–1166.

[14] S. Baluja, "Browsing on small screens: Recasting web-page segmentation into an efficient machine learning framework," in *Proceedings of the 15th International Conference on World Wide Web (WWW'06)*. New York, NY, USA: ACM, 2006, pp. 33–42.

[15] J. Gibson, B. Wellner, and S. Lubar, "Adaptive web-page content identification," in *Proceedings of the 9th annual ACM international workshop on Web information and data management (WIDM '07)*. New York, NY, USA: ACM, 2007, pp. 105–112.

[16] C. Kohlschütter, P. Fankhauser, and W. Nejdl, "Boilerplate detection using shallow text features," in *Proceedings of the third ACM international conference on Web search and data mining (WSDM '10)*. New York, NY, USA: ACM, 2010, pp. 441–450.

[17] W3C Consortium, "Document Object Model (DOM)." [Online]. Available: www.w3.org/DOM

[18] T. Gottron, "Evaluating content extraction on html documents," in *Proceedings of the 2nd International Conference on Internet Technologies and Applications (ITA'07)*. Wrexham, North Wales: 2007, 2007, pp. 123–132.

A Flexible Table Parsing Approach

Frank Schilder, Ravi Kondadadi, and Yana Kadiyska

***Abstract*—Relational data is often encoded in tables. Tables are easy to read by humans, but difficult to interpret automatically. In cases where table layout cues are not obtainable (missing HTML tags) or where columns are distorted (by copying from a spreadsheet to text) previous table extraction approaches run into problems. This paper introduces a novel table parsing approach. Our approach is based on a set of simple assumptions: (a) every table can be split up in data cells and headers, and (b) every table can be parsed beginning from a data cell utilizing the overall table structure. The table parsing is defined as "table flattening" in this paper. That is, the parsing starts with a data cell and pulls out all token (i.e., headers and sub-headers) associated with a respective data cell. We propose a parsing technique that uses two simple parsing heuristics: table headers are to the left of and above a data cell. We experimented with trader emails that contained instrument information with bid-ask prices as data cells. We developed a clustering and classifying method for finding prices reliably in the data set we used. This method is transferable to other data cell types and can be applied to other table content.**

***Index Terms*—Information retrieval, document processing, tables.**

I. Introduction

THIS paper proposes a flexible table parsing approach that can be applied to a majority of table types. In our work we applied the approach to financial tables containing information on various instruments (e.g., bonds, loans). Extracting information from tables has been shown to be notoriously difficult in particular when they occur in plain text (e.g., email) where columns can be easily misaligned after copying them from another document such as a spread sheet.

The task we define in this paper is called *table token sequencing* and requires to (a) find all anchor cells (i.e., bid/ask prices) in the table and (b) determine the tokens in the table that are associated with the respective bid/ask price. The anchor cells are domain-specific and in our case we need to make sure bid-ask prices are found and extracted reliably. We also propose a clustering and classifying approach for finding bid-ask prices in this paper.

Except for the anchor of each sequence given by the bid/ask price, we are mostly agnostic with respect to the other token types. We assume general types such as numbers, capitalized words and a small set of closed class of keywords (e.g. ratings such as AAA).

Manuscript received on October 31, 2011, accepted for publication on December 9, 2011.

Frank Schilder and Ravi Kondadadi are with Thomson Reuters Corporate R&D and Yana Kadiyska is with Thomson Reuters Fixed Income, USA (e-mail:{frank.schilder, ravikumar.kondadadi, yana.kadiyska} @thomsonreuters.com).

It is important to note that this table parsing approach that results in extracting table token sequences can be applied also to other types of tables. Other tables may have different anchors and token types, but the parsing strategy will be the same. Drug dosage tables, for example, contain information about dosages depending on age or weight of the patient.

Table I, for example, can be parsed in the following way: (a) determine the anchors in this table (i.e., numbers), (b) iterate over all the cells in a row that contain anchor cells (e.g. *0.8, 1.2, 1.6*) and collect cells to the left that are not anchors, (c) go up the same column for a respective anchor cell and associate the header cell with the anchor, and (d) go to the right and collect all remaining cells not yet collected. if we start, for example with *1.2* in the second row and the third column, we collect *Infant Drops 7.5mg./0.8 ml, 27-35, ml* as table token sequence. The remaining tokens *Child's weight (pounds)* and *lbs* are type cells describing the type (or unit) and need to be linked to the respective column and row. For this table they both apply to the top row. This linking requires domain knowledge and is the final step in the table parsing process.

For our experiments, we used tables containing financial information where bid/ask prices are the anchor cells. The tables were often copied from spreadsheets into the body of an email and the the formatting was not always maintained The following figure 1, for example, contains a specific format for the bid/ask price, it contains bonds consisting of security name, coupon, maturity date and is slightly misaligned with the headings. The first instrument in this table is specified as HXN- 9.5-14 40.000-41.000:

Given the specified task, the parser needs to extract all instruments specified by the respective token sequence in the table. The task can also be described as "table flattening" because the two dimensional structure of the table needs to be understood in order to end up with a sequence of tokens and a price.

We divide all tokens in a table into two categories. A token is either a data cell or a authority cell (or authority header). Data cells are prices and authority cells/headers are everything else (e.g. coupon, maturity date, contract term). The distinguishing factor between a data cell and an authority cell/header is based on the question of whether the cell can have scope over another cell. The following table, for example, contains contract terms that take scope over the row of prices. Here, the contract terms (i.e. 3yr and 4yr) are authority headers because their scope is vertical. Authority cells, on the other hand, carry horizontal scope, such as the company names *ABC* and *EDF* in the following example:

TABLE I
A MEDICAL TABLE CONTAINING DRUG DOSAGE INFORMATION

Child's weight (pounds)	18-26	27-35	36-53	54-71	72-139	140+	lbs
Infant Drops 7.5 mg./0.8 ml	0.8	1.2	1.6	–	–	–	ml
Liquid 15mg/5ml (tsp)	1/2	3/4	1	1 1/2	2	–	tsp
Chewable 15 mg.	–	–	1	1 1/2	2	4	tablets
Tablets 30 mg.	–	–	–	–	1	2	tablets
Tablet 60 mg.	–	–	–	–	–	1	tablet

```
Bid                 Ask     Bid                              Ask      Bid
Security              Px       Px    YTW    YTW ZSPR ZSPR Notes  Notes2
-------------------------------------------------------------------------
HXN  9.5 14   40.000-41.000   34.994/34.198 3244/3164
HXN  0    14   34.000-35.000   30.389/29.639 2938/2863
HXN  7.5   23   24.000-25.000   34.130/32.910 3132/3007
MOMENT 9.7  14   34.000-35.000   40.260/39.262 3773/3672 B 4MM
```

Fig. 1. A misaligned table containing financial information

```
3yr 4yr
ABC 23/24 25/27
EDF 26/28 30/32
```

The four instruments extracted from this simple table are ABC-3yr-23/24, ABC-4yr-25/27, EDF-3yr-26/28, and EDF-4yr-30/32. Note that the authority headers may not always be perfectly align with the columns, as in this example, but our approach is still able to extract the correct token sequences.

The focus of our work is to provide a table parsing approach that derives all single data cells and their associated authority cells/headers. The output is token sequences of cells and headers. Since our table parsing approach is mainly domain agnostic, further processing can be carried out on these sequences and can factor in further domain-specific reasoning.

The main contributions of this paper:

1) A new approach to table parsing that relies on the distinction between table data cells and headers and is capable of detecting even misaligned table structures.
2) Learning data cells (here: prices) on the fly by clustering and learning a classifier for a specific table.
3) Providing a confidence metric for the table parsing

II. RELATED WORK

Many researchers have studied the problem of extracting information from tables. [1] provides a good overview of the field. Table extraction tasks can be divided into different categories. First, there is the table segmentation task that requires to correctly determine type of table rows, identify columns and data cells. Second, there is the table interpretation task that is more complicated and is an information extraction task that results in correctly filled templates.

Earlier work focused on tables scanned in from documents where ASCII text was produced with the help of OCR programs [2], [3]. In the past, heuristics were employed to solve the table segmentation task, i.e., determining which word belong to which column, and similarly for rows. Our approach mostly avoids the table segmentation task and only requires the identification of the data cells. Because of restricting our approach to a simpler task, early errors in the processing pipeline are avoided. We need to make sure that the extraction of data cell has high accuracy though.

More recent work presented in [4], for example, addressed the task of classifying label rows and data rows in ASCII documents using a machine learning approach (i.e., Conditional Random Fields (CRF)). One of the difficulties in a supervised learning approach is how to obtain accurately and reliably annotated data. Work from MITRE [5] specifically created a platform to streamline such annotation effort. Still, a lot of effort needs to go into the annotation of tables in order to cover a large enough representative sample of the entire data set. In addition, a disadvantage of a general supervised machine learning approach is that it will optimize on the most-frequent table structures and treat infrequent tables that are still well-formed as noise. In contrast to a supervised ML approach, we rely on clustering cells in each individual table and train a classifier for each table on the fly and hence do not require any annotated training data. This classifier identifies cells based on the table context and avoids the pitfalls producing a model based on the most frequent table structures.

Because a lot of useful information are represented in tables in HTML documents, there have been a lot of effort in extracting information from Web documents, known as *wrappers*. Many efforts are based on manual created rules. More advanced approaches used machine learning techniques [6]. The major problem with wrapper-based approach for information extraction is that those wrappers are not robust across different sites and slight modifications to existing pages might invalidate working wrappers.

[7] explored an approach that used the rendering information from a web browser. The approach is not limited to HTML table tag, but requires some sort of table representation in order to produce the spatial information it derives the information from. Because of

domain independence, the accuracy reaches only about 50% for the table interpretation task, but higher results for the table segmentation task (precision:68, recall: 81). This approach fails if no table structure representation is supplied and will run into problems if misaligned table columns are part of the table.

III. Table parsing

We propose a new technique to table parsing where the goal is to determine the data cells and the headings that refer to the respective data cell. Our approach is motivated by the observation that every table has a set of data cells of the same types and that headings referring to one cell are mostly to the left and above this cell. Most importantly, we do not assume a perfectly aligned table structure. Tables where cells in a row have shifted when copied from medium to another (e.g., spreadsheet to email editor) maintain the constraints of having the headings to the left and somewhere above. Since we do not rely on the spatial structure of the tables (e.g., column separation), we are able to parse even very misaligned tables

Our new approach is based on the following heuristics:

1) Detect data cells with high accuracy (here: bid/ask prices).
2) Utilizing table structure by parsing the text left to each data cell and the rows above for detecting header cells (go left and up)
3) Link all remaining tokens to the right of a data cell to that data cell (these are often notes or secondary information)
4) Collect all tokens associated with each cell and therefore flatten the table

A. Finding header and data cells

In our chosen domain, data cells are bid/ask prices, header cells can be percent, maturity dates, company names or specific identifiers (i.e., facilities). As a first step, tokenization ensures that the basic tokens used in a table are identified. We developed a specific ANTLR grammar that distinguish between basic types such numbers, words, special keywords.[1] Another ANTLR grammar identified prices and computed their bid and ask price values.

1) Parsing tokens and prices: We developed a tokenizer grammar that detects simple token types that are frequent in trader emails. For example, Words starting with an upper-case letter are identified as follows:

```
UPPER :
 (INT? 'A'..'Z' ('A'..'Z'|'a'..'z'|INT)*);
```

All number tokens are translated into decimals since we use this number values for various feature when classifying them as prices or coupons.

[1]ANTLR is a parser generator based on LL^* parsing technology. ANTLR grammars are written in EBNF and can be translated into many different programming languages including Java code.

Prices are often easily identifiable as `NUMBER SEP NUMBER`, where `SEP` can be `-` or `/`, as in `23-26`. However, there are also cases where there are no separators or where the price actually describes a different token sequences such as a date or a version number, as in `3/10` or `07-10`. In order to increase recall and weed out false positives, we developed a "cluster and classify" approach described in the following section.

2) Clustering and classifying prices: Another approach to detecting prices was based on an approach that used clustering and self-training on the derived clusters. The clustering was based on the following observation. The Ask price is always higher than the Bid price. Hence the ratio of the Bid price divided by the Ask price has be greater than 1. Moreover, the difference between the two prices is never very large. As a rule of thumb we assume that the Ask price is never more than twice as big as the Bid price.

1) Compute the ratio between two consecutive numbers N1 and N2: BidAskRatio = N2/N1
2) Assign the following clusters:
   a) If $1 <$ BidAskRatio < 2, then N1= -1 and N2 = 1
   b) Else: N1,N2=0

Note that the assignment is based on consecutive number tuples in an email. It is therefore possible that number assignments are overwritten because two tuples fulfill the cluster assignment as in Ford 8 10 13 14. The tuples 8-10, 10-13, as well as 13-14 fulfill the condition for assigning -1 and 1 for the first and the second number, respectively. Given the assignment procedure defined above, the numbers are assigned to the following clusters: 8(-1), 10(-1), 13(-1), 14(1). Even though the numbers 8 and 10 which are coupon and maturity date in this made-up example are wrongly assigned as bid prices, the overall structure of the table will be used to weed them out of the pool of bid/ask prices.

Given the assignment of a cluster to each number in an email, a feature vector can be created for each number. These vectors are then used to generate a multi-class classifier via an SVM. We used the WEKA package for training this classifier and also normalized the data set [8]. That means that the actual training set contains approximately the same number of 0, -1, and 1 instances.

After training, the classifier is run over the training set again. Using the following feature set, the classifier is able to detect the prices that are within the most frequent context for this given table:

- `current_tok` current token type (e.g., UPPER)
- `prev_tok` previous token type
- `next_tok` next token type
- `prev_prev_tok` type of the token previous to the previous token
- `next_next_tok` type of the next token to the next token
- `pos` n^{th} token in a line
- `char_pos` character position of the middle of the token

Note that the training and testing on the same email has certain advantages over training over an annotated data set of emails used for supervised learning. If we train over such a large data set, as we did in a preliminary study, feature weights are derived via the frequency in the overall data set. The peculiarities of a certain table are not taken into account, even though for a given email the position of the price and the context may be perfectly understandable following a pattern not often seen in the training data.

After determining the bid/ask prices and tokenizing and labeling the tokens according to some general types, the token sequences can be derived via the parsing strategy of *going left and up* described in the following.

B. Go left

Most authority cells and headers are to the left of a data cell in tables. If we start with the first data cell (i.e., bid/ask price), we go to the left of this token and collect non-price tokens as associated to this particular data cell. A new token can be added to the sequence only if it had not been associated to another cell.

This parsing strategy can also be described by head-driven parsing approach such as Lexical Functional Grammar (LFG) [9] or Head-Driven Phrase Structure Grammar (HPSG) [10]. Both linguistic approaches combine syntactic and other constraints in one grammar formalism. In LFG, the c-structure is derived by syntactic grammar rules. In addition, semantic information — the so-called s-structure — can further constrain the c-structure.

For the table parsing, we assume a simple context-free grammar that parses tokens per line

```
LINE --> INSTRUMENT* NL
INSTRUMENT --> TOKEN* PRICE
```

Since the goal of the table parsing is the derivation of token sequences per price, another structure is derived via the parsing. This table signature structure (ts-structure) is added as further constraint onto the c-structure.

```
INSTRUMENT([Price(Value)]) --> PRICE
INSTRUMENT([TokenType(Value) | Rest]) -->
 TOKEN INSTRUMENT(REST)
```

After parsing the first instrument, a so-called signature is derived, that constrains the subsequent instruments and completes them, if necessary. Figure 2 shows the derivation for the line *ABC 23/24 25/27*. The first INSTRUMENT is derived up to the first PRICE. The signature derived is the list $[UPPER(ABC)]$. This signature is then used to constrain the subsequent instruments in this line. In this case, the company name *ABC* is added to the next price.

The constraints invoked on the instruments in the line can be of different types. If the signature of the next instrument is a sub-sequence, further authority cells are added, as in the example line. If the signatures do not match, on the other hand, the information from the previous instrument is not carried over.

C. Go right

Occasionally, there are unassigned cells to the right of the anchor cells. In a last step, these cells are collected and added to the right-most anchor cell. In Figure 1, the spread information and additional notes are added to the respective anchor cell (e.g., *40.260/39.262 3773/3672 B 4MM* are added to *MOMENT 9.7 14 34.000-35.000*)

After all tokens have been parsed and the authority cells have been applied to all instruments, we can go up the table and associate other sequences or data headers with a token sequence.

D. Go up

There are two cases for finding further tokens for a token sequence above the line where the data cell is in. First, it is possible to find a longer token sequence in an earlier line. For example, the name of the instrument is only mentioned in the first line including coupon and maturity date. All subsequent lines omit the name, but have different coupons and maturity dates.

```
Ford  3.4 12 13-14
      4.5 13 20-21
```

In order to add further authority cells to sequences, we defined a signature for each sequence. The first line in the above example has the following signature: $[UPPER, NUMBER, NUMBER]$. The second line, on the other hand, has the signature $[NUMBER, NUMBER]$. If we find another sequence above that has a super-signature, the missing tokens (here: Ford) are added to the instrument below.

Second, there are authority headers in header lines. Header lines are defined as lines without any prices. Those authority headers can be of two types: (a) the tokens can be token types such as *coupon* or (b) the authority headers are additional tokens that are added to the token sequence (e.g., *5yr* indicating the contract term).

In order to match these authority headers to the token sequences, we utilize the Hungarian Algorithm [11]. This combinatorial optimization method is used for assignment problems. For our purposes, we need to define a $n \times n$ matrix consisting of the instruments in a line and the authority cells in a line above. For the Hungarian method, the instruments in one line and the authority cells in a line above are represented in form of a a complete bipartite graph $G = (S, T; E)$ where $s \in S$ instruments need to be matched to $t \in T$ authority cells in the line above. For each link between all vertices, a non-negative cost $c(s, t)$ is defined. We define the cost by the distance between the beginning of the price information of the instrument and the authority header. The cost matrix for the following example table is transformed by subtracting the

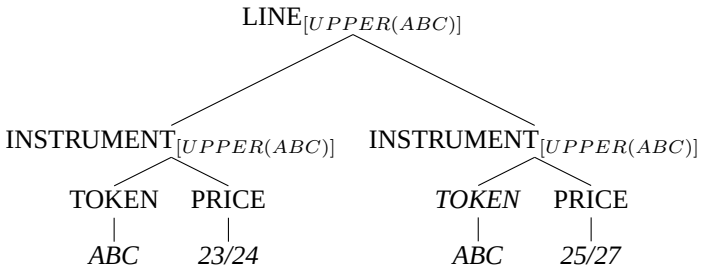


Fig. 2. A simple table parse tree

lowest number in each row and column from all the cells in the same row and column. Then the matching is indicated by the 0 in each row and column for this simple example.[2]

```
 3yr 4yr
ABC 23/24 25/27
EDF 26/28 30/32
```

Cost metric:

$$\begin{pmatrix} 5 & 1 \\ 14 & 10 \end{pmatrix}$$

1) Reduce the rows by subtracting the minimum value of each row from that row.
2) Reduce the columns by subtracting the minimum value of each column from that column.
3) Select a matching by choosing a set of zeros so that each row or column has only one selected:

$$\begin{pmatrix} 0 & 4 \\ 4 & 0 \end{pmatrix}$$

IV. Confidence metric

As our approach is mostly used as a part of a table extraction workflow, it is very important that we provide a confidence score to identify tables we can derive information from with high confidence. This main workflow can thus avoid processing complex tables. The Confidence measure is defined as a number between 0 and 1 indicating how confident we are about the correctness of the output instruments. Given a confidence metric we are able to identify a subset of the test set where we can extract instrument token sequences with high accuracy (i.e. $>$ 90 precision/recall).

A. Feature set

The most important factor in the confidence estimation is the complexity of the input itself. For example, our system performs better with tables that have only one level of sub-headers compared to nested tables. Features in this category include the number of blank lines in the table or the number of lines where the number of token signatures is not same as the one from line above.

[2]There are more steps to consider for a more complex example

The other set of features in our confidence metric is related to the system's performance on the input. Features in this category include the number of prices that are also dates or the ratio of numbers to prices in the email.

B. Gaining confidence

We developed a binary classifier to classify between complex and simple input texts according to our system. This will let the user of our output decide whether to use it not. We had a training set of 202 emails where the instruments were identified. We used a threshold on the F-score of our system on those emails to create training data for our classifier. We experimented with different thresholds on the F-score and found that 0.7 works the best. With this threshold, we had around 130 positive examples and 72 negative examples for the classifier. We used an SVM to build the classification model. We tested the model on the same data using 10-fold cross validation. The accuracy of the classifier was 85.8. The precision and recall of the positive class were 89.8 and 91 respectively.

V. Evaluation

The evaluation we carried out had two phases. First, we needed to show that the table flattening achieves high precision and recall. Secondly, we used the table flattening in the table extraction workflow that produces instrument templates used to populate a data base of instruments and their bid/ask prices.

A. Data

We obtained 202 emails of various length and annotated the token sequences for each price found in the emails. In addition, we annotated the tokens by token type, as in the following example:

```
1238023.txt GS  4.5 06/10 185/175
1238023.txt NAME CPN MAT PRICE
```

The token sequence task requires to check whether these tokens from the first line are part of the tokens derived from the table parsing process. The instrument slot filing part also checks for the correct slots to be filled. In this example, the

TABLE II
RESULTS FOR (A) TABLE SEQUENCE TASK (B) TABLE SEQUENCE TASK WITH CONFIDENCE METRIC

	Precision	Recall	F-Value
micro-averaged			
Table Parsing	0.84	0.85	0.85
Baseline	0.34	0.36	0.32
macro-averaged			
Table Parsing	0.81	0.81	0.81
Baseline	0.39	0.37	0.38
	Precision	Recall	F-Value
most confident top 60%			
Table Parsing	0.88	0.85	0.86
Baseline	0.34	0.34	0.34
most confident top 75%			
Table Parsing	0.90	0.89	0.89
Baseline	0.33	0.33	0.33

instrument name, the coupon, maturity date and the price have to be placed into the respective slot.

In addition to the original emails, we also generated randomly distorted tables from the gold data token sequences in order to show the robustness of our approach. Section

B. Results

We carried out three evaluations. The first one was concerned with matching the token sequences via the table flattening resulting from our table parsing technique. The second evaluation investigated whether the confidence score could improve the overall precision and recall values for a subset of the emails from the test set where we computed high confidence. Finally, we tried to incorporate the table flattening into the email parsing workflow and derive templates for instruments. this last task is the most challenging task, but we can show that using the table flattening that the performance improves overall.

The table flattening module pulls out token sequences that were compared against a token sequence gold data file. A parsed token sequence was correct if it covered the same token (and possibly more tokens) from the gold data. Table II (a) summarizes the results from our experiments comparing the Table Parsing approach with a Baseline approach. The baseline uses the same price tagging and only takes the tokens to the left of a price.

In order to boost performance we utilized the confidence metric and excluded a certain percentage of the test set based on the confidence score. Table II (b) indicates two different cup-off levels where the overall performance significantly improved when only a certain percentage of emails were tested.

In a final experiment, we automatically generated tables of varying complexity from a quote database. Each record in the database corresponds to a quote sequence with all the fields associated with that quote. Each record also has an indication of the source of the record. We grouped records by source and generated a table corresponding to each source. The most important field in a quote sequence is the company name. A company can have more than one quotes in the same source. So within each source, we grouped the records by company name. We introduced randomness in the following aspects of the table generation:

	Description
1	No randomization
2	A
3	B
4	C
5	A+B+C

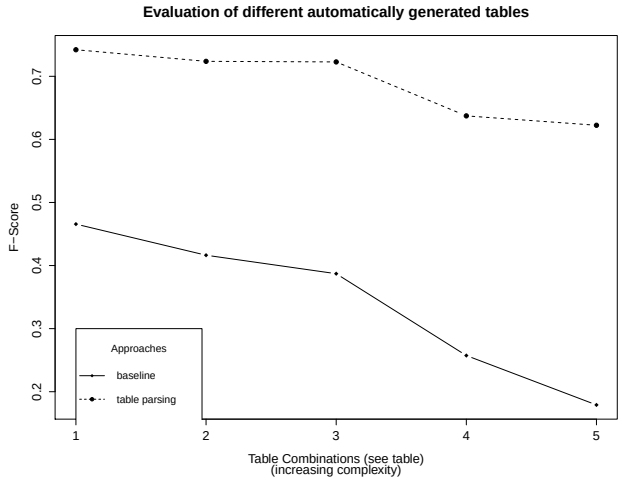


TABLE III
RANDOMLY GENERATED TABLE COMPLEXITY

- A We always generate the company name for the first instrument (e.g., Ford 6 15 30-32), but instruments in subsequent lines that are of the same type do not need to have this authority cell in order to be correctly understood.
- B A line in the table can have more than one sequence
- C The company name can appear to the left of the quote sequence or at the top.

We generated tables of different complexity using different combinations of the above mentioned features.

VI. CONCLUSIONS

We presented a table parsing approach that is based on a heuristics of detecting anchor cells (i.e., bid/ask price) and derives token sequences represented in the table. The proposed approach does not required any training data and is unsupervised regarding the clustering and classification of the so-called anchor cells.

Even though we focus on financial data, the approach can be easily adapted to other domains. Moreover, the approach does not rely on the spatial information encoded in HTML tags and shows a robust performance when tested on tables where noise in form of misalignment is introduced.

ACKNOWLEDGMENT

The authors would like to thank Khalid Al-Kofahi, VP Research, Thomson Reuters and Ted Healey, Global Head of Web Development, Thomson Reuters for supporting this work. We also would like to thank the anonymous reviewers for their valuable suggestions.

REFERENCES

[1] R. Zanibbi, D. Blostein, and J. Cordy, "A survey of table recognition: Models, observations, transformations, and inferences," *Int'l J. Document Analysis and Recognition*, vol. 7, no. 1, 2004.

[2] M. Hurst and S. Douglas, "Layout and language: Preliminary investigations in recognizing the structure of tables," in *Proc. of Int'l Conf. of Document Analysis and Recognition*, 1997.

[3] P. Pyreddy and W. B. Croft, "Tintin: A system for retrieval in text tables," in *Proc. of Int'l Conf. of Digital Libraries*, 1997.

[4] D. Pinto, A. McCallun, X. Wei, and B. Croft, "Table extraction using conditional random fields," in *Proc. of SIGIR*, Toronto, 2003.

[5] M. Vilain, J. Gibson, B. Wellner, and R. Quimby, "Table classification: An application of machine learning to web-hosted financial documents," MITRE, Technical Report, 2006.

[6] W. Cohen, M. Hurst, and L. Jensen, "A flexible learning system for wrapping tables and lists in HTML documents," in *Proc. of WWW*, 2002.

[7] W. Gatterbauer, P. Bohunsky, M. Herzog, B. Krüpl, and B. Pollak, "Towards domain-independent information extraction from web tables," in *Proceedings of the 16th International World Wide Web Conference (WWW 2007)*. ACM Press, May 8–12, 2007, pp. 71–80. [Online]. Available: http://www2007.org/paper790.php

[8] J. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods - Support Vector Learning*, B. Schoelkopf, C. Burges, and A. Smola, Eds. MIT Press, 1998. [Online]. Available: http://research.microsoft.com/\~jplatt/smo.html

[9] R. M. Kaplan and J. Bresnan, "Lexical-functional grammar: A formal system for grammatical representation," in *The Mental Representation of Grammatical Relations*, J. Bresnan, Ed. Cambridge, MA: MIT Press, 1982, pp. 173–281.

[10] C. Pollard and I. A. Sag, *Head-Driven Phrase Structure Grammar*. Chicago: University of Chicago Press, 1994.

[11] H. W. Kuhn, "The Hungarian Method for the Assignment Problem," in *50 Years of Integer Programming 1958-2008*, M. Jünger, T. M. Liebling, D. Naddef, G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi, and L. A. Wolsey, Eds. Springer Berlin Heidelberg, 2010, pp. 29–47. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-68279-0_2

String Distances for Near-duplicate Detection

Iulia Dănăilă, Liviu P. Dinu, Vlad Niculae, and Octavia-Maria Șulea

***Abstract*—Near-duplicate detection is important when dealing with large, noisy databases in data mining tasks. In this paper, we present the results of applying the Rank distance and the Smith-Waterman distance, along with more popular string similarity measures such as the Levenshtein distance, together with a disjoint set data structure, for the problem of near-duplicate detection.**

***Index Terms*—Near-duplicate detection, string similarity measures, database, data mining.**

I. Introduction

A. Motivation

THE concept of *near-duplicates* belongs to the larger class of problems known as *knowledge discovery* and *data mining*, that is identifying consistent patterns in large scale data bases of any nature. Any two chunks of text that have possibly different syntactic structure, but identical or very similar semantics, are said to be near duplicates. During the last decade, largely due to low cost storage capacity, the volume of stored data increased at amassing rates; thus, the size of useful and available datasets for almost any task has become very large, prompting the need of scalable methods. Many datasets are noisy, in the very specific sense of having redundant data in the form of identical or nearly identical entries. In an interview for The Metropolitan Corporate Counsel (see http://www.metrocorpcounsel.com/articles/7757/near-duplicates-elephant-document-review-room), Warwick Sharp, vice-president of Equivio Ltd., a company offering information on retrieval services to law firms with huge legal document databases, noted that 20 to 30 percent of data they work with are actually near-duplicates, and this is after identical duplicate elimination. The most extreme case they handled was made up of 45% near-duplicates. Today it is estimated that around 7% of websites are approximately duplicates of one another, and their number is growing rapidly. On the one hand, near-duplicates have the effect of artificially enlarging the dataset and therefore slowing down any processing; on the other hand, the small variation between them can contain additional information so that, by merging them, we obtain an entry with more information than any of the original near-duplicates on their own. Therefore, the key problems regarding near-duplicates are identification (detection) and aggregation. It is probable that different methods are needed to treat different types of data: for example, small texts, large texts, or images.

Manuscript received on November 15, 2011, accepted for publication on January 6, 2012.

The authors are with the Faculty of Mathematics and Computer Science, University of Bucharest, Romania (e-mail:danailaiulia@yahoo.com, ldinu@fmi.unibuc.ro, vlad@vene.ro, mary.octavia@gmail.com).

Octavia-Maria Șulea is also with the Faculty of Foreign Languages and Literatures, University of Bucharest, Romania.

The work [1] identified the following domains that can benefit from efficient near-duplicate detection and aggregation methods.

- Web mirrors identification
- Clustering for related documents
- Data extraction
- Plagiarism detection
- Spam detection
- Duplicates in domain-specific corpora

These are by no means exhaustive; the problem finds applications in countless fields.

When looking for duplicates in domain-specific corpora, the goal is to identify near-duplicates arising out of revisions, modifications, copying or merger of documents, etc. Example datasets for such an application are TREC benchmarks, Reuters news articles, and Citeseer data (duplicate scientific article citations). See [1, Conrad and Schriber (22)] for a case-study involving legal documents at a law firm. [1, Manber (42)] initiated an investigation into identification of similar files in a file system, with applications in saving disk space. [1, Review (2009)] identifies a few sample situations when we might deem two text documents as being duplicates of each other:

- Files with a few different words - widespread form of near-duplicates
- Files with the same content but different formatting - for instance, the documents might contain the same text, but dissimilar fonts, bold type or italics
- Files with the same content but different file type - for instance, Microsoft Word and PDF versions of the same file.

For short texts such as text messages, [2] indicated the fundamental differences that must be taken into account when doing term weighting, for example. For short messages, larger differences need to be tolerated, and as much semantic information needs to be taken into account. This technique is also relevant for title matching or for comparing short fields from a database. The literature contains various methods, each more suited for specific applications. Depending on the domain and of the specific goals, certain methods are better than others.

A new algorithm could be tailored to a particular task, improving in the measures that have more weight for that particular application, while possibly scoring less from other

points of view: for example, a duplicate detection algorithm for handheld devices is subject to heavy computational and memory limits, so some accuracy needs to be traded. Alternatively, an innovative and general algorithm could improve the state of the art performance in multiple applications, without trading off any resources.

B. State of the art

The state of the art methods in near-duplicate detection cover a broad spectrum of applications and are based sometimes on radically different background techniques. We will first review the web crawling and mining domain and its particular applications. [1] made two research contributions in developing a near-duplicate detection system intended for a multi-billion page repository. Initially, they demonstrated the appropriateness of Charikar's fingerprinting technique [3] for the objective. Locality-sensitive hashing methods have been used in the context of Map-Reduce systems in order to efficiently do approximate nearest neighbour searches in parallel, on big data: this method is taught at Stanford in their class CS246: Mining Massive Data Sets [1]. The major advantage of it is the speed and scalability, while the drawback of this method is the lack of room for tweaking. [4] from Google developed a two-step duplicate identification method that first finds candidates using Charikar's fingerprinting method, followed by refining the query response using similarity measures on the tractable subsets identified by the first step. (US Pat. 8015162). [5] proposed a novel algorithm called I-Match, which they have shown to perform well on multiple datasets, differing in size, document length and degree of duplication. This is step forward, but its drawbacks are that it relies on term frequencies, which can mislead when compared to a ranking-based approach. Secondly, it requires a lexicon, and therefore domain knowledge and language assumptions. For this reason, the system cannot be used out of the box for different problems, but its performance might be better after appropriate tweaking. Another key discussion in duplicate identification is whether to assume the transitivity of the duplicate relation. Granted, this reduces the number of total comparisons needing to be made. Hashing-based detectors use this fact in order to say that objects assigned to the same bucket are duplicates. In practice, however, because we are facing noisy near duplicates, such a procedure can propagate and augment errors.

On the problem of near-duplicate image detection, [6] applied compact data structure to region-based image retrieval using EMD (Earth Mover's Distance) and compared their results positively with previous systems. [7] have applied the neuroscience-inspired Visual Attention Similarity Measure in order to give more weight to regions of interest. A previous, but nonetheless efficient system was given by Chum et al., using locality-sensitive hashing on local descriptors (SIFT), with *tf-idf*-like weighting schemes, which suggests a unified approach based on deep learning, that would work on text and images. An extension of this method is used by Gao and Tang, wherein they initially compare a subset of local features from subsets of two images, followed by crossed near-neighbour searches which should succeed if the images are near-duplicates (US Pat. App 12576236). Furthermore, recent developments in dictionary learning gave way to powerful applications in image classification, denoising, inpainting and object recognition (the Willow team at INRIA [8]). These methods can prove very useful as feature learners for near-duplicate image detection and we intend to leverage them in our system. Andrew Ng and his team at Stanford have successfully applied this kind of unsupervised feature learning and sparse coding, traditionally used in image processing for text processing tasks [9], which encourages the idea that the features for our system can be learned automatically from domain specific data, and thus work efficiently on different types of data.

[1] http://cs246.stanford.edu

C. Our approach

As far as we are aware, there is no research combining deep / unsupervised feature learning with near-duplicate identification and detection. After building a tractable feature-representation of the data, any duplicate detection algorithm needs a notion of similarity. At the moment we sticked with text features, but tried out different metrics. There is a number of metrics used to define similarity [10], around which duplicate detection algorithms are built.

Identification of an adequate metric for determining the similarity of two objects is an intensely studied problem in linguistic and in social sciences. The numerous possible applications (from establishing text paternity, measuring the similarity between languages, text categorization [11]) place this problem in the top of open problems in domains like computational linguistics.

This paper focuses on finding duplicates represented as textual strings. The similarity between two strings is generally measured by Levenshtein (edit) distance or variants. In this paper we use other two distances (Rank distance and Smith-Waterman distance) and compare them. We will introduce them in the following part, along with the union-find disjoint set data structure used to manage the data and optimize the number of comparisons.

Section 3 is dedicated to experimental results, and the final section presents our conclusions and our intended future work.

II. Preliminaries

A. Rank distance

The rank-distance metric was introduced by Dinu in [12] and was successful used in various domains as natural languages similarities, authorship identification, text categorization, bioinformatics, determining user influence [13], etc. To measure rank distance between two strings, we use the following strategy: we scan (from left to right) both

strings and for each letter from the first string we count the number of elements between its position in first string and the position of its first occurrence in the second string. Finally, we sum all these scores and obtain the rank distance. Clearly, the rank distance gives a score zero only to letters which are in the same position in both strings, as Hamming distance does (we recall that Hamming distance is the number of positions where two strings of the same length differ). On the other hand, an important aspect is that the reduced sensitivity of the rank distance w.r. to deletions and insertions is of paramount importance, since it allows us to make use of *ad hoc extensions to arbitrary strings*, such as do not affect its low computational complexity,

When rank distance is restricted to permutations (or full rankings), it is an *ordinal* distance tightly related to the so-called *Spearman's footrule*.

Let us go back to strings. Let us choose a finite alphabet, say $\{A, C, G, T\}$ as relevant for DNA strings, and two strings on that alphabet, which for the moment will be constrained to be a permutation of each other. E.g. take the two strings of length 6, $AACGTT$ and $CTGATA$. To compute rank distance, we proceed as follows: number the occurrences of repeated letters in increasing order to obtain $A_1A_2C_1G_1T_1T_2$ and $C_1T_1G_1A_1T_2A_2$. Now, proceed as follows: in the first sequence A_1 is in position 1, while it is in position 4 in the second sequence, and so the difference is 3; compute the difference in positions for all letters and sum them. In this case the differences are 3, 4, 2, 1, 3, 1 and so the distance is 14. Even if the computation of the rank distance as based directly on its definition may appear to be quadratic, two algorithms which take it back to linear complexity are presented in [14].

Let $u = x_1x_2 \dots x_n$ and $v = y_1y_2 \dots y_m$ be two strings of lengths n and m, respectively. For an element $x_i \in u$ we define its *order* or *rank* by $ord(x_i|u) = i$: we stress that the rank of x_i is its position in the string, counted from the left to the right, *after* indexing, so that for example the second T in the string $CTGATA$ has rank 5.

Note that some (indexed) occurrences appear in both strings, while some other are *unmatched*, i.e. they appear only in one of the two strings. In definition 1 the last two summations refer to these unmatched occurrences. More precisely, the first summation on $x \in u \cap v$ refers to occurrences x which are common to both strings u and v, the second summation on $x \in u \setminus v$ refers to occurrences x which appear in u but not in v, while the third summation on $x \in v \setminus u$ refers to occurrences x which appear in v but not in u.

Definition 1. *The rank distance between two strings u and v is given by:*

$$\Delta(u,v) = \sum_{x \in u \cap v} |ord(x|u) - ord(x|v)| + \sum_{x \in u \setminus v} ord(x|u) + \sum_{x \in v \setminus u} ord(x|v). \quad (1)$$

Example 1. *Let $w_1 = abbab$ and $w_2 = abbbac$ be two strings. Their corresponding indexed strings will be: $\overline{w_1} = a_1b_1b_2a_2b_3$ and $\overline{w_2} = a_1b_1b_2b_3a_2c_1$, respectively. So, $\Delta(w_1, w_2) = \Delta(\overline{w_1}, \overline{w_2}) = 8$*

Remark 1. *The ad hoc nature of the rank distance resides in the last two summations in **(1)**, where one compensates for unmatched letters, i.e. indexed letters which appear only in one of the two strings.*

B. Smith-Waterman Distance

The Smith-Waterman algorithm was introduced in [15], beeing a variation of Needleman-Wunsch algorithm. Since it is a dynamic programming algorithm, it has the desirable property that it is guaranteed to find the optimal local alignment with respect to the scoring system being used (which includes the substitution matrix and the gap-scoring scheme). The main difference to the Needleman-Wunsch algorithm is that negative scoring matrix cells are set to zero, which renders the (thus positively scoring) local alignments visible. Backtracking starts at the highest scoring matrix cell and proceeds until a cell with score zero is encountered, yielding the highest scoring local alignment

For this application, it is not necessary to build string alignement seeing as we are only interested in the final score, so we will exclude this portion for minimizing the execution time. We considered delta = 1 (the cost value for a gap), the matched score = 2 and the unmatched score = -1.

C. Union-Find Algorithm

Under the assumption that the *is-a-duplicate-of* relation is transitive, by building the similarity graph (thresholded according to table I), the problem of near-duplicate detection amounts to finding the connected components of the resulting graph. This way we can avoid unnecessary comparisons between nodes that are already connected, and reduce computations for a memory cost.

The Union-Find structure was proposed for the task of finding and storing connected components in a graph, for the specific task of near-duplicate entry detection, in [16]. This method is based on disjoint sets with a distinguished item in each, called the representative. An implementation of this well-known data structure was used in our experiment.

III. Experimental results

A. Datasets

In this section we will test the near-duplicate text document detection algorithms discussed above on two data bases: one representing a collection of IT products, and the other containing bibliographic entries.

The first database was put together from different online sources [2] to which near duplicates (containing noise in

[2]Data were collected from catalogues such as http://www.cdw.com/, http://www.itproducts.com/ and http://www.streetdirectory.com/.

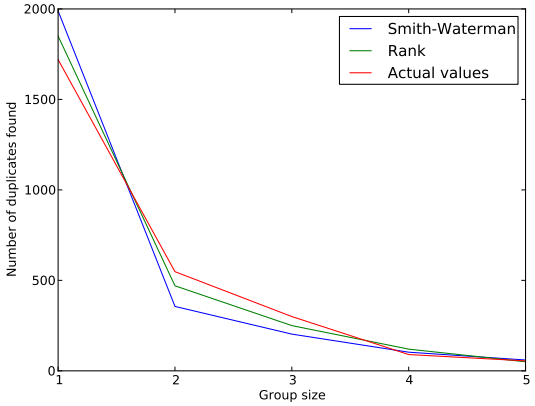


Fig. 1. Results of the first two algorithms on the artificially distorted database, along with the ground truth

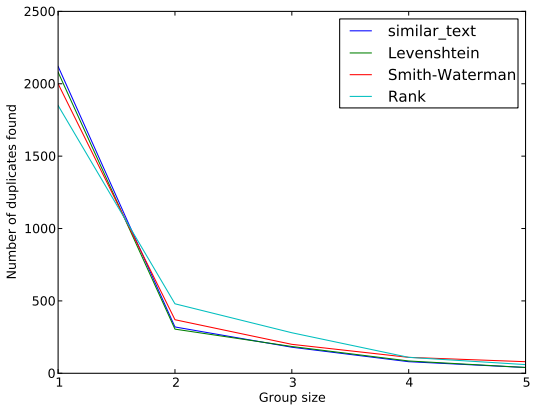


Fig. 2. Results of all similarity algorithms on the artificially distorted database

the form of character insertion, deletion, substitution and transposition) were added.

The second database represents a real-world, undistorted bibliographic collection in BibTeX format, from which we extracted only the title and the author names, in order to lighten the workload given our assumption that the most errors occur in these fields. The source of the data is “A Collection of Computer Science Bibliographies” [17]. Since this collection has over 600,000 entries, we filtered only the ones from the

TABLE I
THE ALGORITHMS USED, AND THE THRESHOLD THAT DEFINES NEAR-MATCHES, WHEN COMPARING STRINGS a AND b, OF LENGTH n_a AND n_b RESPECTIVELY

Metric	Perfect matching	Near matching
Rank distance	$d = 0$	$d \leq \frac{n_a n_b}{2})$
Smith-Waterman	$d = 2\max(n_a, n_b)$	$d \geq \min(n_a, n_b)$
Levenshtein	$d = 0$	$d \leq \frac{\max(n_a, n_b)}{2}$
Similar-text	$p = 100\%$	$p \geq 50\%$

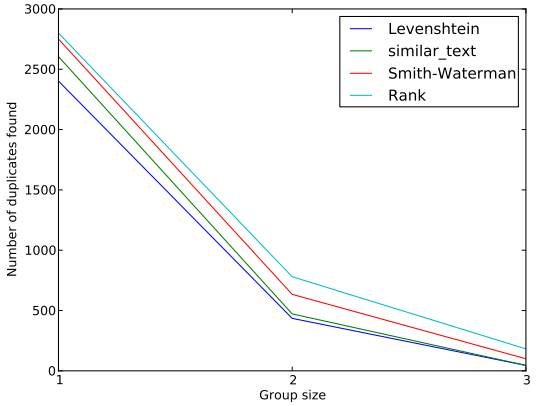


Fig. 3. Results of all similarity algorithms on the bibliography database

TABLE II
DUPLICATES DISTRIBUTION IN ARTIFICIALLY DISTORTED DATA

Group size	Groups number	Input number	Percent
1	1720	1720	62.77%
2	274	548	20.00%
3	96	288	10.51%
4	26	104	3.79%
5	16	80	2.91%
Total	2132	2740	100%

“Planning and Scheduling” category, leaving only 3436 entries such as

{`author: "Andrew G. Barto and S. J. Bradtke and Satinder P.Singh", title: "Learning to Act Using Real Time Dynamic Programming"`}.

A sample duplicate entry of this would be

{`author: "Andrew Barto, J. S. Bradtke and S. P. Singh", title: "Learning to Act Using Realtime Dynamic Programming"`}.

We sought out to investigate the problem of recognizing near duplicates by employing two basic tools: the Union-Find algorithm of grouping data efficiently and the algorithms proposed above. We also looked at the efficiency and the correctness of these algorithms. In what follows we will present the algorithms and the results.

B. Results

The distribution for the duplicates in the artificially distorted database is shown in table II.

The results of the algorithms on the artificial database are displayed in figures 1 and 2 while the results on the real database are shown in figure 3. The figures are distribution plots, the y-value at the position $x = k, k \in \{1, 2, ...\}$ showing the number of documents that can be captured in groups of k. In other words, for $k = 1$ it shows the number of documents

that the algorithm thinks have no duplicates, for $k = 2$ it shows the documents that can be grouped in duplicate pairs, while for $k = 3$ they can be grouped in triples. Note that the points should add up to the total size of the database.

The *similar_text* function used for comparison is the text similarity algorithm from [18], as implemented in the PHP programming language's standard library. It is included as reference because of its accesibility, due to this fact.

In the case of the artificially generated noisy database, we have access to the ground truth. From 1 we can see that the results found by Rank distance are closer to the real distribution of duplicates than the ones found by the Smith-Waterman distance.

For the bibliographic entry database, we assume that the ground truth probability of duplication is lower than in the artificial case. No algorithm found more than 3 duplicate entries for the same information. However under visual inspection, the identified duplicates look correct, confirming the precision of the methods. The Rank distance again seems to have a slower decay rate than the other methods, which can be interpreted as higher recall in the tail of the distribution, assuming a fixed precision.

IV. Conclusions

Our methods for verifying existence of aproximate duplicates exhibit improvement over the previous work in this field. The use of the Union- Find algorithm for grouping the entries significantly reduces the number of comparissons, hightening the efficiency of the general agorithm and its run time. Although it is relyies on the existence of transitivity for the similarity reltion, we have seen that no entries were lost and no errors occured in the grouping of objects.

Until now, the majority of studies on the subject of duplicate detection were based on classic distances, such as Hamming or Levenshtein, yet the results were not always correct. The use of the Smith-Waterman algorithm for strings of characters representing words may seem incertain, taking into consideration that DNA chains are not in the same domain as the one choosen here, yet the results of our experiments show a good performance, an excellent precision and an runtime comparable with classic metrics. Rank distance is usually used for computing distance between ranks, but its adaptation to character strings proved to be fast aqnd precise. We note that there are yet many other metrics and algorithms, which may at first seem unsuitable for a certain problem, but through proper study may prove to be a new solution for a classical problem, possibly even better, faster, and more precise. In our case, the Rank algorithm proved to be more precise than the Smith-Waterman algorithm, being the one closest to the real situation of the duplicates in our datasets.

As future work, we plan to extend these methods in such a way as to minimize the number of comparisons needed, using fingerprinting techniques, as well as to extend them in an unified manner for different data types (images, long text fields, etc.)

V. Acknowledgements

All authors contributed equally to the work presented in this paper. The research of Liviu P. Dinu was supported by the CNCS, IDEI - PCE project 311/2011, "The Structure and Interpretation of the Romanian Nominal Phrase in Discourse Representation Theory: the Determiners."

References

[1] G. S. Manku, A. Jain, and A. Das Sarma, "Detecting near-duplicates for web crawling," in *Proceedings of the 16th international conference on World Wide Web*, ser. WWW '07. New York, NY, USA: ACM, 2007, pp. 141–150. [Online]. Available: http://doi.acm.org/10.1145/1242572.1242592

[2] C. Gong, Y. Huang, X. Cheng, and S. Bai, "Detecting near-duplicates in large-scale short text databases." in *PAKDD'08*, 2008, pp. 877–883.

[3] M. S. Charikar, "Similarity estimation techniques from rounding algorithms," in *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, ser. STOC '02. New York, NY, USA: ACM, 2002, pp. 380–388. [Online]. Available: http://doi.acm.org/10.1145/509907.509965

[4] M. R. Henzinger, "Finding near-duplicate web pages: a large-scale evaluation of algorithms," in *SIGIR*, 2006, pp. 284–291.

[5] A. Chowdhury, O. Frieder, D. Grossman, and M. C. McCabe, "Collection statistics for fast duplicate document detection," *ACM Trans. Inf. Syst.*, vol. 20, pp. 171–191, April 2002. [Online]. Available: http://doi.acm.org/10.1145/506309.506311

[6] Q. Lv, M. Charikar, and K. Li, "Image similarity search with compact data structures," in *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, ser. CIKM '04. New York, NY, USA: ACM, 2004, pp. 208–217. [Online]. Available: http://doi.acm.org/10.1145/1031171.1031213

[7] L. Chen and F. Stentiford, "Comparison of near-duplicate image matching," in *Proceedings of the 3rd European Conference on Visual Media Production*, 2006. [Online]. Available: http://discovery.ucl.ac.uk/41711/

[8] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML '09. New York, NY, USA: ACM, 2009, pp. 689–696. [Online]. Available: http://doi.acm.org/10.1145/1553374.1553463

[9] A. Maas and A. Ng, "A probabilistic model for semantic word vectors," in *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2010.

[10] M.-M. Deza and E. Deza, *Dictionary of Distances*. Elsevier Science, Oct. 2006. [Online]. Available: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20\&path=ASIN/0444520872

[11] L. P. Dinu and A. Rusu, "Rank distance aggregation as a fixed classifier combining rule for text categorization," in *Proceedings of CICLing*, 2010, pp. 638–647.

[12] L. P. Dinu, "On the classification and aggregation of hierarchies with difererent constitutive elements," *Fundamenta Informaticae*, vol. 55, no. 1, pp. 39–50, 2002.

[13] X. Tang and C. Yang, "Identifing influential users in an online healthcare social network," in *Proc. IEEE Int. Conf. on Intelligence and Security Informatics, 2010 (ISI '10)*, May 2010.

[14] L. P. Dinu and A. Sgarro, "A low-complexity distance for dna strings," *Fundamenta Informaticae*, vol. 73, no. 3, pp. 361–372, 2006.

[15] T. F. Smith and M. S. Waterman, "Identification of common molecular subsequences," *Journal of Molecular Biology*, vol. 147, pp. 195—197, 1981.

[16] A. E. Monge and C. P. Elkan, "Efficient domain-independent detection of approximately duplicate database records," *Engineering*, 1997. [Online]. Available: http://www.cecs.csulb.edu/~monge/research/vldb97.pdf

[17] A.-C. Achilles, "A collection of computer science bibliographies," 1996. [Online]. Available: http://liinwww.ira.uka.de/bibliography/index.html/

[18] I. Oliver, *Programming classics - implementing the world's best algorithms*. Prentice Hall, 1994.

Comparing Sanskrit Texts for Critical Editions: The Sequences Move Problem

Nicolas Béchet and Marc Csernel

***Abstract*—A critical edition takes into account various versions of the same text in order to show the differences between two distinct versions, in terms of words that have been missing, changed, omitted or displaced. Traditionally, Sanskrit is written without spaces between words, and the word order can be changed without altering the meaning of a sentence. This paper describes the characteristics which make Sanskrit text comparisons a specific matter. It presents two different methods for comparing Sanskrit texts, which can be used to develop a computer assisted critical edition. The first one method uses the L.C.S., while the second one uses the global alignment algorithm. Comparing them, we see that the second method provides better results, but that neither of these methods can detect when a word or a sentence fragment has been moved. We then present a method based on N-gram that can detect such a movement when it is not too far from its original location. We show how the method behaves on several examples.**

***Index Terms*—Sanskrit, text alignment.**

I. Introduction

A CRITICAL edition is one that takes into account all the different known versions of the same text. If the text is mainly known through a great number of manuscripts that include non trivial differences, the critical edition often looks rather daunting for readers unfamiliar with the subject: the edition is mainly made up by footnotes that highlight the differences between manuscripts, while the main text (that of the edition) is rather short, sometimes just a few lines per page. The differences between the texts are usually described in terms of words (sometimes sentences) that are missing, or have been added or changed in a specific manuscript. This bring up to mind the edit distance but in term of words rather than characters. The text of the edition is established by the editor according to his own knowledge of the text. It can be a particular manuscript or a 'mean' text built according to some specific criteria. Building a critical edition by comparing texts one with another, especially manuscript ones, is a task which is certainly long and, sometimes, tedious. This is why, computer programs have long been helping philologists in their work (see [1] or [2] for example), but most of them are dedicated to texts written in Latin (sometimes Greek) scripts.

In this paper we focus on the problems involved by a critical edition of manuscripts written in Sanskrit. Our approach is illustrated by texts that are extracted from manuscripts of the "Banaras gloss": the *kāśikāvṛtti*.

The Banaras gloss was written around the 7th century A.D., and is one of the most famous commentaries on Pāṇini's grammar, which is known as the first **generative** grammar ever written, and was written around the fifth century B.C. as a set of rules. These rules cannot be understood without the explanations provided by a commentary such as the *kāśikāvṛtti*. This corpus was chosen, because it is one of the largest collection of manuscripts (about hundred different ones) of the same text actually known. Notice that, since some manuscripts have been damaged by mildew, insects, rodents, etc.., they are not all complete.

In what follows we first describe the characteristics of Sanskrit that matter for text comparison algorithms, we will then show that such a comparison requires the use of a lemmatized text as the main text. Using a lemmatized text induces the need of a lexical preprocessing. Once the lexical preprocessing has been carried out, we can proceed to the comparison, where two approaches have been developed, the first one based on the Longest Common Subsequence (L.C.S.) by [3], and the second one on edit distance by [4]. The second method is easier to use, so we use it to align our Sanskrit texts before moving sequences.

Because the word order is not always meaningful in Sanskrit, some manuscripts have some words sequences which are not in the same place than in the main text, and the alignment procedure, whichever it is, is not able to align such sequences together. The misplaced sequence appears to be missing in on place, and added in another one. This is why we use here a word n-gramm based method, to discover if some sequence moves are likely to exist, and then determine their precise limits, where they have been moved, and display them. This improvement is detailed in the section IV of this paper. Remark that the sequence move problem is quite similar to the translocation problem which exists in genomics.

II. How to compare Sanskrit manuscripts

A. Sanskrit and its graphical characteristics

One of the main characteristics of Sanskrit is that it is not linked to a specific script. Here however we provide all our examples using the Devanāgarī script, which is nowadays the most most widely used. The script has a 48-letter alphabet.

Due to the long English presence in India, a tradition of writing Sanskrit with the Latin alphabet (a transliteration) has

Manuscript received on October 20, 2011, accepted for publication on December 9, 2011.

Nicolas Béchet is with GREYC Université de Caen Basse-Normandie, France (e-mail: nicolas.bechet@unicaen.fr)

Marc Csernel is with INRIA Roquencourt, Université Paris Dauphine, France (e-mail: Marc.Csernel@inria.fr)

long been established and used by many European scholars such as Franz Bopp in 1816. All these transliteration schemes were originally carried out to be used with traditional printing. It was adapted for computers by Frans Velthuis [5], more specifically to be used with TEX. According to the Velthuis transliteration scheme, each Sanskrit letter is written using one, two or three Latin characters; notice that all our corpus is written according to the Velthuis scheme and not in Devanāgarī Unicode [6].

In ancient manuscripts, Sanskrit is written without spaces, and this is an important graphical specificity, because it greatly increases the complexity of text comparison algorithms.

On the other hand, each critical edition deals with the notion of word. Since electronic Sanskrit lexicons such as the one built by Huet [7], [8] do not cope with grammatical texts, one must find a way to identify each Sanskrit word within a character string, without the help of either a lexicon or of spaces to separate the words.

The reader interested in exploring deeper approach of the Sanskrit characteristics which matter for a computer comparison can refer to [3].

B. How to proceed?

The solution comes from the lemmatization of one of the two texts of the comparison: the text of the edition. The lemmatized text is prepared **by hand** by the editor. It is called *padapāṭha*, according to a mode of recitation where syllables are separated. ¿From this lemmatized (the *padapāṭha*) text, we will build the text of the edition, which is called *saṃhitapāṭha*, according to a mode of recitation where the text is said continuously. The transformation of the *padapāṭha* into the *saṃhitapāṭha* is not straightforward because of the existence of *sandhi* rules.

What is called *sandhi* — from the Sanskrit: liaison — is a set of phonetic rules which apply to the morpheme junctions inside a word or to the junction of words in a sentence. These rules are perfectly codified in Pāṇini's grammar. Roughly speaking, written Sanskrit reflects (via the *sandhi*) the liaison(s) which are made by a human speaker. A text with separators (such as spaces) between words, can look rather different (the letter string can change greatly) from a text where no separator is found.

An example of *padapāṭha*:

```
vi^u^panna_ruupa_siddhi+v.rtti+iya.m
kaa"sikaa_naama
```

We can see that words are separated by spaces and three different lemmatization signs: +, _ , ^.

The previous *padapāṭha* now becomes the following *saṃhitapāṭha*:

```
vyutpannaruupasiddhirv.rttiriya.mkaa"si
kaanaama
```

where the bold letters represent the letters (and the lemmatization signs) which have been transformed, according to a *sandhi* rule.

We call the typed text, corresponding to each manuscript: *Typed Manuscript (T.M.)*. Each *T.M.* contains the text of a manuscript and some annotation commands. The annotation commands keep trace of all the modifications of the manuscript not explicitly present in the text, such as change of ink color, a hole made by a rodent, etc.. They provide a kind of meta-information. Each manuscript is typed by a scholar.

The processing is done in four steps, but only two of them will be considered in this paper:

- **First step**: A lexical preprocessing. The *padapāṭha* is transformed into a virtual *saṃhitapāṭha* in order to make a comparison with a *T.M.* feasible.
  The transformation consists in removing all the separations between words and then in applying the *sandhi*. This virtual *saṃhitapāṭha* form the text of the edition.
- **Second step:** An alignment of a *T.M.* and the virtual *saṃhitapāṭha*. The aim is to identify, as precisely as possible, the words in the *T.M.*, using the *padapāṭha* as a pattern.
- **Third step:** Once the alignment has been achieved and the words of the *T.M.* have been determined, try to improve the alignments results. Determine which word have been added suppressed changed , or moved.
- **Fourth step:** : Display the results in a comprehensive way for the editor. This step is accomplished using XML.

The comparison is done paragraph by paragraph.

C. Why not use the `diff` algorithm

The very first idea to compare Sanwkrit text was to use `diff` in order to obtain the differences between two Sanskrit sequences.

But the results related in [3] were quite disappointing. The classical `diff` command line provided no useful information at all. The result of the comparison of the two following sequences: `"srii ga.ne"saaya nama.h` and `tasmai "srii_gurave namas` just said that they were different.

This is why [3] started to implement their own L.C.S. based algorithm. Its results appear in the right-hand column of Table I. We can see that they are expressed in term of words.

D. The L.C.S based method

This method was developed by [3], and was the first method used to build critical edition of Sanskrit texts. The L.C.S matrix associated with the previous result can be seen in Figure 1 (p. 3). In this figure the vertical text represents the *saṃhitapāṭha*, the horizontal text is associated with the *T.M.*. The horizontal bold dark lines were provided by the *padapāṭha*, before it was transformed into the *saṃhitapāṭha*.

The rectangles indicate how the correspondences between the *saṃhitapāṭha* and the *T.M.* were done. One corresponds to a missing word (`tasmai`) two correspond to a word present in both strings the words (`s"rii` and `nama.h`), and the last one corresponds to a word with a more ambiguous status, we

TABLE I
EXAMPLE OF RESULTS WITH L.C.S.

```
1c1
< "sriigane"saayanama.h
---
> tasmai"sriiguravenama.h
```

`diff` without space

```
1d0
< tasmai
4c3,5
< gurave
---
> gane
> "
> saaya
```

`ediff` with space

```
Word 1 ‘tasmai’ is :
- Missing
Word 2 ‘"srii’ is :
- Followed by Added word(s) ‘ga.ne"saaya’
Word 3 ‘gurave’ is :
- Missing
```

L.C.S. based results without space

can say either that the word has been replaced or that one word is missing and another word has been added.

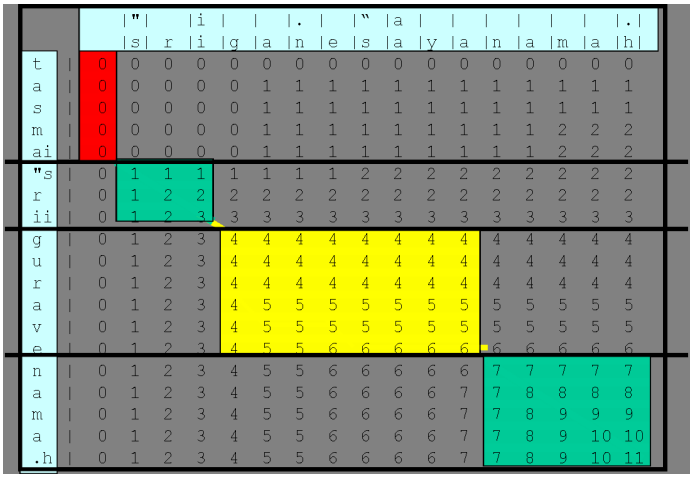


Fig. 1. The L.C.S. Matrix

If the result appears quite obvious within this example, it is not always so easy, particularly when different paths within the matrix can lead to different alignments providing different results. This led the authors to look for some navigation rules which are precisely related in [3].

III. THE ALIGNMENT BASED ON EDIT DISTANCES

This method has been developed by [4] because the L.C.S based method needed too much post processing to be really efficient and because the lack of “substitution” relative to L.C.S method was really a handicap when just one or two letters were changed in a word. The method is still not perfect but is a better base than L.C.S. to provide in the simplest way good alignments, but some improvements still need to be done before this method can be used effectively. Building an alignment using the edit distance matrix, starts by the low right-hand corner, going to the upper left. So the letters of the of the *T.M.* are aligned on the first letter of the *saṃhitapāṭha* with the same value they meet coming from the end of the paragraph. Sometimes this procedure induces the presence of Orphan letters which will be moved to obtain a better alignment as explained in the next subsection .

A. Shifting the orphan letters

We call an orphan letter an isolated a letter belonging to an incomplete word of (usually) a manuscript. To obtain a proper alignment, these letters must fit with the words to which they belong.

Table III gives a good example. The upper line of the table represents the *padapāṭha*, the lowest one a *T.M.*. The word separation induced by the *padapāṭha* are indicated by double vertical lines. Because the *padapāṭha* is used as a template the separations appears also within the *T.M.*. The orphan letters appears in bold. The words `pratyaahaaraa` and `rtha.h` are missing in the *T.M.*. Consequently the letters `a.h` are misplaced, with the word `rtha.h`. The goal is to shift them to the right place with the word `upade"sa.h`. The result after shifting the letters appears in Table II. The bold letters are the letters which has been shifted.

In the second example (Table IV) we see on the left side of the table that the letter `a` must just be shifted from the beginning of `asyddhy` to the end of `saavarny` giving the right-hand part of the table.

Another kind of possible shift is the one linked to the presence of supplementary letters within the *T.M.* such as appears in the left part of Table V. The letters `a` and `nam` of the *padapāṭha* are shifted to the left of the word as appears in the

B. Measuring the quality of the alignment.

It is difficult to find a unique method to measure the quality of the alignments, because each method (L.C.S. and Edit Distance) produces a different type of alignment where the improvement must be made in a different way. L.C.S. methods do not induce any substitution, so measuring of the quality is quite easy to find: more “empty” characters we have, the lower the quality is; even if it can be nuanced in some particular cases. On the other hand with the edit distance based method, the algorithm provides substitutions and we can observe that the more substitutions we have, the lower quality. Remark that all substitutions are not irrelevant, and should not lower the quality, but some of them do, mainly the one corresponding to a possible translocation. Missing or added letters obtained by the edit distance method have always been relevant. Because an important part of the substitution are relevant we will never obtain a null score. We know we have to improve our measure

TABLE II
ORPHAN LETTERS AFTER BEING SHIFTED

v	a	r	n	aa	n	aa	m	u	p	a	d	e	"s	a	.h	p	r	a	t	y	aa	h	aa	r	aa	r	th	a	.h	p	r	a	t	y	aa	h	aa	r	o
v	a	r	n	aa	n	aa	m	u	p	a	d	e	"s	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**a**	**.h**	p	r	a	t	y	aa	h	aa	r	o

TABLE III
AN EXAMPLE OF ORPHAN LETTERS

v	a	r	n	aa	n	aa	m	u	p	a	d	e	"s	a	.h	p	r	a	t	y	aa	h	aa	r	aa	r	th	a	.h	p	r	a	t	y	aa	h	aa	r	o
v	a	r	n	aa	n	aa	m	u	p	a	d	e	"s	**a**	**.h**	-	-	-	-	-	-	-	-	-	-	-	-	-	-	p	r	a	t	y	aa	h	aa	r	o

TABLE IV
ANOTHER EXAMPLE

s	aa	v	a	r	.n	y	a	p	r	a	s	y	d	dh	y
s	aa	v	a	r	.n	y	-	-	-	**a**	s	y	d	dh	y

The orphan letters

s	aa	v	a	r	.n	y	a	p	r	a	s	y	d	dh	y
s	aa	v	a	r	.n	y	**a**	-	-	-	s	y	d	dh	y

After being shifted

TABLE V
SHITFING WITHIN THE *padapāṭha*

p	r	a	y	o	j	-	-	-	-	-	**a**	-	-	-	-	**n**	**a**	**m**
p	r	a	y	o	j	a	n	a	m	s	a	.m	j	"n	aa	n	a	m

Before shifting

p	r	a	y	o	j	**a**	**n**	**a**	**m**	-	-	-	-	-	-	-	-	-
p	r	a	y	o	j	a	n	a	m	s	a	.m	j	"n	aa	n	a	m

After being shifted

TABLE VI
AN ALIGNMENT WITH A LOW SCORE

n	**u**	**b**	-	a	**n**	**dh**	-	a	.h	**l**	**a**	**k**	**aa**	**r**	**e**	**t**	**v**	a	n	u	n	aa	s	i	k	a	.h
r	**e**	**t**	v	a	**k**	**aa**	r	a	.h	**i**	**t**	**s**	**a**	**.m**	**j**	**~n**	**o**	a	n	u	n	aa	s	i	k	a	.h

TABLE VII
THE ALIGNMENT IMPROVED

n	u	b	-	a	n	dh	-	a	.h	**l**	**a**	**k**	**aa**	**r**	**e**	**t**	**v**	-	-	-	-	-	-	-	-	-	-	-	-	-	a	n	u	n	aa	s	i	k	a	.h
-	-	-	-	-	-	-	-	-	-	**l**	**a**	**k**	**aa**	**r**	**e**	**t**	**v**	a	k	aa	r	a	.h	i	t	s	a	.m	j	~n	o	a	n	u	n	aa	s	i	k	a	.h

IV. AN IMPROVEMENT OF THE ALIGNMENT BY AN N-GRAM BASED METHOD.

The goal of this approach is to improve the alignment (obtained by edit distance method) of the *saṃhitapāṭha* and a *T.M.* once the orphan letters have been shifted. An example of an alignment which can be improved is written in Table VI. The bold letters are letters which have been substituted, and lowered the quality.

We define the alignment ***score*** by the number of substituted letters, better the alignment, lower the score. We obtain with this alignment a score of **13**, which is poor.

This score can be improved if we refer to the complete sentence in the *saṃhitapāṭha* and the *T.M.*: we can then move the sequence `la kaare tv` of the *T.M.* by inserting empty letters on the left (*i.e.* the '-' symbol), in the *T.M.* and on the right in the *saṃhitapāṭha* we obtain a good alignment (*with a score of 0*[1]) by aligning `la kaare tv` in a manuscript with `la kaare tv` of the *saṃhitapāṭha* as in the alignment below in Table VII. The bold letters indicate the sequence moved.

A. Overview of the procedure

It is described by the steps below, which are detailed in the next subsections.

1) We first extract word n-gramfrom the *saṃhitapāṭha* (cf. Section IV-B).
2) We search for each word n-gram all the possible better alignments (called a *candidate*) in the *T.M.*
3) For each *candidate*, we modify the original alignment to take the *candidate* into account, and build new

[1]Note that an insertion (denoted '-') is not lowering the score with the edit distance alignment.

TABLE VIII

4-GRAM ALIGNMENT EXAMPLE WITH A SCORE $\mathbf{13}$.

n	u	b	-	**a**	n	dh	-	**a**	**.h**	l	a	k	aa	r	e	t	v
r	e	t	v	**a**	k	aa	r	**a**	**.h**	i	t	s	a	.m	j	˜n	o

TABLE IX

THE WORD 4-GRAM IN THE *saṃhitapāṭha*

u	c	c	aa	r	a	.n	aa	r	th	a	.h	n	-	-	aa	**n**	**u**	**b**	**-**	**a**	**n**	**dh**	**-**	**a**	**.h**	**l**	**a**	**k**	**aa**	**r**	**e**	**t**	**v**

word borders according to the *saṃhitapāṭha* (cf. Section IV-C).

4) We apply optimizations to the new alignment (cf. Section IV-D1).
5) For each new generated alignment, we build a new score and write a proposition of improvement.

B. Extracting the word n-gram

The first step is to extract ***word n-gram***. Word n-gram are frequently employed in the literature, one of the first uses being [9], or more recently in [10]. An *n*-gram of X can be defined as a sequence of n successive X with $\boldsymbol{X} \in \{\boldsymbol{word}, \boldsymbol{letter}\}$. Figure 2 provides an example of letter 2-gram and word 3-gram extraction .

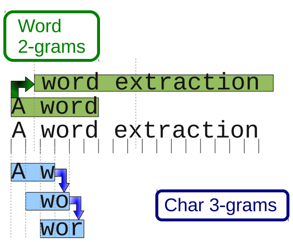


Fig. 2. Example of n-gram extraction.

We use word n-gram extraction in the hope to place all occurrences of a word in a unique context. A single word can have many occurrences in the *saṃhitapāṭha*, but only one when it is preceded and followed by some specific words. The word `kaare` in Table VIII has a lot of instances but only one which is preceded by word `la` and followed by the word `tv`. Thus, by taking a 3-gram, we obtain only one instance of the word 3-gram `la kaare tv` and our extraction becomes relevant.

We use the extraction to make a (word) segmentation of the *saṃhitapāṭha*. An example of 4-gram from the *saṃhitapāṭha* is given in the upper line of Table VIII, the lower line contains letters which do not fit the 4-gram, and that we will try to move, the bold letters are well aligned. In the following when we talk about *word n-gram*, we always refer to the *saṃhitapāṭha*.

The *n* value is set by the user, but could, in the future, be computed by program.

C. The search for candidates

Candidates are defined as strings in a manuscript which could provide a better alignment for a given word n-gram. Thus, *candidates* must contain as many letters than the word n-gram.

The number of shifting is given by the parameter $\boldsymbol{m}$; in the example shown in Figure 3, the parameter $\boldsymbol{m} = \mathbf{4}$: we have shifted the sliding window of $\mathbf{4}$ positions to the left [2]

To estimate the new alignment quality we compute the score, as defined in section IV: the number of substitutions made between the two strings (the one from the *saṃhitapāṭha* is the current word n-gram, the other one from the *T.M.* is a *candidates*. The alignment of strings: `nub-andh-a.h la kaare tv` and `retvakaara.h it sa.mj ˜no` in Table VIII obtain, for instance, a score of $\mathbf{13}$.

If a *candidate* provides a better score that the original one, we keep it. At the end, we keep the best, the on with the lowest score. If several *candidates* have the same score, we keep the last one, but this situation never occurred with the examples we have considered.

To summarize, we first need to compute the score of the original alignment ($\mathbf{13}$). Then, the original word n-gram `nub-andh-a.h la kaare tv` is compared with the 4 *candidates*:

- `aaretvakaara.hit sa.mj˜n`,
- `kaaretvakaara.hitsa.mj`,
- `akaaretvakaara.hitsa.m`,
- `lakaaretvakaara.hitsa`

The word separator ‘ ’ is not reported because word separation is no longer relevant (*i.e.* words are no longer aligned). At the end, with these four *candidates*, none of the scores is better than $\mathbf{13}$, the four *candidates* must be rejected.

D. Integrating the best candidates

After the selection of the best *candidate* for a given word n-gram, we need to build new word boundaries. Let us consider the example in Table VIII with the word n-gram `nub-andh-a.h la kaare tv`. With the parameter m = 20, we find `caara.naartha.hlakaaretv` as the best *candidate*. To align this *candidate* with the word n-gram we need first to adjust the corresponding word boundaries.

[2] An *m* value of 4 means that 8 candidates are tested, 4 from left m-letter l-gram and 4 from right m-letter l-gram

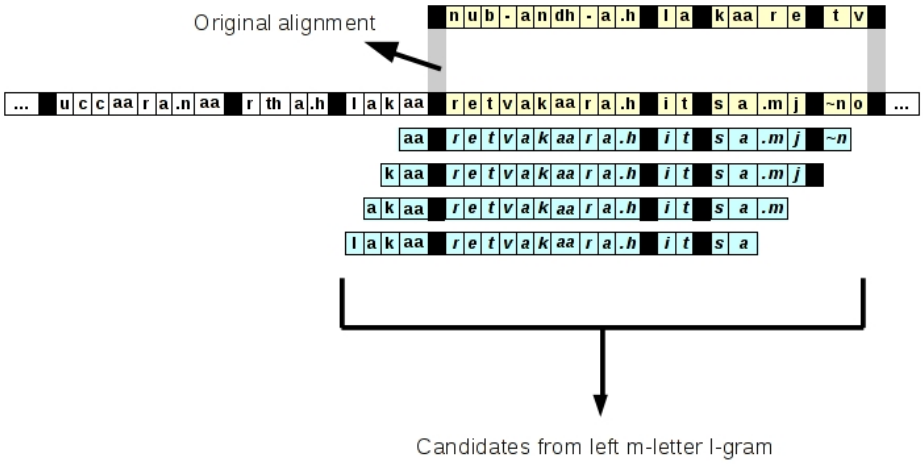


Fig. 3. The search for candidates for the word 4-gram `nub-anh-ah la kare tv`.

1) Aligning the n-gram and the candidate: The n-gram and its corresponding best *candidate* are in bold in Table X.

To align the string `nub-andh-a.h la kaare tv` with the string `caara.naartha.hlakaaretv`, we need to adjust the two texts by adding insertions of empty characters both in the *saṃhitapāṭha* and the *T.M.*

The number of insertions is equal to the absolute value of the substraction of the position of the first letter of the word n-gram and the position of the first letter of the candidate. In our example, the first letter of the word n-gram is **24** and the position of the first letter of the candidate is **10**. Thus, we need to insert **14** letters. We obtain the alignment described in Table XI, in this table the dots represent the places where the insertions have been done. The number of dot is not relevant. In the sequence corresponding to a manuscript, the word limits have disappeared, because they are no more relevant, we will reconstruct them on the next step.

The insertion made is relative to the position of the word n-gram compared to the position of the candidate. If the candidate precedes the word n-gram (as in our example), we first need to shift the *T.M.* and then the *saṃhitapāṭha*, otherwise we shift the *saṃhitapāṭha* first.

2) The saṃhitapāṭha words alignment: After *candidates* has been correctly aligned with the word n-gram, we need to align the existing words of the word n-gram with the unsegmented *candidate*, according to the *saṃhitapāṭha* words. For example, with the example from Table VIII, the new word segment appears in Table XII. The alignment represented is not a good one, he still has a score = 6, which can be ameliorated by the following step, the bold letter are the letters which has been substituted.

We can now consider the new word boundaries and report them within the *T.M.*

E. Optimizing the new alignments

Making a new alignment on texts can induce new orphan letters. Thus, we apply the approach presented in Section III-A to obtain a better alignment. For instance, with our previous example, we identify many orphan letters as ‘c’, ‘aa’, ‘r’, etc. on the *T.M.* as shown below:

F. Building the final score

The final step of our approach is the final score of the new alignment computation, according to the whole environment (the neighboring words) and it delivery to an expert. The new computed score is quite different from the previous one used to select the candidates. The definition is the same: the number of substitutions made between the *saṃhitapāṭha* and the *T.M.* The difference lies in the texts we use for the comparison. For candidate selection, we only focus on a word n-gram and a *candidate*, which has the same number of letters. For the final score we use the new alignment.

G. Experiments

1) Experimental protocol: We use as a corpus the first chapter of the *kāśikāvṛtti*: the *Pratyāhrās sutra*, and we have used only 8 manuscripts (*T.M.*) randomly chosen from the fifty available. Each manuscript contains a maximum of 34 paragraphs. We first apply the alignment method, then after shifting the orphan letters, we obtain a baseline alignment. Our goal is to improve the baseline by lowering the number of letter substitutions. Recall that with edit distance, the more letter substitutions we have the poorer the alignment and the higher the score. Then we apply our approach. Considering $\boldsymbol{P}$ paragraphs, $\boldsymbol{K}$ manuscript, $\boldsymbol{w}$ maximum words by paragraphs, which means w - (n-1) word n-gram by paragraphs, and $\boldsymbol{m}$ tests of possible candidates for each word n-gram, the maximum numbers of operation $\boldsymbol{Nb}$ to apply our method is approximated by

$$\boldsymbol{Nb = P \times K \times m \times (w - (n - 1)).}$$

We have a complexity depending mostly on the $\boldsymbol{n}$ of n-gram, the number of paragraphs, the number of manuscripts, and the number of possible candidates to be tested.

TABLE X
THE BEST CANDIDATE IN A *manuscript*

k	aa	r	a	u	c	**c**	**aa**	**r**	**a**	**.n**	**aa**	**r**	**th**	**a**	**.h**	**l**	**a**	**k**	**aa**	**r**	**e**	**t**	**v**	a	k	aa	r	a	.h

TABLE XI
THE ALIGNMENT BEFORE THE WORD LIMITS RECONSTRUCTION

u	c	c	aa	r	a	.n	aa	r	th	a	.h	n	-	-	aa	**n**	**u**	**b**	**-**	**a**	**n**	**dh**	**-**	**a**	**.h**	**l**	**a**	**k**	**aa**	**r**	**e**	**t**	**v**	.....

u	c	.....	**c**	**aa**	**r**	**a**	**.n**	**aa**	**r**	**th**	**a**	**.h**	**l**	**a**	**k**	**aa**	**r**	**e**	**t**	**v**	a	k	aa

TABLE XII
THE NEWLY RECONSTRUCTED WORD BOUNDARIES

n	**u**	**b**	-	**a**	**n**	**dh**	-	a	.h	l	a	k	aa	r	e	t	v
c	**aa**	**r**	a	**.n**	**aa**	**r**	th	a	.h	l	a	k	aa	r	e	t	v

TABLE XIII
IDENTIFICATION OF THE NEW ORPHAN LETTERS.

u	c	c	aa	r	a	.n	aa	r	th	a	.h	n	aa	n	u	b	-	a	n	dh	-	a	.h	l	a	k	aa	r	e	t	v	-	-	-	-	.
u	c	-	-	-	-	-	-	-	-	-	-	-	-	**c**	**aa**	**r**	**a**	**.n**	**aa**	**r**	**th**	**a**	**.h**	l	a	k	aa	r	e	t	v	a	k	aa	r	.

TABLE XIV
THE NEW ORPHAN LETTERS AFTER A SHIFT.

u	c	c	aa	r	a	.n	aa	r	th	a	.h	n	aa	n	u	b	-	a	n	dh	-	a	.h	l	a	k	aa	r	e	t	v	-	-	-	-	...
u	c	**c**	**aa**	**r**	**a**	**.n**	**aa**	**r**	**th**	**a**	**.h**	-	-	-	-	-	-	-	-	-	-	-	-	l	a	k	aa	r	e	t	v	a	k	aa	r	...

For instance, with the parameters P=34, K=8, n=4, m= 40, we obtain:

$$\boldsymbol{Nb = 34 \times 8 \times 40 \times (w - (4-3)) = 10,880 \times (w-3).}$$

TABLE XV
EXPERIMENTAL RESULTS

n	*m*	*Nb*	*Score*
baseline			1,110
1 (words)	40	10,880 * l	922
2	40	10,880 * l	908
3	40	10,880 * (l – 2)	898
4	40	10,880 * (l – 3)	940
3	20	10,880 * (l – 2)	1,080
3	60	10,880 * (l – 2)	784
3	80	10,880 * (l – 2)	784

2) Experimental results: The results are displayed in Table XV. The baseline obtains a score of 1,110. It means that 1,110 substitutions were made during the construction of the alignment of *saṃhitapāṭha* and the 8 different manuscripts.

We first want to see the influence of the $\boldsymbol{n}$ parameter. According to our experiments, the best score is obtained with $\boldsymbol{n = 3}$ allowing 212 substitutions to be suppressed from the baseline. However, the $\boldsymbol{n}$ parameter does not seam to have an important influence on the scores.

In comparison, a modification of $\boldsymbol{m}$ parameter provides a better score with only 784 substitutions. It seams that 326 substitutions were deleted compared with the baseline. Note that we obtain the same score for $\boldsymbol{m = 60}$ and $\boldsymbol{m = 80}$; $\boldsymbol{m = 60}$ is better because we need fewer operations to obtain the same quality of alignment.

V. FUTURE WORK AND CONCLUSION

A. Future Improvement

Our first trials made us discover some unexpected situations that we do not take into account. Consider the alignment produced by our program in Table XVI, where the bold letters are orphan letters unusually placed in the *saṃhitapāṭha*. We need to shift them to get a better result which can be seen in Table XVII, where the italic letters refer to letters concerned by our second improvement.

Because of the bad alignment induced by the orphan letters in the *saṃhitapāṭha* our n_gram based method was unable to discover that the sequences in italic letters of Figure XVII was corresponding to a translocation, and then did not made the move which lead us to the right alignment such as displayed

TABLE XVI
AN UNEXPECTED ALIGNMENT.

a	t	r	a	d	v	i	r	v	a	c	a	n	a	m	p	r	aa	p	n	o	t	i	a	c	o	r	a	h	aa	bh	y	aa	.	m	...
a	t	r	a	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	a	c	o	r	a	h	aa	bh	y	aa	.	m	...

...	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**d**	**v**	**e**	**i**	-	-	**t**	**i**
...	d	v	e	i	t	i	d	v	i	r	v	a	c	a	n	a	m	p	r	aa	p	n	o	t	i

TABLE XVII
THE ALIGNMENT AFTER SHIFTING THE ORPHAN LETTERS OF THE *saṃhitapāṭha*

a	t	r	a	*d*	*v*	*i*	*r*	*v*	*a*	*c*	*a*	*n*	*a*	*m*	*p*	*r*	*aa*	*p*	*n*	*o*	*t*	*i*	a	c	o	r	a	h	aa	bh	y	aa	.	m	...
a	t	r	a	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	a	c	o	r	a	h	aa	bh	y	aa	.	m	...

...	**d**	**v**	**e**	**i**	**t**	**i**	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
...	d	v	e	i	t	i	*d*	*v*	*i*	*r*	*v*	*a*	*c*	*a*	*n*	*a*	*.m*	*p*	*r*	*aa*	*p*	*n*	*o*	*t*	*i*

TABLE XVIII
NEW WORD ALIGNMENT OBTAINED BY A SEQUENCE MOVE.

a	t	r	a	**d**	**v**	**i**	**r**	**v**	**a**	**c**	**a**	**n**	**a**	**m**	**p**	**r**	**aa**	**p**	**n**	**o**	**t**	**i**	a	c	o	r	a	h	aa	bh	y	aa	.	m	d	v	e	i	t	i
a	t	r	a	**d**	**v**	**i**	**r**	**v**	**a**	**c**	**a**	**n**	**a**	**m**	**p**	**r**	**aa**	**p**	**n**	**o**	**t**	**i**	a	c	o	r	a	h	aa	bh	y	aa	.	m	d	v	e	i	t	i

in Table XVIII, where the bold letters correspond to the letters newly aligned.

To solve this problem, it seems at first sight that we need only to take a better care of the orphan letters in the *saṃhitapāṭha*, but we need a real trial to check if this simple action will be sufficient .

We can remark that the use of n-gram is not the only possible way to solve the translocation (sequence move) problem. A problem quite similar exists in genomics, and can be solved by methods such as "**Glocal** alignment" [11]. These methods do not follow the same goal than we do, but they may be source of inspiration for our future work.

B. Improvement of the score computation

The score we use provides good results according to our purpose. But we would like to get a criterion which can provide a value near from zero when the two texts compared differ only by the usual differences that one can expect when comparing two different versions of the same text. Two reach this goal we will have to go step by step to drop out some elements that we don't want to take into account for the score.

For example some Sanskrit letters are prone to be confused by a scribe, so they need a special treatment, together for our score computation and in our alignment method. A second step could be a special treatment for the substitution of sparse letters which is obviously due to a different cause than a long sequence substitution.

On the other hand we have to build a second critter which can be used with a L.C.S. If we succeed on both of these goal we will be able to compare the two approaches (L.C.S. and edit distance) with a measure.

C. Using the critical edition as a distance

While the definition of a critical edition reminds us the one of the edit distance, we can build a distance between two manuscripts using the *saṃhitapāṭha* as an intermediary. This will provide us the informations for the construction of a phylogenetic tree or Stemma codicum [12] between the different manuscript. The construction of Stemma codicum will be a great help for all philologists interested in Sanskrit texts.

D. Conclusion

We have presented a tool for computer assisted construction of a critical edition which provides results which are quite satisfactory. The tool is still not perfect an we have described in the previous paragraphs some of the possible amelioration.

However, if we display our results with XML based tool, we can provide to the philologists a great help in the visualization of a critical edition allowing to chose dynamically which text should be compared and investigated. Sanskrit will have at last a tool which can be compared with the one existing for European languages.

REFERENCES

[1] P. O'Hara, R.J. Robinson, "Computer-assisted methods of stemmatic analysis," in *Occasional Papers of the Canterbury Tales Project*, N. Blake and P. Robinson, Eds. Oxford University: Office for Humanities Communication, 1993, vol. 1, pp. 53–74.

[2] C. Monroy *et al.*, "Visualization of variants in textual collations to analyse the evolution of literary works in the cervantes project," in *Proceedings of the 6th European Conference, ECDL 2002*, M. Agosti and e. Constantino Thanos, Eds. Rome, Italy: Springer, September 2002, pp. 638–53.

[3] M. Csernel and F. Patte, “Critical edition of sanskrit texts,” in *Sanskrit Computational Linguistics*, ser. Lecture Notes in Computer Science, vol. 5402, 2009, pp. 358–379.
[4] M. Csernel and T. Cazenave, “Comparing sanskrit texts for critical editions,” in *COLING*, Beijing, 2010, pp. 206–213.
[5] F. Velthuis, *Devanāgarī for TEX, Version 1.2, User Manual*, University of Groningen, 1991, http://www.ctan.org/tex-archive/language/devanagari/velthuis/.
[6] U. Consortium, “Unicode standard version 6.0: Devanagari,” http://unicode.org/charts/PDF/U0900.pdf, Inria, 2010.
[7] G. Huet, “Héritage du sanskrit: Dictionnaire français-sanskrit,” http://sanskrit.inria.fr/Dico.pd, Inria, 2006.
[8] ——, “Design of a lexical database for sanskrit,” in *COLING Workshop on Electronic Dictionaries*, Geneva, 2004, pp. 8–14.
[9] R. L. Solso, “Bigram and trigram frequencies and versatilities in the english language,” *In Behavior Research Methods & Instrumentation*, vol. 11, no. 5, pp. 475–484, 1979.
[10] H. Lei and N. Mirghafori, “Word-conditioned phone N-grams for speaker recognition,” in *Proc. of ICASSP*, Honolulu, 2007.
[11] M. Brudno and al, “Glocal alignment: finding rearrangements during alignment,” in *ISMB (Supplement of Bioinformatics)*, 2003, pp. 54–62.
[12] M. Le Pouliquen, “Using lattices for reconstructing stemma,” in *Fifth International Conference on Concept Lattices and Their Applications, CLA.*, 2007.

Razonamiento espacial para determinar el dominio de un conjunto de etiquetas que representan objetos geográficos

Eduardo Loza-Pacheco, Miguel Torres-Ruiz y Giovanni Guzmán-Lugo

***Resumen*—Actualmente, existe una gran cantidad de información geográfica, proveniente de diversas fuentes como imágenes de satélite, fotografías aéreas, mapas, bases de datos, entre otras. Estas fuentes proporcionan una descripción exhaustiva de los objetos geográficos. Sin embargo, la tarea de identificar el dominio geográfico al que pertenecen involucra un procesamiento semántico, el cual está basado en la conceptualización de un dominio, lo que permite interpretarlo de una manera similar a como los seres humanos reconocen a las entidades geográficas y evitar así la vaguedad. Este trabajo propone un método para realizar un proceso de razonamiento espacial cualitativo en representaciones geográficas. El método se basa en el conocimiento *a priori* del dominio, el cual se encuentra explícitamente formalizado a través de una ontología. El conocimiento descrito en la ontología se valora de acuerdo con un conjunto de etiquetas que pertenecen a algún tipo de dominio geográfico para realizar el análisis semántico correspondiente y mapear esas etiquetas con los conceptos definidos en la ontología. Como resultado, se obtiene un conjunto de dominios geográficos ordenados por su relevancia, para proporcionar un concepto general relacionado directamente con las etiquetas de entrada, simulando la forma en que cognitivamente percibimos algún dominio geográfico en el mundo real.**

***Palabras Clave*—Inteligencia artificial y computacional, sistemas inteligentes, adquisición de conocimiento, representación de conocimiento.**

Spatial Reasoning for Determining the Domain of the Set of Tags that Represent Geographic Objects

Eduardo Loza-Pacheco, Miguel Torres-Ruiz
and Giovanni Guzmán-Lugo

***Abstract*—Nowadays, there is much geospatial information from different sources such as satellite images, aerial photographs, maps, databases, and so on. It provides a comprehensive description of geographic objects. However, the task to identify the geographic domain to which it belongs is not simple, because this task involves semantic processing based on a conceptualization of a domain. It allows us to understand information in a way similar to how humans recognize the geographic entities and avoid vagueness. We propose a method for qualitative spatial reasoning in geospatial representations. The method is based on *a priori* knowledge, which is explicitly formalized through an ontology. The knowledge described in the ontology is assessed according to a set of tags that belong to any geographical domain for semantic analysis, to map those tags to concepts defined in the ontology. As a result, a set of geographic domains in order of relevance is obtained, for providing a general concept directly related to the input tags, simulating the way in which humans cognitively perceive a geographic domain in the real world.**

***Index Terms*—Computational and Artificial Intelligence, Intelligent Systems, Knowledge Acquisition, Knowledge Representation.**

Manuscrito recibido el 9 de marzo de 2012, manuscrito aceptado el 7 de mayo de 2012.

Los autores trabajan en el Centro de Investigación en Computación del Instituto Politécnico Nacional, Av. Juan de Dios Bátiz, s/n, UPALM-Zacatenco, 07738, México D.F., México (email: eduardo.loza@gmail.com, mtorres@cic.ipn.mx, jguzmanl@cic.ipn.mx).

I. Introducción

Hoy en día, los Sistemas de Información Geográfica (SIG) se han convertido en herramientas muy populares para la representación y razonamiento de datos geográficos [1]. Estas aplicaciones requieren de métodos de razonamiento acerca de las entidades geográficas y de las relaciones que se definen entre sí, incluyendo todas aquellas relaciones espaciales [2]. Los algoritmos de razonamiento son ampliamente utilizados en el campo de la Inteligencia Artificial, cuyas tareas más relevantes son la capacidad de verificar la consistencia de los conjuntos de datos, actualizar el conocimiento compartido, derivar nuevo conocimiento y encontrar una representación mínima [3, 4].

No obstante, antes de realizar alguna tarea de razonamiento, es necesario tomar en cuenta una representación formal que nos permita conceptualizar el conocimiento del dominio de interés [5]. En este caso, las ontologías son herramientas muy ponderosas para conceptualizar cualquier contexto, describiendo sus conceptos y expresando sus relaciones. Además, las ontologías han sido referidas como un método para llevar a cabo este tipo de razonamiento [6–8]. Sin embargo, existen algunas metodologías que no manejan adecuadamente la vaguedad inherente de los datos espaciales.

Las entidades geográficas son frecuentemente dependientes del contexto en el cual residen, con un conocimiento local que afecta a las definiciones [9]. Asimismo, los objetos geográficos no demarcan claramente en algunas ocasiones a una entidad, ya que pueden formar parte de algún otro objeto [10]. Por lo tanto, la individualización de las entidades es más importante, con respecto a los dominios geográficos que representan o pueden pertenecer.

De acuerdo con [11], la *vaguedad* es inherente a los dominios geográficos, con relación a muchos elementos que

son dependientes del contexto, así como carentes de definiciones y límites precisos. La vaguedad no es un defecto de nuestro lenguaje de comunicación sino más bien una parte útil e integral. Como una consecuencia, los SIG no manejan adecuadamente múltiples interpretaciones, por lo cual la carencia de esta característica implica la creación de nuevas técnicas que permitan el manejo de múltiples significados, así como la inferencia basada en razonamiento.

En este trabajo se propone un método que permite llevar a cabo un proceso de razonamiento espacial cualitativo, sobre un conjunto de objetos geográficos que están representados como etiquetas de entrada y pertenecen a algún dominio geográfico. Para el proceso de inferencia se proponen tres algoritmos que realizan un razonamiento espacial, los cuales consideran el conocimiento *a priori*, el cual se define por medio de una ontología de aplicación y marcos conceptuales. El proceso de razonamiento se basa fundamentalmente en el procesamiento de las relaciones topológicas, las cuales se encargan de describir la interacción o comportamiento de un objeto geográfico con respecto a otros.

Este artículo está organizado de la siguiente manera: en la Sección 2 se presenta el estado del arte con respecto a los trabajos más relevantes en esta área. Asimismo, la Sección 3 describe la metodología propuesta para llevar a cabo el razonamiento espacial cualitativo. La Sección 4 muestra los resultados obtenidos. Por último, las conclusiones son presentadas en la Sección 5.

II. Trabajos relacionados

La geometría de puntos y líneas es una de las más ancestrales ramas del razonamiento espacial [12]. La abstracción de un punto sin dimensión es el elemento básico de todas las entidades espaciales que tienen que ser construidas con base en los puntos. Una de las teorías más antiguas es la Geometría Euclidiana, cuyo sistema axiomático se utiliza hoy en día. La idea de todo el razonamiento geométrico puede estar basada, en las aplicaciones orientadas a razonamiento matemático.

Los trabajos de razonamiento espacial cualitativo son precedidos por un conjunto de representaciones espaciales, en donde se busca que puedan ser leídos e interpretados por una máquina [8]. En [13], se menciona la importancia de que exista una correcta representación de la realidad para llevar a cabo un proceso de razonamiento espacial, ya que una de las razones para representar el conocimiento acerca de un dominio más que el mismo dominio en sí mismo, es a través del conocimiento, por lo cual el mundo sería mucho más accesible por medios formales. Esto se debe a que las máquinas están sujetas al uso de enfoques formales si se quiere realizar algún proceso de razonamiento. Sin embargo, la información capturada debe contener descripciones lo más cercanas a como el ser humano percibe su entorno [Egenhofer, 1995]. Además de que uno de los principales objetivos del razonamiento espacial cualitativo es encontrar maneras adecuadas de representar propiedades continuas del mundo, utilizando un sistema basado en símbolos discretos [14], [15].

Las representaciones espaciales cualitativas han estado evolucionado rápidamente, esto se puede observar en [16], en donde se propone un conjunto de relaciones binarias *C(x,y)*, la cual se puede leer como "x conecta y", y se demuestra que esta relación cumple con las propiedades de simetría y reflexividad. Este tipo de relaciones han sido definidas para el trabajo con regiones espaciales, en donde se puede presentar la mayor ambigüedad entre las entidades geográficas. Para el caso del *razonamiento espacial temporal*, en [17], se propone un conjunto de relaciones que se encargan de capturar el comportamiento entre dos intervalos, siendo este conjunto explícito muy utilizado actualmente en los modelos espacio-temporales. En la Figura 1 se muestra el conjunto de las relaciones de Allen.

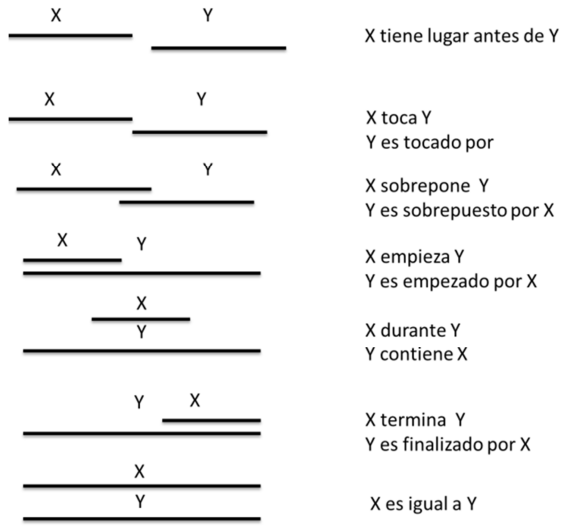


Fig. 1. Relaciones temporales propuestas por Allen.

El uso de estas relaciones se extiende más allá del razonamiento espacial temporal, como ha sido descrito en trabajos como [18], donde este conjunto de relaciones puede expandirse a un dominio $\mathbb{R}^2$. Por otro lado, existen nuevos modelos de representación como el modelo 9-intersección de Egenhofer [19], el cual descompone una región en términos de su interior y frontera. Este modelo define un conjunto relaciones espaciales topológicas entre dos regiones que se describen por una tupla de cuatro elementos, donde el interior se denota por el símbolo ($^\circ$) y la frontera se define por (∂). Las relaciones espaciales topológicas se denotan por una tupla de cuatro elementos $<a,b,c,d>$. Las entradas corresponden al orden de los valores topológicos invariantes, asociados a las cuatro intersecciones, las cuales se denominan *frontera-frontera*, *interior-interior*, *frontera-interior* y la cuarta *interior-frontera*. Además, se restringen las relaciones espaciales topológicas definidas por los valores vacío ($\emptyset$) y no-vacío ($\neg\emptyset$) a las entrada de la tupla [20].

Asimismo, se ha propuesto el modelo RCC8, el cual es una representación del espacio que provee un conjunto de ocho relaciones topológicas entre dos regiones [21]. En la Figura 2 se muestra el conjunto de relaciones propuesto. Este modelo

describe las relaciones: dentro, afuera, toca, sobrepone, contiene, cubre, disjunto e igual para objetos que se definen como conjuntos regulares. Por tanto, el conjunto RCC8 cuenta con una semántica bien formada y basada en la Geometría Euclidiana [22].

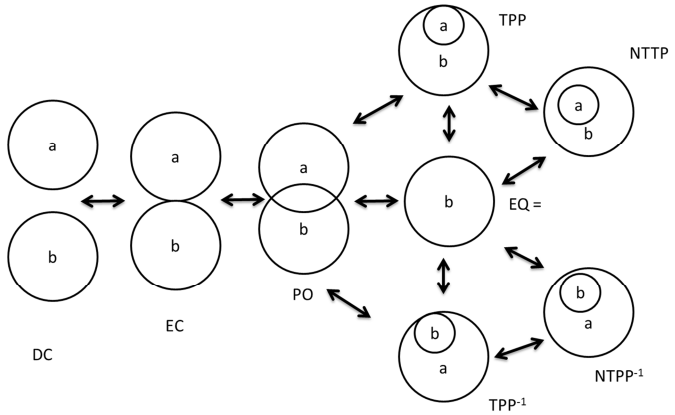


Fig. 2. Modelo de relaciones topológicas RCC8.

Por otra parte, existen diversas propuestas para motores de razonamiento espacial, entre ellas se encuentra el trabajo de El-Geresy [23], el cual propone SPARQS. Este modelo es un motor de razonamiento espacial cualitativo, cuyo enfoque se basa en la generación de relaciones espaciales que se definen mediante tablas de composición de manera automática. El formalismo presentado en este trabajo está dividido en dos partes que consisten en las restricciones generales que rigen a las relaciones espaciales entre los objetos en el espacio y las reglas generales para propagar las relaciones entre los objetos. Tanto las restricciones y las reglas están basadas en una representación uniforme de la topología de los objetos, el espacio que lo rodea y la representación de las relaciones entre los objetos. Un trabajo similar es el propuesto por Grutter [24], el cual propone un conjunto de reglas y restricciones a partir del modelo RCC8. Asimismo, Wang [25] propone nuevos modelos de relaciones espaciales que agrupan a las relaciones principales (dirección, topología, distancia y forma), con la finalidad de implementar una aplicación que permita recuperar esta información en una base de datos. Por otra parte, Eagleson en [26], enumera las desventajas de las fronteras en los mapas geográficos y en especial con relación a las fronteras políticas. La utilidad de definirlas de manera adecuada para sus posibles usos en razonamiento espacial jerárquico utilizando agregaciones espaciales.

En el caso de las relaciones de dirección, en [Frank,1996] se describe que los SIG y en Geografía en general, se tiene que lidiar de manera común con espacios de gran escala, y es ahí donde las relaciones espaciales de dirección tienen su mayor utilidad, puesto que éstas son utilizadas casi exclusivamente para estos casos. De igual forma, se han planteado trabajos que utilizan el razonamiento espacial enfocado en aplicaciones como la navegación, robótica y planificación de rutas [27]. Este trabajo consiste en recibir un conjunto de descripciones del arreglo espacial de los objetos, en la forma de un lenguaje definido por el usuario, con el propósito de que sea entendible. Además, se utiliza un subconjunto de relaciones espaciales que son las relaciones de dirección (arriba, abajo, derecha, izquierda).

La propuesta de Clementini [28], tiene como objetivo definir una taxonomía para emplearse en aplicaciones de razonamiento espacial. Este método define tres tipos de marcos de referencia, los cuales son: el *intrínseco* que es establecido con un objeto de referencia que determina el origen del sistema de coordenadas, así como la orientación. El *extrínseco* que puede heredar su origen de un objeto de referencia; sin embargo, su orientación se determina por factores externos como podría ser un objeto convencional o una marca. Finalmente, el *diético* que envuelve los tres objetos, el objeto primario que es una relación particular con respecto al objeto de referencia y el punto de vista. La relación es impuesta en el objeto de referencia de acuerdo con su punto de vista.

Existen varios trabajos que han sugerido la combinación de diversas relaciones geográficas como el descrito en [8], el cual combina las relaciones topológicas y las de dirección, o bien propuestas que combinan las relaciones de Allen, utilizadas normalmente en $\mathbb{R}^1$ para proyectarlas en $\mathbb{R}^2$ y usar esta forma de representación para relaciones topológicas y de dirección [18]. Otros esfuerzos están orientados directamente a combinar representaciones [29], en donde se aplican técnicas de razonamiento espacial para solucionar problemas donde existe una cantidad pequeña de datos y cuyos formalismos utilizan el modelo RCC8 y el álgebra de Allen. Por otro parte, están los problemas con una gran cantidad de información por ejemplo, la información que se recaba para los servicios meteorológicos. Por lo anterior, es necesario el uso de agregaciones espaciales. En este sentido, Shultz [30] utiliza la combinación de dos modelos el de proximidad cualitativa y el RCC, con la finalidad de determinar la distancia cualitativa entre dos objetos. Utilizando el modelo RCC, el primer paso es determinar la relación entre los objetos geográficos. Después utilizando distancias cualitativas, se toman las distancias entre cada par de objetos, mediante la fórmula de distancia entre dos puntos para un espacio en dos dimensiones. Aquí se combinan los dos enfoques el razonamiento espacial cuantitativo y el cualitativo.

Actualmente han surgido aplicaciones orientadas a la recuperación de imágenes para comprender que tipo de información envuelve a una imagen [31], [32]. Asimismo, es necesario entender como se conceptualiza una imagen o bien que información contiene y si es posible definirla, en caso contrario se pueden presentar problemas relacionados con la ambigüedad [33]. En [34] se menciona que una imagen es una descripción ontológica de sí misma. Como es el caso de [35], cuyo trabajo está enfocado en la obtención de un formalismo que sea lo suficientemente expresivo para incrementar la utilidad de una descripción pictográfica. Se menciona que los mecanismos de recuperación deben trabajar con información textual, imágenes, sonidos, videos, etc. Además se establece que existen dos tipos de recuperación de imágenes. El primero es del tipo sintáctico, donde las imágenes son guardadas en

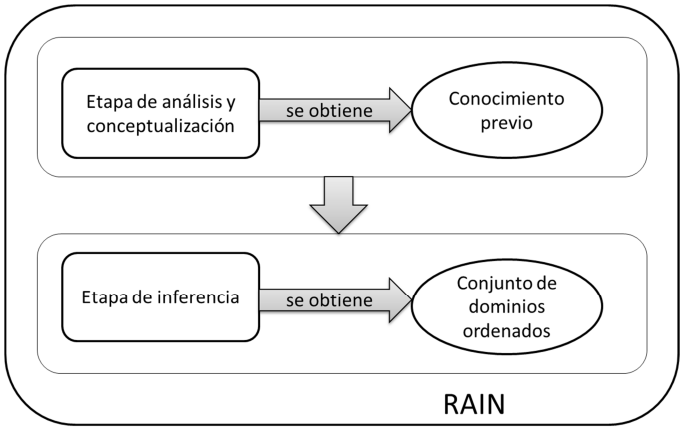


Fig. 3. Marco general de la metodología RAIN.

una base de datos, tomando como criterio algún valor físico de ésta. El segundo tipo es el enfoque basado en su significado, se propone que la imagen sea descrita en términos del modelo RCC8 para expresar las relaciones topológicas entre los objetos [35].

En [36] se presenta una metodología para la recuperación de imágenes para la búsqueda en grandes colecciones de datos. La aproximación propuesta emplea un algoritmo de segmentación no supervisado, el cual permite dividir la imagen en regiones. A partir de cada una de estas regiones, se hace una extracción de características de bajo nivel como son: color, posición, tamaño y forma, con el objeto de emplear un conjunto de descriptores que permitan describir dichas regiones, asociando a cada descriptor un vocabulario simple conocido como "objeto ontológico". En [37], se propone utilizar una herramienta más semántica, la cual pretende realizar una búsqueda y anotación semántica dentro de una colección de imágenes, empleando ontologías como la *Art and Architecture Theasaurus* (AAT), *WordNet, Union List of Artist Names* (ULAN) e *IconClass*. Finalmente, en [38] se presenta una técnica basada en razonamiento espacial, la cual está compuesta por dos partes: la extracción de los objetos de la imagen remota y la construcción de una representación gráfica que sirve para realizar una etapa de razonamiento. Para la parte de extracción de objetos se propone una segmentación multiespacial de algoritmos de segmentación cóncavo, convexo, luminoso y oscuro e imágenes a diferente escala. Posteriormente, se utiliza el modelo RCC8 para construir un grafo basado en las relaciones entre las regiones y se construye un árbol de decisión binario.

III. METODOLOGÍA RAIN

RAIN es una técnica cuyo enfoque principal consiste en establecer un conjunto de técnicas para realizar un proceso de razonamiento espacial, en descripciones semánticas del contexto geográfico, tomando en consideración el conocimiento *a priori* del dominio geoespacial. Para ello, se formaliza este conocimiento por medio de una estructura conceptual (ontología), con la finalidad de que sea legible para una máquina.

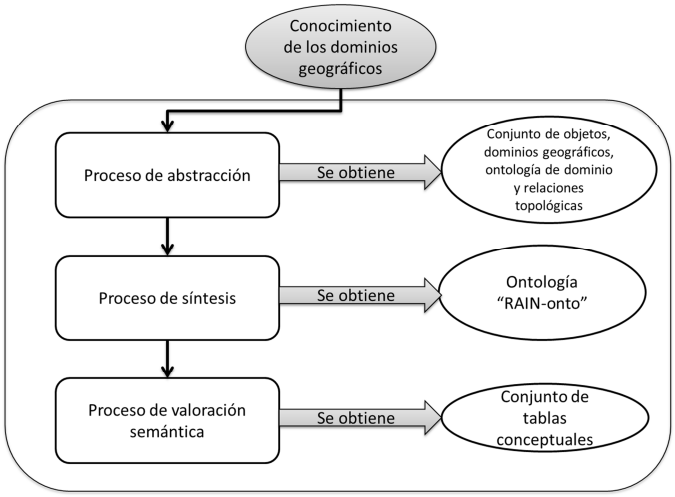


Fig. 4. Etapa de análisis y conceptualización.

Este método está orientado en la conceptualización de los conceptos y las relaciones que existen en un dominio dado, con la finalidad de responder a una pregunta sobre a qué dominio pertenecen un conjunto de descripciones y cuál es la relevancia de los conceptos en ese dominio. RAIN permite conocer el contexto de un conjunto de descripciones que a simple vista parecen inconexas. Esta técnica se compone de dos etapas: 1) *Análisis y conceptualización* y 2) *Inferencia*. En la primera, se obtiene un conocimiento previo, definido con base en las necesidades del razonamiento. En la etapa de Inferencia se obtiene un conjunto de dominios ordenados, de acuerdo con la cercanía o similitud de las descripciones recibidas como entrada. En la Figura 3 se muestra el marco general de la metodología RAIN.

A. Etapa de análisis y conceptualización

El proceso de conceptualización (ver Figura 4) está basado en el método para diseñar ontologías del dominio geográfico GEONTO-MET, propuesto en [39].

Con base en esta técnica, se han utilizado los dos conjuntos de relaciones axiomáticas base que han sido definidos: $A_1 = \{es, tiene, hace\}$ y $A_2 = \{preposiciones\}$ para traducir directamente las relaciones entre los conceptos como parte de la conceptualización; es decir, la esencia fundamental es reducir las relaciones axiomáticas en la ontología, con ello por ejemplo, relaciones topológicas como conecta, cruza, contiene, entre otras; son definidas como conceptos del tipo relación en la conceptualización, con lo cual se obtiene una mayor expresividad, granularidad y riqueza semántica en la representación [40]. Asimismo, estas relaciones axiomáticas son utilizadas para definir a los conceptos y clases en la ontología.

Proceso de abstracción: Esta tarea tiene como propósito realizar una revisión exhaustiva sobre los objetos geográficos que se encuentran involucrados en el conjunto de dominios geográficos, con la finalidad de llevar a cabo un proceso de abstracción que defina un conocimiento *a priori*. Para esta tarea la información fue recopilada de una ontología de dominio *Kaab* (significa Tierra en el lenguaje maya), definida

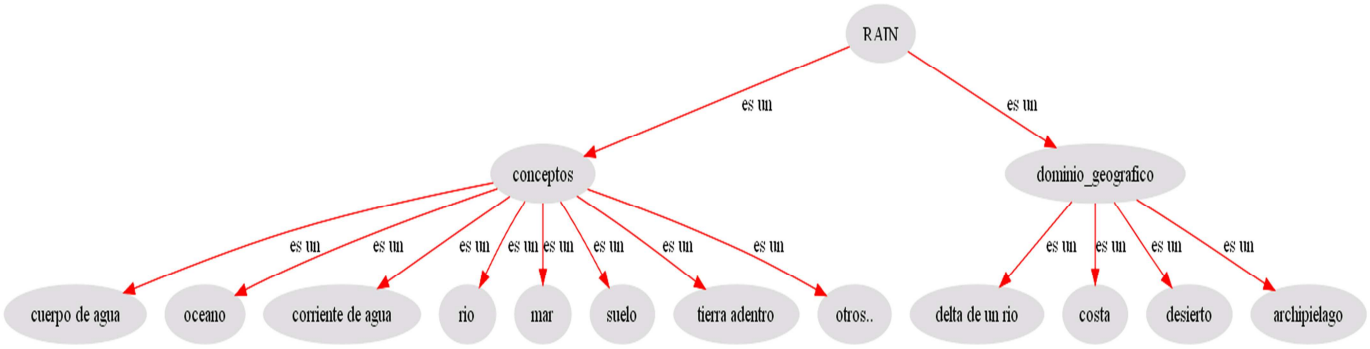


Fig. 5. Fragmento de la ontología RAIN.

en [40], cuya ventaja radica en que se pueden encontrar entidades abstractas y sus relaciones [39]. Se utilizaron definiciones del diccionario de datos del Instituto Nacional de Estadística, Geografía e Informática (INEGI) [41] y del Centro Nacional para Ontologías Biomédicas [42]. Como resultado de esta etapa, se obtuvo la información necesaria de todos aquellos objetos y dominios geográficos que nos interesa identificar en la etapa de razonamiento o inferencia. En la Figura 5 se muestra la información recabada y un fragmento de la ontología construida a partir de *Kaab*.

Proceso de síntesis: Para el proceso de síntesis ya se cuenta con los objetos y dominios que intervienen en el proceso de razonamiento. Sin embargo, esta información no se encuentra estructurada.

Por tanto, es necesario definir cada dominio, de acuerdo con la pertenencia y relación de los objetos geográficos que intervienen en él.

Entonces, se define la *relación topológica* que existe entre los objetos geográficos, la cual se representa por medio de una relación jerárquica descrita entre ellos, junto con sus propiedades y sinónimos o *alias*, tanto de los dominios como de los objetos geográficos.

En este proceso se lleva a cabo un *mapeo* de los objetos geográficos con respecto a los conceptos definidos en la ontología, con la finalidad de iniciar el poblado de la estructura conceptual.

La ontología *Kaab* cuenta con un conjunto de entidades abstractas del dominio geográfico que ayudan a delimitar el dominio y restringir el número de conceptos que están involucrados en cada uno.

En la Figura 6 se muestra el proceso de síntesis, en donde se puede observar que el conjunto de conceptos y relaciones topológicas permiten la definición del conjunto de dominios, los cuales interactúan con las ontologías para extraer las instancias que generalizarán el proceso; es decir, a partir de la descripción de muchos conceptos especializados, se obtiene un concepto general que describe al dominio en cuestión. Asimismo, en la Figura 7 se presenta el proceso de mapeo de la información recabada con la ontología *Kaab*.

Proceso de valoración semántica: El proceso de valoración semántica utiliza la ontología propuesta, junto con un conjunto de tablas para obtener la información con respecto a la definición de conceptos, sus relaciones con otros conceptos como con los dominios a las que pertenecen, sus reglas y restricciones. Esto con la finalidad de refinar y asignar un valor semántico a cada dominio, con base en la construcción de las tablas de conceptos que contienen las propiedades, lugar en la jerarquía, nombres y sinónimos; así como una tabla de sinónimos de los conceptos, una tabla de dominios, una tabla de sinónimos de los dominios, una tabla de frecuencia de conceptos en los dominios, una tabla de composición de relaciones topológicas ordenadas, de acuerdo con su relevancia y por último, una tabla de refinación semántica para mejorar el proceso de inferencia, basada en la retroalimentación. A continuación se definen las tablas mencionadas para la valoración semántica.

Tabla de conceptos: En esta tabla aparecen todos los posibles conceptos que forman parte de cada uno de los dominios. Esta tabla permite expresar de manera explícita las características que contiene cada uno de los conceptos. La tabla contiene la información acerca de los sinónimos o de los *alias* con que se conoce a los conceptos y sus propiedades. Adicionalmente, permite conocer la ubicación de cada concepto en la jerarquía, identificando los conceptos padre e hijo de cada concepto. Cada concepto puede tener *n* número de hijos; sin embargo, solo puede tener un padre.

Asimismo, se expresan las relaciones que se pueden operar sobre ese concepto. Finalmente, se obtiene una ontología con todo el conjunto de propiedades, conceptos que contienen los dominios y la jerarquía de cada concepto (ver Figura 8). De

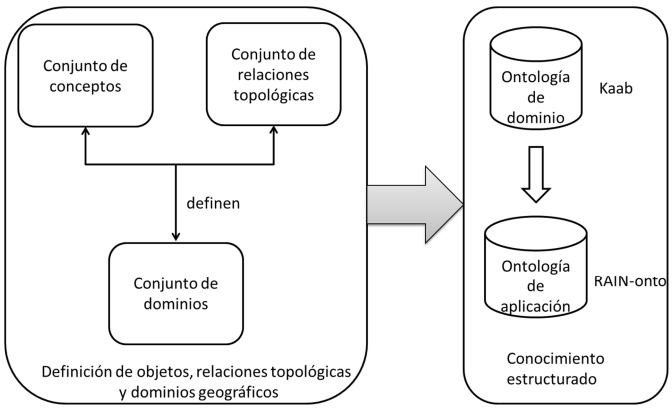


Fig. 6. Representación del proceso de síntesis.

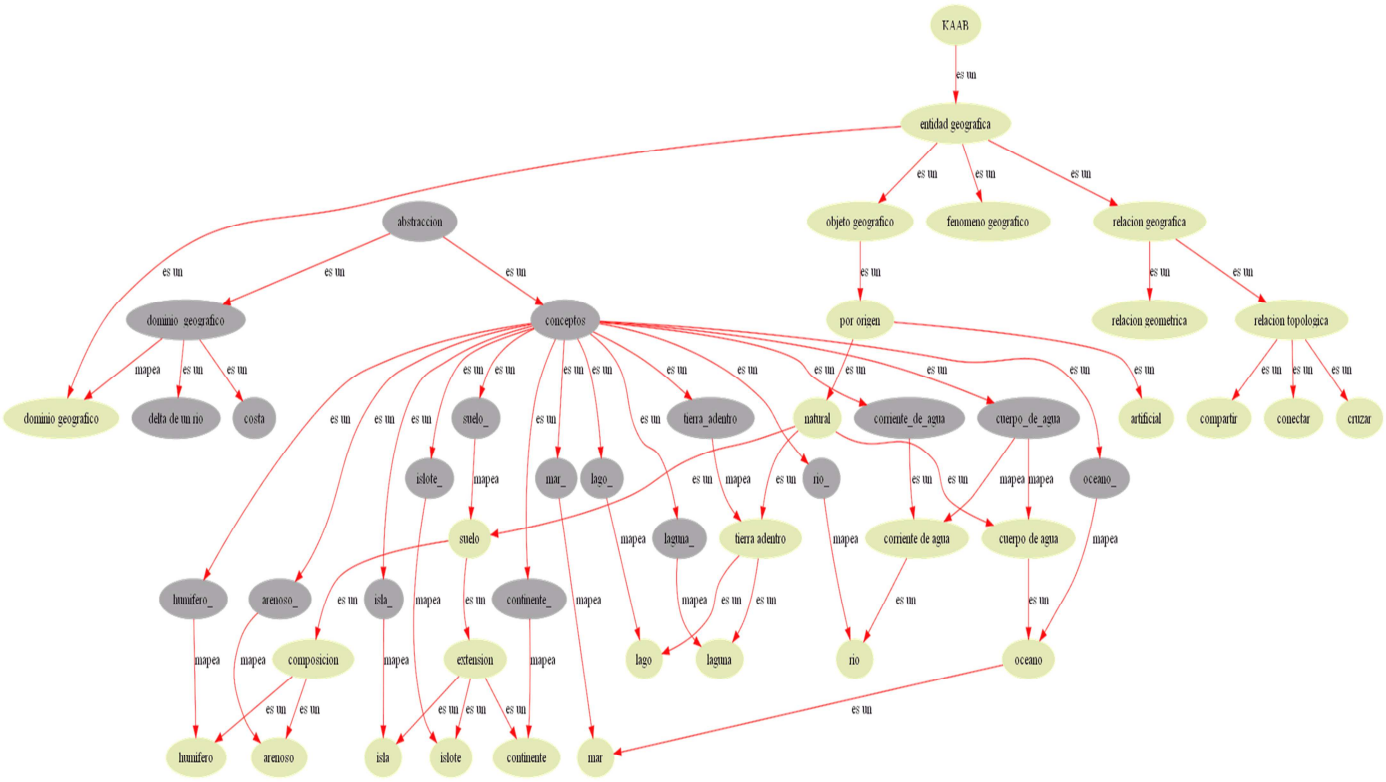


Fig. 7. Proceso de mapeo.

igual forma, se describen las definiciones para la generación de la tabla de conceptos (ver tabla I).

Para una tabla de conceptos, sea un concepto $c_{i,j}$ que pertenece al conjunto C, por lo cual se propone la función $father(c_{i,j})$ que devuelve el concepto $c_{p,q}$, definindo como el padre de $c_{i,j}$. Por lo tanto, $father(c_{i,j}) = c_{p,q}$ es definida.

Asimismo, sea un concepto $c_{i,j}$ relacionado con la función $son(c_{i,j})$ que devuelve el (los) concepto (s) $c_{p,q}$ que es (son) hijo (s) de $c_{i,j}$, por tanto, la función queda definida como sigue: $son(c_{i,j}) = c_{p,q}$.

De igual manera, sea $P_{i,j}$ el conjunto de las propiedades que pertenecen directamente a un concepto $c_{i,j}$, de un dominio en particular $d_i|d_i \in D$, definido como se expresa a continuación: $P_{i,j} = \{p(i,j,1), p(i,j,2), \dots, p(i,j,n)$.

Con respecto a las relaciones, sea $R_{i,j}$ un conjunto de relaciones topológicas que aplican a un concepto $c_{i,j}$ de un dominio en particular $d_i|d_i \in D$, definido de la siguiente forma: $R_{i,j} = \{r(i,j,1), r(i,j,2), \dots, r(i,j,n)$.

TABLA I
TABLA DE CONCEPTOS

Nombre Concepto	Sinónimo	Propiedad	Padre	Hijo	Relación Topológica
$c_{i,1}$	$S_{i,1}$	$P_{i,1}$	$father(c_{i,1})$	$son(c_{i,1})$	$R_{i,1}$
$c_{i,2}$	$S_{i,2}$	$P_{i,2}$	$father(c_{i,2})$	$son(c_{i,2})$	$R_{i,2}$
...	...	...	...	...	...
$c_{i,n}$	$S_{i,n}$	$P_{i,n}$	$father(c_{i,n})$	$son(c_{i,n})$	$R_{i,n}$
...	...	...	...	...	...
$c_{m,n}$	$S_{m,n}$	$P_{m,n}$	$father(c_{m,n})$	$son(c_{m,n})$	$R_{m,n}$

De acuerdo con lo anterior, la función *exist_concept*, recibe una etiqueta et_i, la cual regresa como resultado el número de concepto de interés o relevancia. En caso contrario, devuelve un valor de falso si no existe el concepto. Esta función se define de la siguiente forma:

$$exist\_concept(et_i) = \begin{cases} c_j | 0 \leq j \leq n \\ falso \end{cases}$$

De igual forma, para la generación de la tabla de conceptos, es necesario definir la función *exist_synonymous*, la cual recibe una etiqueta, regresando como resultado el número de concepto de interés o relevancia. En caso contrario, devuelve un valor de falso cuando no existe algún sinónimo.

$$exist\_synonymous(et_i) = \begin{cases} c_j | 0 \leq j \leq n \\ falso \end{cases}$$

Por lo tanto, de acuerdo con las definiciones anteriores la tabla de conceptos se define como sigue:

Tabla de frecuencia de conceptos en dominios: De acuerdo con la información recopilada de cada dominio que fue definido previamente, existe un conjunto de conceptos pertenecientes a un dominio en particular, con la finalidad de obtener la frecuencia de los conceptos en los dominios geográficos.

Por tanto, sea D el conjunto de dominios, donde d es un dominio geográfico en particular que ha sido definido en una conceptualización. Entonces ese conjunto se define de la siguiente forma: $D = \{d_1, d_2, \dots, d_n\}$.

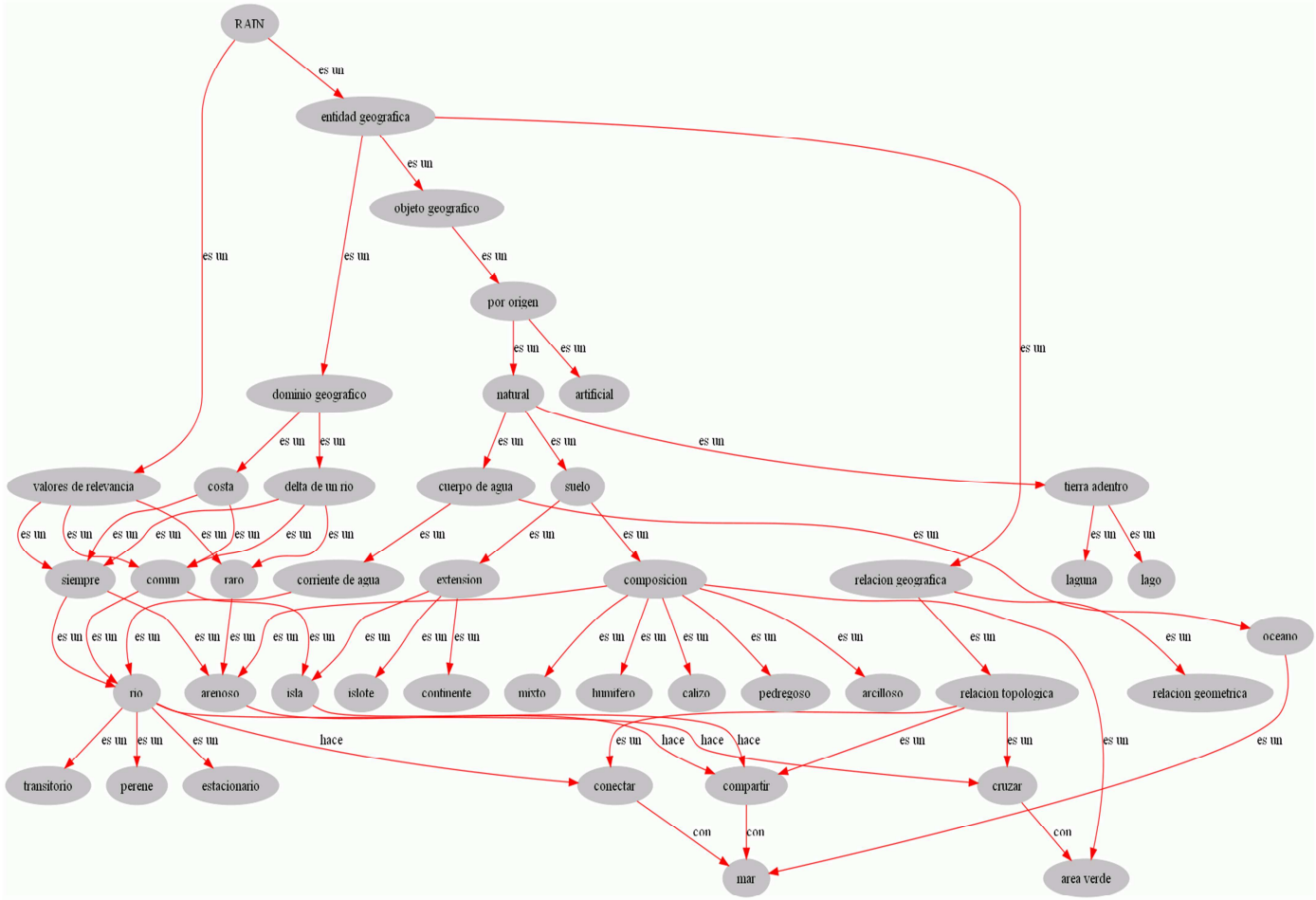


Fig. 8. Ontología RAIN obtenida del proceso de valoración semántica.

Asimismo, sea C el conjunto de conceptos, en donde c son los conceptos involucrados en un dominio geográfico, definido por la siguiente expresión: $C = \{c_1, c_2, \dots, c_n\}$.

Entonces, sea C_i un conjunto de conceptos que pertenece a un dominio en particular, denotado por $d_i | d_i \in D$ y definido de la forma siguiente: $C_i = \{c_{(i,1)}, c_{(i,2)}, \dots, c_{(i,n)}\}$.

Con las expresiones anteriores, se construye una tabla de frecuencia de conceptos en dominios, la cual contendrá todos los dominios que se desean introducir en la ontología O; así como todos los conceptos que se encuentran en cada dominio (ver Tabla II).

Para simplificar la notación referente a los conceptos, se puede definir una función que permita asociar un concepto, con un número de dominio, así como el índice del concepto dentro de su propio conjunto, la cual esté definida de la siguiente manera.

Sea la función *concept*, la cual recibe dos parámetros que describen el número de dominio y el número de concepto, regresando como resultado el número de concepto de interés o falso en caso de que no exista el concepto. Por tanto, la función es: $concept(d, c) = c_{j,k} | 0 \leq j \leq m, 0 \leq k \leq n$.

Tabla de sinónimos de dominios: Una tabla de sinónimos, permite conocer los posibles *alias* que puede tener un dominio en particular, relacionado directamente con su búsqueda (ver Tabla III). Para ello, sea S_i el conjunto de sinónimos para un determinado dominio d_i, el cual está definido por: $S_i = \{s_{(i,1)}, s_{(i,2)}, \dots, s_{(i,n)}\}$.

Tabla de composición de relaciones topológicas: Como parte del procesamiento semántico, es necesario obtener el conjunto de relaciones existentes entre los conceptos que intervienen directamente en cada dominio y definir su relevancia para

TABLA II
TABLA DE FRECUENCIA DE CONCEPTOS EN DOMINIOS

Dominio	Conceptos
D_1	$C_1 = \{c_{(1,1)}, c_{(1,2)}, \dots, c_{(1,n_1)}\}$
D_2	$C_2 = \{c_{(2,1)}, c_{(2,2)}, \dots, c_{(2,n_2)}\}$
D_3	$C_3 = \{c_{(3,1)}, c_{(3,2)}, \dots, c_{(3,n_3)}\}$
...	...
D_m	$C_m = \{c_{(m,1)}, c_{(m,2)}, \dots, c_{(m,n_m)}\}$

TABLA III
TABLA DE SINÓNIMOS DE DOMINIO

Dominio	Conceptos
D_1	$S_1 = \{s_{(1,1)}, s_{(1,2)}, \dots, s_{(1,n_1)}\}$
D_2	$S_2 = \{s_{(2,1)}, s_{(2,2)}, \dots, s_{(2,n_2)}\}$
D_3	$S_3 = \{s_{(3,1)}, s_{(3,2)}, \dots, s_{(3,n_3)}\}$
...	...
D_m	$S_m = \{s_{(m,1)}, s_{(m,2)}, \dots, s_{(m,n_m)}\}$

TABLA IV
TABLA DE COMPOSICIÓN DE RELACIONES TOPOLÓGICAS

Dominio	Etiquetas de Relevancia		
	Necesario	Común	Raro
D_{t1}	N_{D1}	CO_{D1}	RA_{D1}
D_{t2}	N_{D2}	CO_{D2}	RA_{D2}
D_{t3}	N_{D3}	CO_{D3}	RA_{D3}
...	...	...	...
D_{ti}	N_{Di}	CO_{Di}	RA_{Di}

cada dominio. Por tal motivo, se propone la generación de una tabla de composición de relaciones topológicas, la cual se encuentra ordenada con base en su relevancia o importancia dentro de algún dominio en particular (ver la tabla IV).

Para este caso, sea r_t una relación de la tabla de composición integrada por los conceptos c_i & $c_j \in C$ & $r_t \in R$, denotada por $r_t = c_i\, r_i\, c_j$.

Entonces, sea *N* el conjunto de relaciones necesarias presentes en un dominio geográfico *i*, en donde la presencia de estas relaciones en el dominio indican que estos conceptos siempre están ligados de esta forma e interactúan en un dominio en particular, de la forma: $N_{Di} = \{r_{tN1}, r_{tN2}, \dots, r_{tNn}\}$.

Para este caso, entonces sea *C* un conjunto de relaciones comunes presentes en un dominio geográfico *i*. La presencia de esta relaciones en el dominio nos indica que estos conceptos son comunes encontrarlos en este dominio en particular, de acuerdo con: $C_{Di} = \{r_{tC1}, r_{tC2}, \dots, r_{tCn}\}$.

Para el caso de las relaciones, sea *RA* un conjunto de relaciones "*raras*" presentes en un dominio geográfico. La presencia de estas relaciones en el dominio indican que es poco probable encontrarlas en ese dominio; sin embargo, es posible encontrarlas en algunas ocasiones y en algunos dominios. Por lo tanto, el conjunto se define como sigue: $RA_{Di} = \{R_{tRA1}, R_{tRA2}, \dots, R_{tRAn}\}$.

El dominio de la tabla de composición D_{ti} está compuesto por la unión de los conjuntos N_{Di}, CO_{Di} y RA_{Di} que se denota por: $D_{ti} = \{N_{Di} \cup CO_{Di} \cup RA_{Di}\}$.

Después de obtener la tabla de composición, se procede a ordenar los conceptos junto con sus relaciones topológicas, de acuerdo con su relevancia en el dominio. Esto indica que las relaciones del conjunto *N* son una parte fundamental para definir el dominio. En tanto, las relaciones comunes no lo son, por lo que éstas tendrán una relevancia menor en la definición del dominio, por lo que no es requisito contar con estas relaciones. De igual forma, las relaciones del conjunto *RA*, no son indispensables en la definición del dominio geográfico, además de que su relevancia está por debajo del conjunto *C* y además tiene menor relevancia que el conjunto *N*.

Para este caso, se define la función *relevant*, la cual recibe tres parámetros, una tripleta de conceptos y un dominio. Estos elementos describen el número de dominios y el número de conceptos en el caso de la tripleta, regresando como resultado la relación de relevancia de la tripleta de nuestro interés o falso en caso de no existir una relación de relevancia para el dominio determinado, lo cual se define de la siguiente forma: $relevant(d, c_a, c_b) = r_{tj} | 0 \leq j \leq m$.

TABLA V
TABLA DE REFINACIÓN SEMÁNTICA

Etiquetas de Entrada	Dominios de Salida
$E_1 = \{e_{11}, e_{12}, e_{13}, \dots, e_{1n}\}$	$Salida_1 = (D)$
$E_2 = \{e_{21}, e_{22}, e_{23}, \dots, e_{2n}\}$	$Salida_2 = (D)$
...	...
$E_m = \{e_{31}, e_{32}, e_{33}, \dots, e_{3n}\}$	$Salida_m = (D)$

Tabla de refinación semántica: Se define para almacenar primeramente las etiquetas que se reciben como entrada, con la finalidad de que se validen en algunos de los algoritmos propuestos en la *etapa de inferencia,* con el propósito de asignar el dominio al que pertenece, siempre y cuando exista una validación por parte del usuario (ver la tabla V).

De acuerdo con lo anterior, sea *return_ref* una función que recibe el conjunto etiquetas de entrada $E_i = \{e_{11}, e_{12}, \dots, e_{1n}\}$ que describen semánticamente ya sea uno o varios conjuntos de dominios en particular, regresando como resultado un conjunto de dominios de salida ordenados, de acuerdo con la cercanía definida por su *similitud semántica* con respecto a la descripción de las etiquetas de $Salida_i(D)$. Por tanto, la función se define de la siguiente manera: $return\_res(E_1) = Salida_i(D) | 0 \leq i \leq m$.

B. Etapa de inferencia

La etapa de *inferencia* está compuesta por cuatro tareas que interactúan de manera conjunta para analizar, describir y deducir a qué dominio pertenece un conjunto de objetos geográficos representados por etiquetas. La primera tarea consiste en establecer un método de mapeo, el cual recibe como entrada un conjunto de etiquetas. Con este conjunto se busca si existe un concepto definido en la ontología y que esté directamente relacionado con esas etiquetas. Posteriormente, con este mapeo se obtiene un conjunto de conceptos que están relacionados con la base de conocimiento definida en la ontología.

La segunda tarea consiste en la definición de un conjunto de *técnicas de razonamiento espacial cualitativo*, descrita por tres algoritmos definidos para tal fin, a saber: 1) algoritmo de frecuencias, 2) algoritmo de relevancia y 3) algoritmo de genealogía semántica. Con estos algoritmos se lleva a cabo el proceso de inferencia, tomando como entrada un conjunto de conceptos descritos en la base de conocimiento (ontología).

La tercera tarea consiste en definir un *selector de salida*, el cual se encarga de proporcionar la salida de uno de los algoritmos de razonamiento, de acuerdo con el grado de efectividad que se ha definido para cada uno. Este grado de efectividad varía con la interacción del usuario. Finalmente, la tarea de *refinación semántica* consiste en brindar una respuesta al usuario, con base en un proceso de validación, en donde la retroalimentación juega un papel preponderante para examinar la riqueza semántica de la representación. La inferencia obtenida se almacena en la base de conocimiento para formar parte de la conceptualización y determinar el grado de efectividad del algoritmo de razonamiento empleado. En la Figura 9 se muestra el marco de trabajo y los elementos que forman parte de la etapa de inferencia.

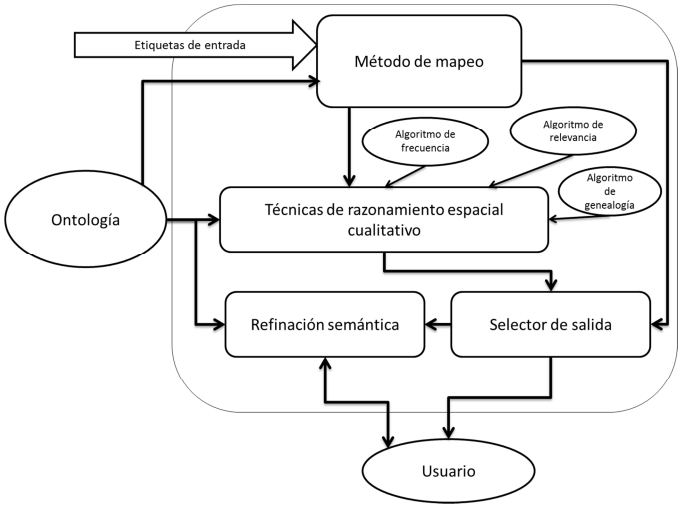


Fig. 9. Marco de trabajo de la etapa de inferencia.

Método de mapeo: El método de mapeo es un algoritmo diseñado para recibir las etiquetas de entrada y localizar si existen conceptos relacionados con esas etiquetas para almacenarlas en la ontología como *instancias* o *individuos* de un concepto clase (proceso de síntesis). Sin embargo, antes de esto es necesario verificar sí es que ya existe un resultado previo para este conjunto de etiquetas, utilizando la tabla de refinación semántica. Como salida de este método se obtiene un conjunto de entradas existentes dentro del conocimiento previo o bien el conjunto de dominios *D* ordenado. En caso de que las etiquetas ya hayan sido agregadas anteriormente y validadas por un usuario, se almacenan como parte de la conceptualización (ontología *O*). A continuación se presenta el algoritmo referente al método de mapeo propuesto (ver la tabla VI).

TABLA VI
ALGORITMO PROPUESTO PARA EL MÉTODO DE MAPEO

Entrada:
Se tiene como entrada un conjunto de etiquetas que representan objetos geográficos: $EtE = \{ete_1, ete_2, \dots, ete_n\}$, una tabla T_c de conceptos y una tabla de refinación semántica T_{re}.

Salida:
Un vector de mapeo *VM* conformado por una bandera llamada *SC* que toma dos valores posibles: falso o verdadero, un conjunto de conceptos de entrada $CE = \{ce_1, ce_2, \dots, ce_n\}$ cuando *SC* sea falso, o bien un conjunto de dominios ordenados $D = \{d_1, d_2, \dots, d_n\}$ cuando *SC* sea verdadero.

Variable $SC \leftarrow falso$
Si $return\_res(EtE)$
Entonces $SC \leftarrow verdadero$
Devuelve (SC,D)
Sino
Entonces
Por cada $c_i \in C$ $donde$ $i = 1:n$
Si $exist\_concept(c_i)$
Entonces $c_i \rightarrow ce_i$
Sino $exist\_synonymous(c_i)$
Entonces $c_i \rightarrow ce_i$
Fin de Ciclo
Devuelve (*SC*,*CE*)
Fin del Algoritmo

C. *Técnicas de razonamiento espacial cualitativo*

En este trabajo se proponen y describen tres algoritmos enfocados a realizar razonamiento espacial. El primero es un algoritmo de *frecuencia conceptual* que cuenta las ocurrencias de cada concepto en un dominio geográfico. El segundo es un algoritmo de *relevancia* que busca la relación que existe entre los conceptos que recibe como entrada y su importancia en el dominio geográfico y el tercero es un algoritmo de *genealogía semántica* que obtiene a los conceptos padre, calculando la jerarquía de conceptos que involucran a un objeto geográfico representado por un concepto en el dominio, con lo cual se asume que este método podría utilizarse para proporcionar una definición del contexto.

Algoritmo de frecuencia conceptual: La finalidad de este algoritmo radica en contar el número de veces o calcular la frecuencia con la que aparece un conjunto de conceptos en un determinado dominio, a lo que se le ha denominado como *frecuencia conceptual*.

Asimismo, la salida se ordena de mayor a menor de acuerdo con su repetición. Este algoritmo recibe un conjunto de conceptos que previamente se han verificado, por tanto, existen en la base del conocimiento (esta verificación se realiza por medio del algoritmo de mapeo).

Posteriormente, es necesario utilizar la tabla de frecuencias de conceptos en los dominios, con la finalidad de buscar cuantas veces aparece cada concepto en cada dominio geográfico que tiene la base de conocimiento. En la Tabla 7 se presenta el algoritmo de frecuencia conceptual.

TABLA VII
ALGORITMO DE FRECUENCIA CONCEPTUAL

Entrada:
Un conjunto de conceptos que existen en la ontología $CE = \{ce_1 ce_2, \dots, ce_m\}$, una tabla Tf de frecuencias de conceptos en dominios y un vector de mapeo VM.

Salida:
Un conjunto de dominios ordenados $D = \{d_1, d_2, \dots, d_n\}$
Variables: $frecuenciadelDominio = \{fd_1, fd_2, \dots, fd_n\}$
Inicializar el conjunto de variables frecuencia de dominio en 0
Si *SC* es falso
Entonces
Por cada $d_i \in D, donde\ i = 1:n$
Por cada $ce_j \in CE,\ donde\ j = 1:m$
Si $concept(d_i, ce_j)$
Entonces incrementa fd_i
Fin de ciclo
Fin de ciclo
Ordena resultados $frecuenciadeldominio = \{fd_{d1}, fd_{d2}, \dots, fd_{dN}\}$
Sino entonces salir
Fin del Algoritmo

Algoritmo de relevancia: El algoritmo de relevancia recibe como parámetro un conjunto de entradas existentes del algoritmo de mapeo y un conjunto de tablas de composición de relevancia en el dominio. Por tanto, como salida se obtiene un conjunto de dominios ordenados, de acuerdo con su relevancia.

TABLA VIII
ALGORITMO DE RELEVANCIA

Entrada:
Un conjunto de conceptos que existen en la ontología $CE = \{ce_1 ce_2, \dots, ce_m\}$, una Tc de composición y un vector de mapeo VM
Salida:
Un conjunto de dominios ordenados $D = \{d_1, d_2, \dots, d_n\}$, junto con un conjunto de etiquetas de relevancia $ERD = \{ ERd_1, ERd_2, \dots, ERd_i\}$ para cada dominio $ERd_i = \{N_{Dti}, CO_{Dti}, RA_{Dti}\}, donde\ i = 1:n$
Variables: Un conjunto de etiquetas de relevancia $ERD = \{ ERd_1, ERd_2, \dots, ERd_i\}$ para cada dominio.
Si *SC* es falso
Entonces
Por cada $d_i \in D,\ donde\ i = 1:n$
Por cada $ce_j \in CE, donde\ j = 1:m$
Por cada $ce_k \in CE, donde\ k = 1:m$
Si $relevant(d_i, ce_j, ce_k)$
Entonces *incrementa la etiqueta correspondiente a* $ERd_i = \{N_{Dti}, CO_{Dti}, RA_{Dti}\}$
Fin de ciclo
Fin de ciclo
Fin de ciclo
Ordena resultados de ERD
Sino entonces salir
Fin del Algoritmo

La utilidad de este algoritmo es que permite clasificar el grado de *importancia* de los conceptos en un dominio, y así proporcionar mayor riqueza semántica a los conceptos en cada dominio en particular. El algoritmo de relevancia realiza una búsqueda de todos los conceptos que recibe como entrada en la tabla de composición de relevancia de cada dominio, con el fin de saber si esos conceptos están relacionados. Finalmente, se obtiene como salida un conjunto de dominios. En la tabla VII se describe el algoritmo de relevancia para aplicar razonamiento espacial a un conjunto de objetos geográficos.

Algoritmo de genealogía semántica: Las operaciones que realiza el algoritmo de genealogía semántica se dividen en tres tareas. La primera es buscar los conceptos padres e hijos de cada uno de los conceptos de entrada. La segunda consiste en utilizar el algoritmo de relevancia para obtener el dominio resultante (solo se tomará el primer resultado obtenido por el algoritmo de relevancia). La última tarea consiste en realizar una suma de las salidas para mostrar el dominio que haya aparecido más con las entradas dadas al algoritmo de relevancia. La salida del algoritmo de genealogía, puede ser un dominio o bien la salida puede ser un conjunto vacío, lo cual indica que no se encontró ningún dominio.

El proceso del algoritmo de genealogía es el siguiente: primero, si el vector de mapeo *VM* es falso, entonces recibe los conceptos de entrada. Luego, busca la clase padre de cada uno de los *n-conceptos* de entrada y sustituye el *i-ésimo* concepto por el concepto padre, para invocar el algoritmo de relevancia con ese nuevo conjunto de entradas. Del resultado obtenido por el algoritmo de relevancia se guarda solo el primer elemento obtenido en un vector de salida, el cual contiene el dominio resultante de las veces que se repite dicho dominio. Este proceso es iterativo y se repite de acuerdo con el número de conceptos padres que se hayan encontrado.

TABLA IX
ALGORITMO DE GENEALOGÍA SEMÁNTICA

Entrada:
Un conjunto de conceptos que existen en la ontología $CE = \{ce_1 ce_2, \dots, ce_m\}$ y un vector de mapeo VM

Salida:
Un conjunto de dominios ordenados $D = \{d_1, d_2, \dots, d_l\}$
Si *SC* es falso
Variables: Una copia temporal CET de los conceptos de entrada CE, Un conjunto de dominios $Dominios = \{d_1, d_2, \dots, d_n\}$.
Entonces
Por cada $i \in \mathbb{N}, donde\ i = 1:n$
$CE \rightarrow CET$
Si $father(cet_i) \neq 0$
Entonces $father(cet_i) \rightarrow cet_i$
Algoritmo de relevancia(**CET**)
$d_1 \rightarrow Dominios$
Fin de ciclo
Por cada $son(ce_i)$
$CE \rightarrow CET$
Si $son(cet_i) \neq 0$
Entonces $son(cet_i) \rightarrow cet_i$
Algoritmo de relevancia(**CET**)
$d_1 \rightarrow Dominios$
Fin de ciclo
Eliminar_Repetidos($Dominios$)
Sino entonces salir
Fin del Algoritmo

TABLA X
ALGORITMO ELIMINAR CONCEPTOS REPETIDOS

Entrada:
Un conjunto de dominios $Dominios = \{d_1, d_2, \dots, d_n\}$.
Salida:
Un conjunto de dominios ordenados $D = \{d_1, d_2, \dots, d_l\}$
Variables: Un conjunto de vectores de salida dominios conformado por el nombre del dominio sd y las ocurrencias en el dominios osd
$SD = \{< sd_1, osd_1 >, < sd_1, osd_1 >, \dots., < sd_n, osd_n >\}$
Por cada $d_i \in Dominios, donde\ i = 1:n$
Por cada $d_j \in Dominios, donde\ j = 1:m$
Si $dj \neq nulo$
Si $d_i = dj\ \&\ i \neq j$
Entonces $dj \rightarrow nulo$
Por cada sd_i
Si $d_i = sd_i$
Entonces incrementa osd_i una unidad
Sino Entonces $di \rightarrow sdi$,
incrementa osd_i una unidad
Fin de ciclo
Fin de ciclo
Fin de ciclo
Ordena(SD)
Fin del Algoritmo

El proceso para los conceptos hijos de cada concepto de entrada es similar, con la diferencia de que cada concepto de entrada puede tener un número *m* de conceptos hijos; por lo que cada concepto hijo será una entrada individual al algoritmo de frecuencia. Este algoritmo se describe en la Tabla 9, así como el algoritmo que representa una función para eliminar conceptos repetidos en la tabla X.

D. Selector de salida

El algoritmo selector de salida recibe los resultados del conjunto de algoritmos y define cual es el resultado de

razonamiento más apropiado, de acuerdo con el grado de éxito de cada algoritmo de razonamiento implementado.

La efectividad o éxito del algoritmo depende directamente de las interacciones que se realicen con el usuario, ya que es necesario validar este conocimiento de una manera cognitiva para que pueda servir como un mecanismo de retroalimentación a la base de conocimiento y se tenga mayor *granularidad semántica*.

La función principal de este mecanismo es la comparación entre un grupo de tres resultados, si los tres conjuntos de resultados son iguales, se desplegará cualquiera de los conjuntos, si encuentra al menos dos resultados iguales de los tres que recibió, se mostrarán los resultados que fueron iguales. En caso de que ninguno de los resultados fue igual se visualiza el que tenga mayor grado de efectividad. Este procedimiento es descrito en el algoritmo mostrado en la Tabla 11.

TABLA XI
ALGORITMO SELECTOR DE SALIDA

Entrada:
Un vector de mapeo *VM*, Un conjunto de dominios ordenados por cada algoritmo de razonamiento $CAr = \{Ar_1, Ar_2, \dots, Ar_n\}$
Salida:
Un vector de salida $VS = \{A, CARS\}$ con los siguientes datos. Un conjunto de dominios ordenados $A = \{D_1, D_2, \dots, D_n\}$, y el conjunto de algoritmos de razonamiento espacial cualitativo de salida utilizados en la forma $CARS = \{A_1, A_2. A_3, salidaConocida\}$
Variables: Tres conjuntos A_1, A_2, A_3 que denotan a los algoritmos con mayor grado de éxito de la forma $A = \{D_1, D_2, \dots, D_n\}$
Si $SC\ es\ verdadero$
Entonces entrega el conjunto $\{(D), (salidaConocida)\}$ del vector de mapeo *VM* como salida.
Sino
Entonces $exito(CAr) \rightarrow \{A_1, A_2, A_3\}$
Si $compara(A_1, A_2)\ es\ verdadero$
& $compara(A_1, A_3)\ es\ verdadero$
Entonces despliega $\{(A_1), (A_1, A_2, A_3)\}$
Si $compara(A_1, A_2)\ es\ falso$ &
$compara(A_1, A_3)\ es\ verdadero$
Entonces despliega $\{(A_1), (A_1, A_3)\}$
Si $compara(A_1, A_2)\ es\ verdadero$ &
$compara(A_1, A_3)\ es\ falso$
Entonces despliega $\{(A_1), (A_1, A_2)\}$
Si $compara(A_1, A_2)\ es\ falso$ &
$compara(A_1, A_3)\ es\ falso$
Entonces
  Si $compara(A_2, A_3)\ es\ verdadero$
  Entonces despliega $\{(A_2), (A_2)\}$
  Sino
  Entonces despliega $\{(A_1), (A_1)\}$
Fin del Algoritmo

Inicio Función Compara
Entrada:
Un par de conjuntos ordenados $A_1\ y\ A_2$
Salida:
verdadero en caso de que sean iguales o *falso* en caso de que sean diferentes
Por cada $DA1_i \in DA1, donde\ i = 1{:}n$
  Si $DA1_i \neq DA2_i$
  Entonces devuelve falso
Fin de ciclo
devuelve verdadero
Fin de función compara

IV. RESULTADOS OBTENIDOS

En la presente sección se presentan los resultados obtenidos de acuerdo con cada una de las etapas definidas en este trabajo.

A. Arquitectura del sistema

La arquitectura del sistema implementado se muestra en la Figura 10 y está compuesta de tres partes. La primera es el repositorio donde se procesan los marcos conceptuales. La segunda realiza la carga del modelo persistente de base de datos construido a partir de la ontología.

Finalmente, se encuentra la entrada del sistema que se relaciona directamente con la etapa de inferencia. Esta etapa se encarga de consultar la información del modelo persistente para devolver como resultado un concepto general al dominio en cuestión o al contexto geográfico.

B. Marcos conceptuales

Para representar el conocimiento y que éste pueda ser leído por la máquina, sin que todo el conocimiento tenga que ser ingresado de una sola vez, se selecciona una estructura básica.

Para este caso, se ha elegido el uso de los marcos conceptuales [43], [44]. Para su implementación, se eligió utilizar el metalenguaje XML [45], ya que de acuerdo con su estructura se puede realizar una apropiada clasificación taxonómica y organización del conocimiento *a priori*. Los marcos conceptuales están clasificados en dos tipos: *particulares* y *generales*.

Los marcos *particulares* (ver Figura 11) contienen la información de todos los conceptos que integran directamente y representan un dominio geográfico en particular, con datos referentes a su nombre, sinónimos, concepto padre e hijo relacionado. Los marcos *generales* (ver Figura 12) contienen la definición del dominio dado por las relaciones topológicas existentes entre los conceptos que componen al dominio, así como sus sinónimos. Este tipo de marcos definen un contexto para los conceptos y sinónimos utilizados.

A. Diseño del modelo persistente basado en la ontología

El modelo persistente ha sido construido a partir de la ontología. Con el propósito de almacenar las instancias de la base de conocimiento se ha convertido este modelo a una base

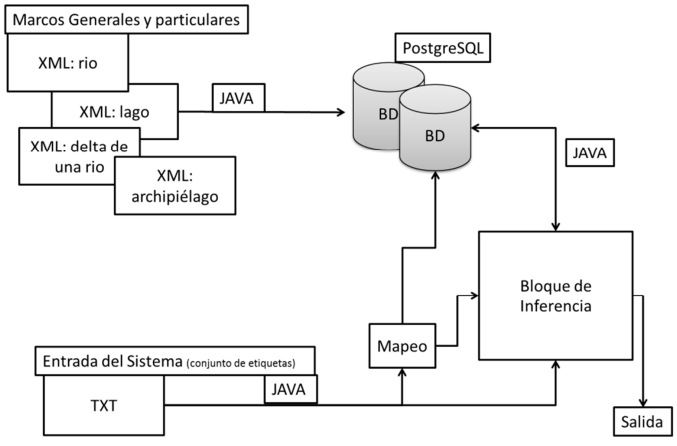


Fig. 10. Arquitectura del sistema.

```
<?xml version="1.0"?>
<particular>
<concepto>
        <nombre>rio</nombre>
        <sinonimos>river,flujo,flujo de agua</sinonimos>
        <propiedades>anamosado,meandrico,rectilineo</propiedades>
        <padre>caudal</padre>
        <hijos>perene,estacionario,transitorio</hijos>
        <relacion_topologica>
                <cruza>area verde,tierra<cruza>
                <conecta>lago,mar</conecta>
                <comparte>isla</comparte>
        <relacion_topologica>
</concepto>
<concepto>
        <nombre>lago</nombre>
        .
```

Fig. 11. Marco conceptual particular.

```
<?xml version="1.0"?>
<general>
<concepto>
        <nombre>delta de un rio</nombre>
        <sinonimos>river,flujo,flujo de agua</sinonimos>
        <conceptos_que_lo_componen>rio,lago, area verde,tierra,mar,
        cuerpo de agua,isla,arena</conceptos_que_lo_componen>
        <definiciones>
                <necesario>rio,conecta,mar</necesario>
                <comun>isla,comparte,mar</comun>
                <raro>arena,comparte,mar</raro>
        </definiciones>
</concepto>
<concepto>
        <nombre>costa</nombre>
        .
```

Fig. 12. Marco conceptual general.

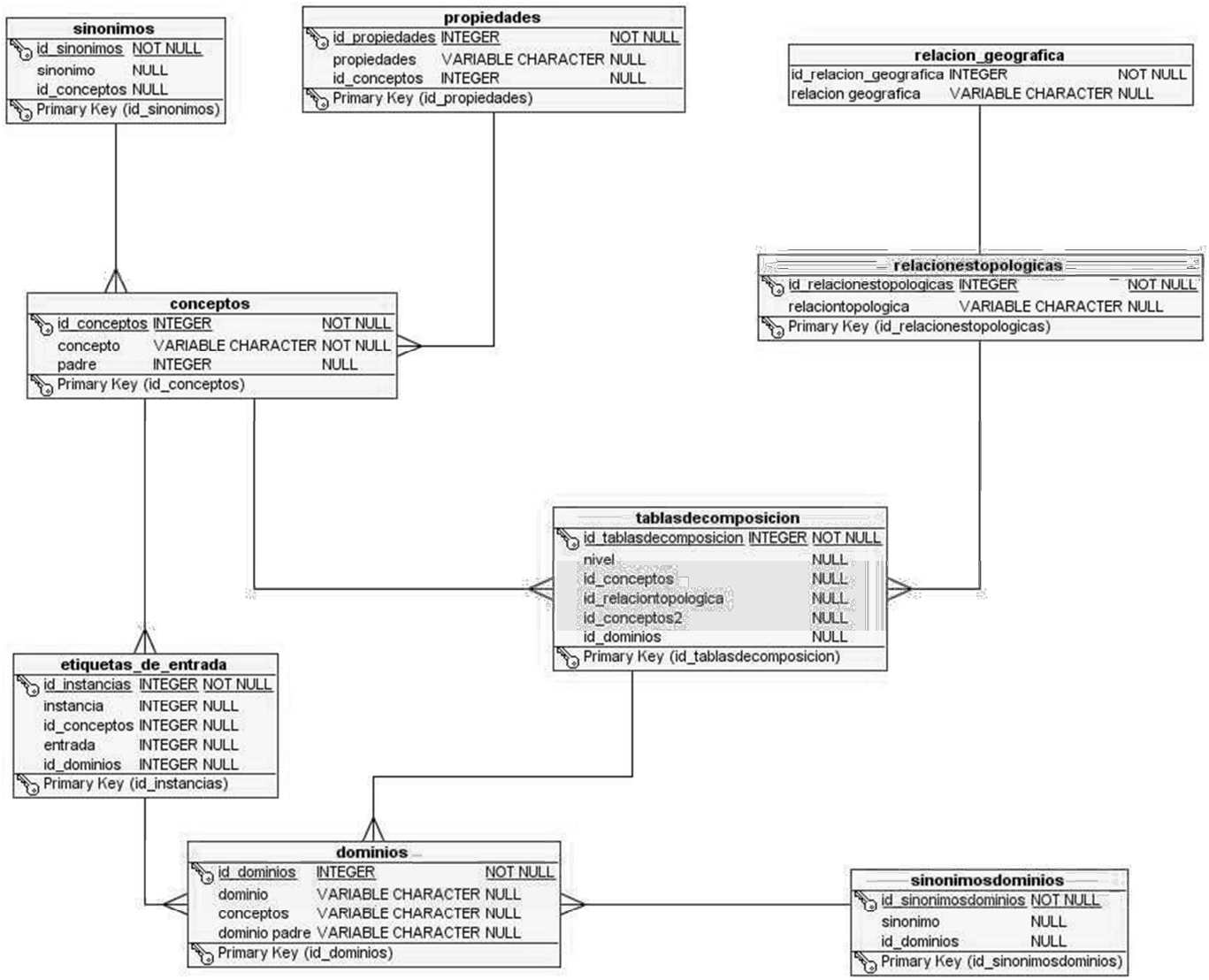


Fig. 13. Modelo persistente generado a partir de la ontología RAIN.

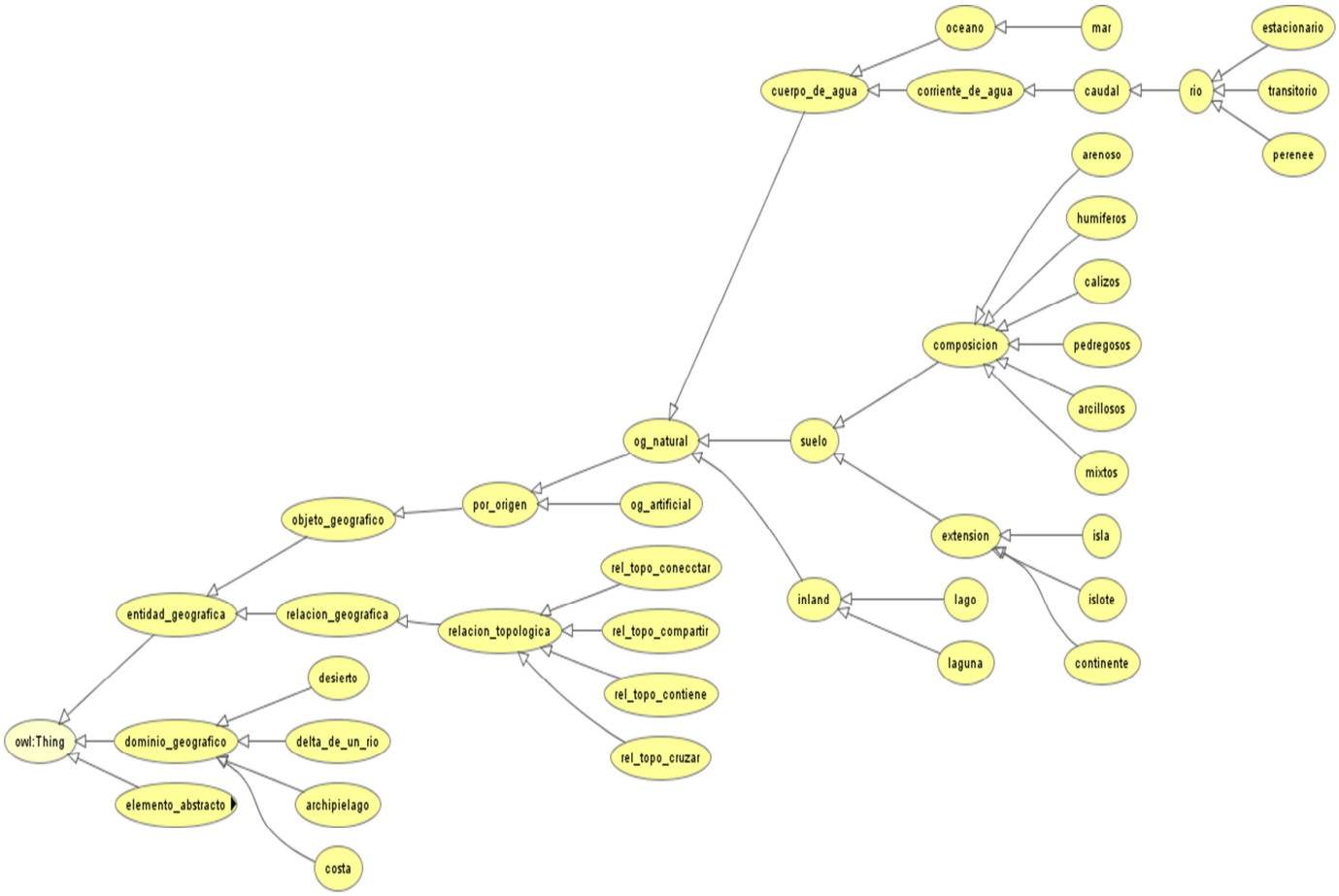


Fig. 14. Ontología RAIN.

de datos, la cual está compuesta por nueve tablas que están diseñadas para recibir la información obtenida en el bloque de análisis, a través de las estructuras de los marcos conceptuales.

Entre las tablas más importantes se tienen la tabla de conceptos que contiene información tal como el nombre del concepto, su ubicación en la jerarquía (sus padres e hijos); la tabla de sinónimos que guarda información acerca de los nombres conocidos del concepto, la tabla de propiedades que almacena las características de cada concepto, la tabla de composición que contiene información acerca de la relación topológica que existe entre los conceptos, el dominio al que pertenece y el nivel de relevancia que tiene dentro del mismo dominio; es decir, si esa relación es necesaria para la definición del dominio, común o es rara en el mismo. Esta tabla se apoya en la tabla de conceptos, relaciones topológicas y dominios.

Por último, se tiene la tabla de etiquetas de entrada que contiene información acerca de las etiquetas que recibe la etapa de inferencia, para que así estas etiquetas sean relacionadas a través del algoritmo de mapeo con un concepto existente en la tabla de conceptos.

Esta tabla recibe soporte de la tablas de dominios y conceptos. El modelo persistente de la base de datos se muestra en la Figura 13.

La información contenida en el modelo persistente está basada en la ontología RAIN (ver Figura 14), la cual contiene la definición de los objetos geográficos y las relaciones topológicas que intervienen en los dominios geográficos y está basada en la ontología de dominio *Kaab*, descrita en [40].

B. Aplicación enfocada al razonamiento espacial

La aplicación está compuesta de cuatro bloques. El primero (referido en la Figura 15 con el número "1") señala la entrada del sistema para ingresar el conocimiento previo, utilizando el formato de los marcos conceptuales. El segundo bloque (señalado con el número "2") es la entrada del sistema, donde se ingresa un conjunto de etiquetas de entrada (se observa el botón "archivo separado por comas" para acceder a la entrada, mediante un archivo de texto), con la finalidad de conocer a que dominio geográfico pertenecen. Además este bloque contiene los botones de "frecuencia conceptual", "relevancia" y "genealogía semántica" que corresponden a los diferentes algoritmo de razonamiento diseñados y que se pueden emplear para definir el dominio al que se refieren las etiquetas que representan a los objetos geográficos. Por otra parte, el botón de "analizador de resultados" ejecuta los tres algoritmos de razonamiento. El tercer bloque muestra las operaciones que se están realizando sobre la base de datos. Por último, el cuarto bloque presenta los resultados del sistema, dependiendo del tipo de algoritmo que se haya seleccionado.

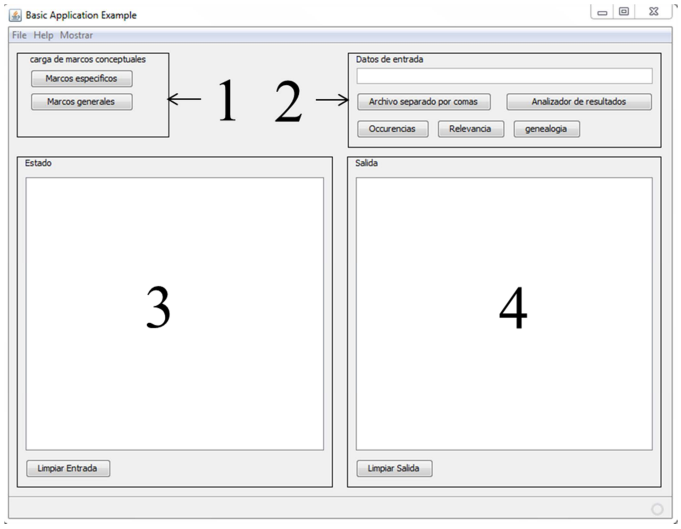


Fig. 15. Aplicación enfocada al razonamiento espacial.

C. *Pruebas realizadas a los algoritmos de razonamiento*

A continuación se muestran los resultados obtenidos con el algoritmo de razonamiento basado en frecuencia conceptual, utilizando los dominios geográficos *delta de un río* y *costa*. Primeramente, se inicia con la carga de los marcos conceptuales particulares y generales a la base de datos. En la tabla XII, se pueden observar las frecuencias en los dominios para este algoritmo.

TABLA XII
TABLA FRECUENCIAS DE CONCEPTOS EN DOMINIOS

Dominio	Algoritmo selector de salida
Delta de un río	*{río, lago, área verde, tierra, mar, cuerpo de agua, isla, arena}*
Costa	*{mar, arena, río}*

Por lo tanto, las etiquetas de entrada son: `<mar, arena, río, isla, área verde>`. La salida que se obtuvo fue *"delta de un río"*, debido a que fue el dominio que más ocurrencias tuvo y en segundo lugar *"costa"*, ya que solo se encontraron tres conceptos en ese dominio.

En la Figura 16, se muestra la salida de la aplicación, donde se puede observar que el algoritmo de frecuencia conceptual contabiliza el número de ocurrencias de cada dominio por concepto, directamente de la representación conceptual.

Sin embargo, se presenta un problema que está asociado directamente entre estos dos dominios. Esto es que para el caso en que la entrada de información se introducen etiquetas que se encuentran en ambos dominios, se obtendría la misma salida (por ejemplo, *mar* y *arena*). Para resolver este conflicto, se hace una distinción entre las etiquetas, tomando en consideración su grado de importancia dentro de un dominio. Para ello, se utiliza el algoritmo de relevancia, en donde es necesario contar con la definición de los conceptos con base en sus relaciones topológicas que intrínsecamente poseen, lo cual es descrito de manera explícita en la tabla XIII.

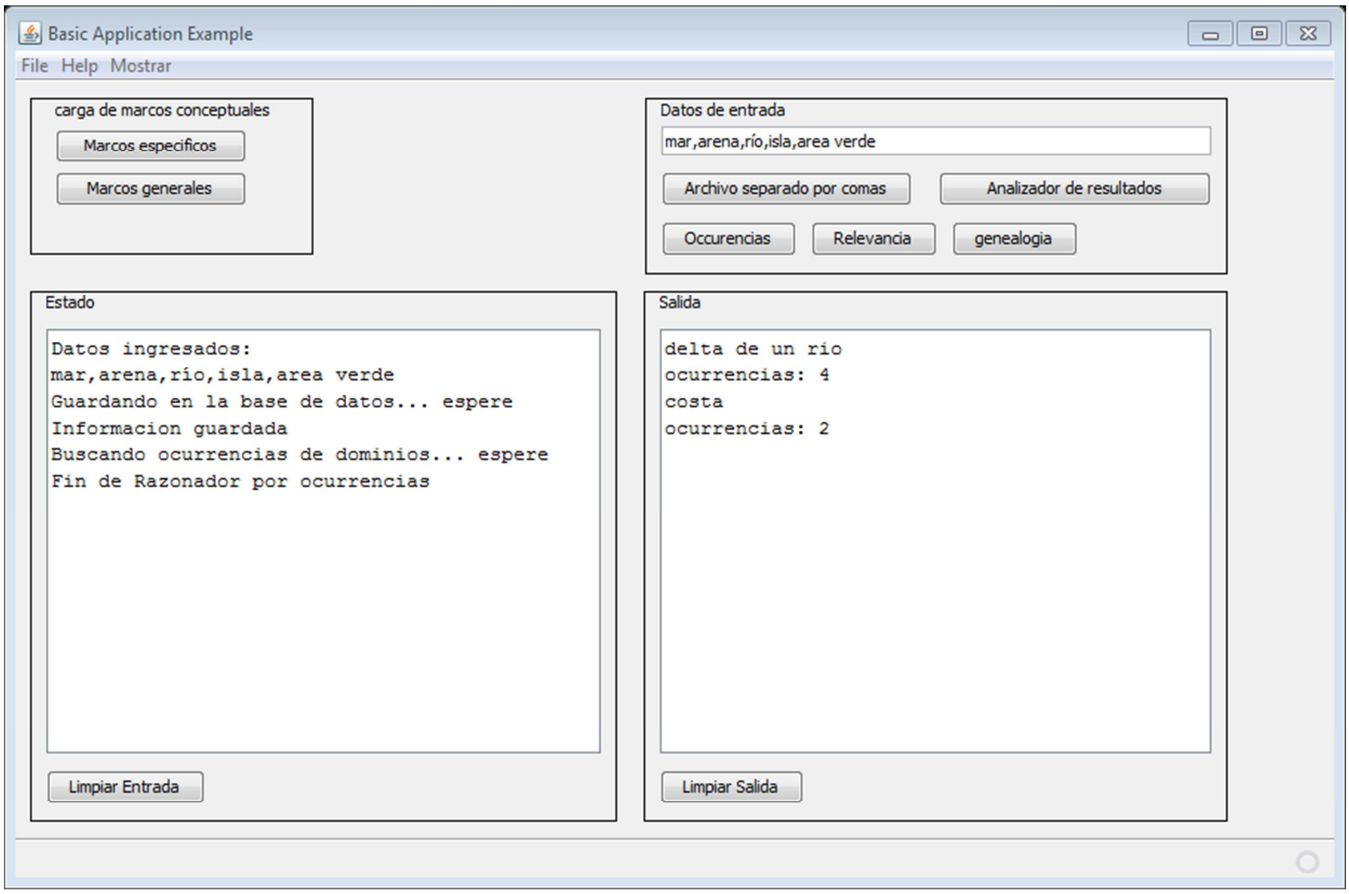


Fig. 16. Salida del algoritmo de frecuencia conceptual.

TABLA XIII
TABLA DE COMPOSICIÓN DE RELACIONES TOPOLÓGICAS

Dominio	Concepto1	Relación Topológica	Concepto2	Nivel de Relevancia
Delta de un río	Río	Conecta	Mar	Necesario
Delta de un río	Isla	Comparte	Mar	Común
Delta de un río	Arena	Comparte	Mar	Raro
Costa	Arena	Comparte	Mar	Necesario
Costa	Río	Cruza	Área verde	Común
Costa	Río	Conecta	Mar	Común

Con base en la tabla anterior, ahora se puede observar que cuando se introduce la entrada *"mar"* y *"arena"* y además se utiliza el algoritmo de relevancia, se obtiene en primer lugar *"costa"* y en segundo *"delta de un río"*. Esto se debe a que en la definición de cada uno de los dominios, ambos conceptos existen; pero solo en *"costa"* se presenta una relación necesaria que está definida por *"arena comparte mar"*, mientras que la relevancia en el dominio *"delta de un río"* es menor, ya que es raro encontrar este tipo de relación en ese dominio o contexto. En la figura 17, se muestra el resultado de aplicar el algoritmo de relevancia.

Por otra parte, cuando se escriben conceptos que no están explícitamente en las definiciones de los dominios; por ejemplo, sea el caso de la entrada `<perenne, mar>`, donde "perenne" pertenece a una subclase de "río". Entonces, cuando esto sucede es necesario utilizar el algoritmo de *genealogía semántica* para buscar al padre de este concepto y ejecutar posteriormente el *algoritmo de relevancia,* con lo cual se debe almacenar el resultado. Consecuentemente, se deben buscar las clases hijos y verificar si existen definiciones. Como último paso, se repetirá este procedimiento para el concepto "mar". En la figura 18 se puede observar la salida de este algoritmo.

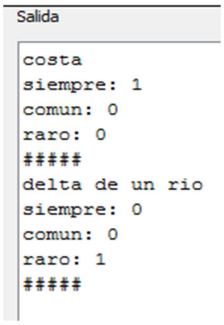


Fig. 17. Resultado de aplicar el algoritmo de relevancia.

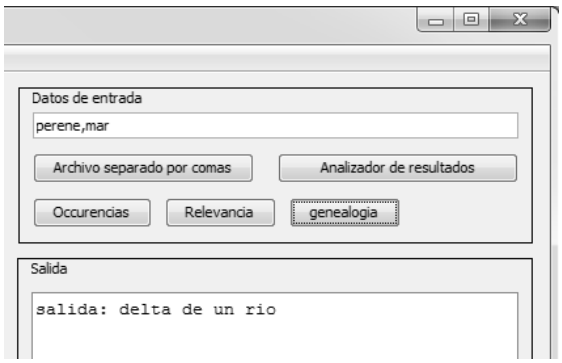


Fig. 18. Salida del algoritmo de genealogía semántica.

V. CONCLUSIONES

En este trabajo, se presenta un método compuesto básicamente por tres algoritmos que realizan un proceso de inferencia, con los cuales se lleva a cabo un proceso de razonamiento espacial cualitativo en descripciones de objetos geográficos.

Para ello, se genera una base de conocimiento, por medio de una ontología de aplicación que ha sido construida en OWL, bajo la metodología de GEONTO-MET. Esta base de conocimiento representa el conocimiento *a priori* de diferentes dominios geográficos.

Asimismo, se propone el uso de marcos conceptuales para representar de forma explícita el conocimiento de algún dominio, con la finalidad de estructurar la semántica del contexto geográfico.

Por otra parte, se puede argumentar que todo algoritmo de razonamiento espacial cualitativo, está compuesto por dos bloques: un bloque asociado con la *búsqueda* y otro relacionado con el *ordenamiento* basado en relevancia.

Asimismo, podemos afirmar que el conocimiento previo es una estructura formal, la cual contiene el vocabulario, las reglas para el lenguaje y un conjunto de proposiciones que son verdaderas (ya sean hechos o restricciones) que permiten solucionar problemas de ambigüedad y vaguedad, fundamentalmente para datos geoespaciales que por su naturaleza presentan estas deficiencias. Además, se propone una definición de razonamiento espacial, la cual consiste en el proceso de transformar una representación descriptiva en otra más general con base en el uso y procesamiento semántico de las relaciones topológicas. Cabe señalar que el proceso que se realiza en este trabajo es intentar *generalizar* hacia una clase o concepto superior, en un sentido inverso a la granularidad semántica (particularidad de objetos geográficos) que puede tener una representación conceptual, con lo cual se busca que la máquina procese las entidades geográficas de una manera similar a como los seres humanos procesamos cognitivamente e interpretamos el mundo real.

AGRADECIMIENTOS

Este trabajo ha sido auspiciado y apoyado por el Instituto Politécnico Nacional (IPN), el Consejo Nacional de Ciencia y Tecnología (CONACYT) y la Secretaría de Investigación y Posgrado del IPN, a través de los proyectos: 20113712, 20113757, 20120563 y 20120482.

REFERENCIAS

[1] G. A. Elmes et al., "GIS and Society: Interrelation, Integration, and Transformation," *A Research Agenda for Geographic Information Science*, vol. 3, CRC, pp. 287, 2005.

[2] J. Hobbs, J. Blythe, H. Chalupsky, and T.A. Russ, *A Survey of Geospatial Resources, Representation and Reasoning*. Public Distribution of the University of Southern California, 2006.

[3] M. Donnelly, T. Bittner, and C. Rosse, "A formal theory for spatial representation and reasoning in biomedical ontologies," *Artificial Intelligence in Medicine*. vol. 36, Elsevier, pp. 1-27, 2006.

[4] D. Hernandez, and A. Mukerjee, "Representation of spatial knowledge," *Information Systems Magazine*, 1995.

[5] J. Renz, *Qualitative spatial reasoning with topological information*. New York: Springer-Verlag, ch. 1, 2002.

[6] D. M. Mark, "Geographic Information Science: Defining the field," *Foundations of Geographic Information Science*, New York: Taylor and Francis, pp. 3-18, 2003.

[7] M. Egenhofer, and D. M. Mark, "Naive Geography," *Spatial Information Theory a Theoretical Basis for GIS*, New York: Springer-Verlag, 1995, pp. 1-15.

[8] J. Sharma, *Integrated spatial reasoning in geographic information systems: combining topology and direction*, University of Maine, 1996.

[9] B. Smith, "Mereotopology: a theory of parts and boundaries," *Data & Knowledge Engineering*, vol. 20, no. 3, Elsevier, pp. 287-303, 1996.

[10] L. Jiming and L. K. Daneshemend. *Spatial reasoning and planning: geometry, mechanism, and motion,* New York: Springer-Verlag, ch. 1, 2004.

[11] B. Bennett, "Physical objects, identity and vagueness," *Principles of Knowledge Representation and Reasoning International Conference*, Morgan Kaufmann, pp. 395-408, 2002.

[12] B. Bennett, *Logical representations for automated reasoning about spatial relationships*, University of Leeds, 1997.

[13] C. Freksa, "Qualitative spatial reasoning," *Cognitive and Linguistic Aspects of Geographic Space*, vol. 63, Springer-Verlag, pp. 361-372, 1991.

[14] A.G. Cohn, B. Bennett, J. Gooday, and N. M. Gotts, "Qualitative spatial representation and reasoning with the region connection calculus," *GeoInformatica*, vol. 3, no. 3, Springer-Verlag, 1997, pp. 275-316.

[15] A.G. Cohn and S. M. Hazarika, "Qualitative spatial representation and reasoning: An overview," *Fundamenta Informaticae*, vol. 46, no. 1-2, IO Press, pp. 1-29, 2001.

[16] L. Clarke, "A calculus of individuals based on connection," *Notre Dame Journal of Formal Logic*, vol. 22, no. 3, pp. 204-218, 1981.

[17] J. F. Allen, "Maintaining knowledge about temporal intervals," *Communications of the ACM*, vol. 26, no. 11, ACM, pp. 832-843, 1983.

[18] D. Papadias and M. J. Egenhofer, "Algorithms for hierarchical spatial reasoning," *GeoInformatica*, vol. 1, no. 3, Springer, pp. 1-23, 1997.

[19] M. J. Egenhofer and K. Al-Taha, "Reasoning about gradual changes of topological relationships," *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, New York: Springer-Verlag, pp. 196-219, 1992.

[20] M. Egenhofer and R. Franzosa, "Point-set topological spatial relations," *International Journal of Geographical Information Science*, vol. 5, no. 2, Taylor & Francis, pp. 161-174, 1991.

[21] D. A. Randell, Z. Cui, and A. G. Cohn, "A spatial logic based on regions and connection," in *Third International Conference on Knowledge Representation and Reasoning*, vol. 92, Morgan Kaufmann, pp. 165-176, 1992.

[22] J. R. Hobbs and S. Narayanan, "Spatial representation and reasoning," *Encyclopedia of Cognitive Science*, John Wiley, Online Library, 2001.

[23] El-Geresy and A. I. Abdelmoty, "SPARQS: a qualitative spatial reasoning engine," *Knowledge-Based Systems*, vol. 17, no. 2, Elsevier, pp. 89-102, 2004.

[24] R. Grütter, B. Bauer-Messmer, and M. Hägeli, "Extending an ontology-based search with a formalism for spatial reasoning," *Proceedings of the 2008 ACM Symposium on Applied Computing*, ACM, pp. 2266-2270, 2008.

[25] S. Wang, D Liu, X. Wang, and J. Liu, "Spatial reasoning based spatial data mining for precision agriculture," *Advanced Web and Network Technologies, and Applications*, New York: Springer-Verlag, pp. 506-510, 2006.

[26] F. J. Escobar, S. Eagleson, and I. P. Williamson, "Automating the Administration Boundary Design Process using Hierarchical Spatial Reasoning Theory and Geographic Information Systems," *International Journal of Geographical Information Science*, vol. 17, no. 2, Taylor & Francis, pp. 99-118, 2003.

[27] S. Wiebrock, L. Wittenburg, U. Schmid, and F. Wysotzki, "Inference and visualization of spatial relations," *Spatial Cognition II*, Springer-Verlag, pp.212-214, 2000.

[28] E. Clementini, "Directional relations and frames of reference," *GeoInformatica*, Springer-Verlag, pp. 1-21, 2011.

[29] Bailey-Kellogg and F. Zhao, "Qualitative spatial reasoning extracting and reasoning with spatial aggregates," *Artificial Intelligence Magazine*, vol. 24, no. 4, pp. 47-60, 2003.

[30] P. L. Schultz, H. W. Guesgen, and R. Amor, "Computer-human interaction issues when integrating qualitative spatial reasoning into geographic information systems," in *Proceedings of the 7th ACM SIGCHI, New Zealand Chapter's International Conference on Computer-Human Interaction: Design Centered HCI*, ACM, pp. 43-51, 2006.

[31] J. Brennan, and A. Sowmya, "Satellite image interpretation using spatial reasoning," in *Australasian Remote Sensing Photogrammet. Conf.,* Sydney Australia, vol. 1, 1998.

[32] T. Barkowsky, S. Bertel, D. Engel, and C. Freksa, "Design of an architecture for reasoning with mental images," in *International Workshop on Spatial and Visual Components in Mental Reasoning about Large-Scale Spaces*, pp. 01-02, 2003.

[33] B. Bennett, "What is a forest? On the vagueness of certain geographic concepts," *Topoi*, Springer-Verlag, vol. 20, no. 2, pp. 189-201, 2003.

[34] G. Câmara, M. J. Egenhofer, F. Fonseca, and A. M. Vieira, "What's in an Image ?" in *Conference on Spatial Information Theory*, New York: Springer-Verlag, pp.474-488, 2001.

[35] M. Aiello, C. Areces, and M. D. Rijke, "Spatial reasoning for image retrieval," *International Workshop on Description Logics*, vol. 22, New York: Springer-Verlag, 1999.

[36] Mezaris, I. Kompatsiaris, and M. G. Strintzis, "An ontology approach to object-based image retrieval," *International Conference on Image Processing*, vol. 2, IEEE, pp. 2-51, 2003.

[37] L. Hollink, G. Schreiber, J. Wielemaker, and B. Wielinga, "Semantic annotation of image collections," *Knowledge Capture*, pp. 41-48, 2003.

[38] J. Inglada and J. Michel, "Qualitative spatial reasoning for high-resolution remote sensing image analysis," *Geoscience and Remote Sensing, IEEE Transactions*, vol. 47, no. 2, 2009, pp. 599-612.

[39] M. Torres and S. Levachkine, "Representación ontológica basada en descriptores semánticos aplicada a objetos geográficos," *Computación y Sistemas*, vol. 12, no.3, pp.356-371, 2009.

[40] M. Torres, R. Quintero, M. Moreno-Ibarra, R. Menchaca-Méndez, and G. Guzmán, "GEONTO-MET: An approach to conceptualizing the geographic domain," *International Journal of Geographical Information Science,* vol. 25, no. 10, Taylor & Francis, pp. 1633-1657, 2011.

[41] Instituto Nacional de Estadística y Geografía, *Diccionario de Datos del INEGI*, INEGI, 2011.

[42] National Center for Biomedical Ontology, *Ontology of BIOMES,* 2011.

[43] M. Minsky, "K-lines: A Theory of Memory," *Cognitive Science*, vol. 4, no. 2, Elsevier, pp. 117-133, 1980.

[44] M. Minsky, *A framework for representing knowledge*, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Memo No. 306, June 1974.

[45] A. Salminen and F. Tompa, *Communicating with XML*, New York: Springer-Verlag, ch. 1, 2011.

Tracking Emotions of Bloggers – A Case Study for Bengali

Dipankar Das and Sivaji Bandyopadhyay

Abstract—**The present paper describes the identification and tracking of bloggers' emotions with respect to time from the structured Bengali blog documents. The assignment of Ekman's six basic emotions to the bloggers' comments is carried out at sentence and paragraph level granularities. The Referential Informative Chain (*RIC*) developed for each blogger consists of the nodes representing the emotional states of that blogger. Each node of a *RIC* contains the identification information of its associated blogger, timestamp, section and emotional sentences. The nodes are arranged in each *RIC* based on the ascending order of the associated timestamps. An affect scoring technique has been employed to capture the emotions from each of the nodes of a blogger's *RIC*. The incorporation of self emotions and influential emotions as extracted from other bloggers plays a significant role in detecting the emotions of a blogger's present state. The *Extrinsic* evaluation produces *precision* (P), *recall* (R) and F-Measure of 61.05%, 69.81% and 65.13% respectively for evaluating the total of 193 emotional states of 20 bloggers. The *Intrinsic* evaluation has been conducted using a manual rater with the help of a statistical agreement coefficient, Krippendorff's alpha α. Two types of alpha, namely nominal alpha and interval alpha produce the average scores of 0.67 and 0.72, respectively.**

Index Terms—**Tracking, emotions, bloggers, affect score, agreement.**

I. Introduction

SENTIMENT Analysis and Opinion Mining have been attempted with more focused perspectives rather than fine-grained emotions [1]. In psychology and common use, emotion is an aspect of a person's mental state of being, normally based in or tied to the person's internal (physical) and external (social) sensory feeling [2]. The determination of emotions expressed in the text with respect to reader or writer is itself a challenging issue [3] as emotion is not open to any objective observation or verification [4]. Not only the classification of reviews [5], newspaper articles [6] or blogs [7], a wide range of other Natural Language Processing (NLP) tasks such as tracking users' emotion about products or events or about politics as expressed in online forums or news, to customer relationship management are also using emotional information.

Researches on emotion show that blogs play the role of a substrate to analyze the reactions of different emotional enzymes. Many blogs act as online diaries of the bloggers reporting their daily activities and surroundings. Sometimes, the blog posts are annotated by other bloggers. If we consider such bloggers as the emotion holders of different stance on diverse social and political issues, we can have a better understanding of the relationships among countries or among organizations [8].

Among all concerns, sentiments or emotions of people are important because people's sentiment has great influence on our society. Recently, the identification of the temporal trends of sentiments and topics has drawn attention of NLP communities [9]. The perspectives of sociology, psychology and commerce along with the close association among people, topic and sentiment motivate us to investigate the insides of emotional changes of people over topic and time.

The present task involves the identification and tracking of bloggers' emotions with respect to time from Bengali web blog archive[1]. The sections of the bloggers' comments for a given topic contain nested tree like structures along with distinguishable information regarding individual blogger and his/her associated blog posting timestamps. Presently, the comment sentences are already annotated during the development of the Bengali emotional blog corpus [10]. Hence, by default, each of the comment sections is assigned with the types of emotions that are present in the annotated sentences of that section. A Referential Informative Chain (*RIC*) for each blogger is developed by using the default annotated timestamp, unique identifier and emotional comments. These information are acquired from the nested tree like comment sections of each blog document. Each node of a *RIC* denotes the emotional state of its corresponding blogger at a particular time instance and all nodes of a *RIC* are arranged based on the ascending order of the associated timestamps.

The identification of Ekman's [23] six basic emotions from the bloggers' comments is carried out at sentence and paragraph level granularities using the information of word level constituents along with sense based scoring mechanism [11]. An affect scoring technique has been employed to identify the emotions of a state or node in each of the Referential Informative Chains (*RICs*) of the bloggers. The *Self Affect Score* (*SAS*) is added with the *Influential Affect Score* (*IAS*) to produce the *Emotional Score* (*ES*) of a blogger at each node with a particular timestamp. We have considered the *Self Affect Score* (*SAS*) of a blogger at a particular

Manuscript received on November 12, 2010, accepted on December 2, 2010.

D. Das is with the Department of Computer Science and Engineering, Jadavpur University, Kolkata, India, 700032 (phone: +91-9432226464; e-mail: dipankar.dipnil2005@gmail.com).

S. Bandyopadhyay is with Department of Computer Science and Engineering, Jadavpur University, Kolkata, India, 700032 (phone: +91-9433579595; e-mail: sivaji_cse_ju@yahoo.com).

[1] www.amarblog.com

timestamp as the *Emotion Score* (*ES*) of that blogger at its immediately previous timestamp. The *Influential Affect Score* (*IAS*) is the cumulative summation of the *Emotion Scores* (*ES*) of all other participated bloggers who commented between the blogger's immediate previous timestamp to its current timestamp. In case of calculating the *Influential Affect Score* (*IAS*) of a blogger, we have hypothesized that the *Emotion Scores* (*ES*) of other participated bloggers are to be calculated independently without considering the inclusion of the self and influential affects of those bloggers.

The best three Ekman's emotions that are acquired from the ordered *Emotion Scores* (*ESs*) are assigned to the associated node of the blogger's corresponding *RIC* as the identified emotions. The change of a blogger's emotions has been tracked based on the emotions that are assigned to the nodes of the blogger's *RIC.*

The importance of self and influential affects in tracking results has been evaluated in two stages based on the identified emotions at each node. One is the *Extrinsic* evaluation that is carried out through the standard metrics such as *precision* (P), *recall* (R) and F-Measure and another one is the *Intrinsic* evaluation that considers the involvement of a manual rater and measures the performance using the statistical agreement coefficient, Krippendorff's alpha α [21] [22]. The system achieves *precision* (P), *recall* (R) and F-Measure of 61.05%, 69.81% and 65.13% respectively in case of *Extrinsic* evaluation. On the other hand, two types of alpha, nominal alpha (N**α**) and interval alpha (I**α**) produces the satisfactory average scores of 0.67 and 0.71 respectively for evaluating a total number of 193 emotional states of 20 bloggers from a collection of 34 blog documents.

The rest of the paper is organized as follows. Section II describes the related work. The preparation of Referential Informative Chains (*RICs*) from the annotated and nested comment sections of the structured blog documents is discussed in Section III. Automatic emotion tagging of the bloggers' comments is described in Section IV. The affect scoring technique for identifying the emotion states and their tracking are discussed in Section V. The evaluation schemes along with the performance measures of the emotional states are discussed in Section VI. Finally, Section VII concludes the paper.

II. Related Work

The earlier researches on emotion analysis reveal that a large number of contributory attempts were conducted for identification, generation or classification of emotions rather than tracking of emotions over event, topic or time. Mishne and de Rijke [7] proposed a system, *MoodViews*[2] to analyze the temporal change of sentiment. MoodViews analyzes multiple sentiments by using 132 sentiments used in LiveJournal[3]. Although our concept for the emotion tracking is similar to MoodViews, instead of using only the temporal relations, we focus on identifying the changes of emotions of the blog users over time. Another system called *ThemeRiver* [15] visualizes thematic flows along with timeline. The present approach is different from ThemeRiver as we focus on tracking the emotion flows of the bloggers considering their self as well as influential emotions. The temporal sentiment identification from social events has been carried out in [9]. In their task, the authors have analyzed the temporal trends of sentiments and topics from a text archive that has timestamps in weblog and news articles and produces two kinds of graphs, *topic graph* that shows temporal change of the topics associated with a sentiment, and *sentiment graph* that shows temporal change of sentiments associated with a topic. In contrast, the present task considers the involvement of self and influential emotions in determining the emotions of the bloggers at a particular timestamp or at each emotional state. We have constructed Referential Informative Chain (*RIC*) for individual blogger to track emotions associated with each node of its *RIC.*

III. Referential Informative Chain (RIC) Preparation

The mode of language technology has changed dramatically since the last few years with the web being used as a data source in a wide range of research activities. There is a long history of creating a standard for western language resources. The Human Language Technology (HLT) society in Europe has been particularly zealous for the standardization of European languages [16]. The authors also mentioned that, in spite of having great linguistic and cultural diversities, Asian language resources have received much less attention than their western counterparts. In Asia, India is a multilingual country with a diverse cultural heritage. Indian languages are resource constrained. Bengali is the fifth popular language in the World, second in India and the national language in Bangladesh but it is less privileged and less computerized compared to English. Following observations have motivated us to develop the Bengali emotion corpus from the web.

Recent study shows that non-native English speakers support the growing use of the Internet[4]. The focus is to improve the multilingual search engines on the basis of sentiment or emotion. This raises the demand of linguistic resources for the languages other than English.

Majority of the existing works in this field have been conducted for English [17]. To the best of our knowledge, at present, there is no such available corpus that is annotated with detailed linguistic expressions for emotion in Bengali or even for other Indian languages. Hence, the work is a foray into emotion analysis for an Indian language such as Bengali.

The blog documents are stored in the format as shown in Figure 1 after retrieval from the Bengali web blog archive (www.amarblog.com). Each of the blog documents is assigned with a unique identifier (*docid#*) followed by a section devoted for topic and several sections devoted for different users' comments. Each comment section consists of several nested

[2] http://moodviews.com/
[3] http://www.livejournal.com/

and overlapped sub sections that also contain the bloggers' comments. Each of the comment sections of an individual blogger is uniquely identified by the notion of section identification number (*secid*#). Each section contains the information regarding identification number of the blog user (*uid*#) and associated timestamp (*tid*#).

We have considered the individual comment section as separate paragraph that contains several emotional sentences. The sentences that are present in the bloggers' comment sections are already annotated during the development of Bengali emotional blog corpus [10]. Hence, by default, each of the comment sections is assigned with all types of emotions that are present at the annotated sentences of that section. The emotions present in such individual comment section represent the emotional state of the blogger at that timestamp. The annotated emotional states are verified by the authors. It is observed that the annotation task for sentential emotions compensate the manual effort of verifying the emotional states and the result produces satisfactory impression.

The Referential Informative Chain (*RIC*) for each of the bloggers is constructed by acquiring the default annotated information like timestamp (*tid*#), unique identifier (*uid*#) and emotional comments that are acquired from the nested tree like structure of the comment sections. Though all of the comment sections (as shown using the tag <*User Comments id=UC#*>) in the individual blog documents refer to a single topic, presently we have considered each comment section as a separate unit by assuming that no inter emotional impact exists among the various comment sections of a single blog document.

The individual *RIC* is developed for each single blogger with respect to each comment section. Each node of a *RIC* denotes the emotional state of the blogger at a particular time instance and the sequence of adding information into the nodes is based on the ascending order of associated timestamps. For example, in Figure 1, the two nodes namely *n1* and *n2* will be added into the front of the *RIC* developed for the blog user with *uid*=1. The associated timestamps (*t1*, *t4*) and emotions will also be added into the nodes accordingly. As *t4* > *t1*, the inclusion of node *n1* is considered before the inclusion of *n2* into the corresponding *RIC* of *uid* 1.

IV. Automatic Emotion Tagging

Each of the comment sections has been considered as a separate paragraph that contains the emotional sentences. The hypothesis of defining an emotional state with respect to each comment section or paragraph is to include all of the annotated sentential emotions that are contained in that paragraph. It is said that sentiment is typically a localized phenomenon that is more appropriately computed at the paragraph, sentence or entity level [24]. Hence, our primary investigation mainly aims to automatically identify the emotions at sentence and paragraph level from the bloggers' comment sections. Ekman's (1993) six basic emotion types, such as *happiness, sadness, anger, fear, surprise,* and *disgust* are considered to accomplish the emotion tagging task as these emotions are termed as universal emotions [23]. The non emotional sentences are considered as neutral.

```
-<DOC docid = xyz>
      +<Topic>…. </Topic>
      -<User Comments id=UC1>
            -<U uid=1, tid=t1, secid=UC1>….
                  -<U uid=2, tid=t2, secid=UC1.1>…</U>
                  -<U uid=3, tid=t3, secid=UC1.2>…</U>
                        -<U uid=1, tid=t4, secid=UC1.2.1>…</U>
                  ….
                </U>
       </User Comments>
      +<User Comments id=UC2>
      +<User Comments id=UC3>
      …
</DOC>
```

Fig. 1. General Structure of a blog document

We have employed a sentential emotion tagging system [11] that consists of two prong approach, word level followed by sentence level. The Conditional Random Field (*CRF*) [19] based machine learning approach that incorporates several singleton features (e.g. *Part of Speech (POS) of the words, Question words, Reduplication, Colloquial / Foreign words, Special Punctuation Symbols, Negative words, Words of Quoted sentence, Emoticons*), context features (*unigram, bigram*) at word level as well as POS tag level along with different combinations of singleton and context features has been used for word level emotion tagging. The only difference that has been considered in our present attempt is the use of the Bengali *WordNet Affect Lists* [13] instead of using the Bengali *SentiWordNet*. The incorporation of error analysis and equal distribution of emotion tags with the non-emotion tag improves the word level emotion tagging system. The system demonstrates satisfactory performance with an average accuracy of 66.74% with respect to all emotion classes [18].

The sense based and corpus based scoring strategies are applied on the acquired word level emotion constituents to identify the sentence level emotion tags. The corpus based scores are calculated based on the frequency of occurrence of an emotion tag with respect to the total number of occurrences of all six types of emotion tags in an annotated corpus whereas sense based scores are calculated using SentiWordNet [20]. But, we have considered only the sense based scoring technique to accomplish our present research goal. The *Sense_Tag_Weights* (*STWs*) are the *tag weights* that are calculated using English SentiWordNet [20]. The basic six words "*happy*", "*sad*", "*anger*", "*disgust*", "*fear*" "*surprise*" are selected as the *seed words* corresponding to each emotion type. The *positive* and *negative* scores of each synset in which each of these *seed words* appear are retrieved from the English *SentiWordNet* [20] and the average of the scores is fixed as the *Sense_Tag_Weight* (*STW*) of that particular emotion tag (*happy:* 0.0125, *sad:* - 0.1022, *anger:* - 0.5, *disgust:* - 0.075, *fear:* 0.0131, *surprise:* 0.0625, and *neutral:* 0.0) [11].

Each sentence is assigned with a *Sense_Weight_Score* (*SWS*) for each emotion tag which is calculated by dividing the total *Sense_Tag_Weights* (*STWs*) of all occurrences of an

[4] http://www.internetworldstats.com/stats.htm

emotion tag in the sentence by the total *Sense_Tag_Weights* (*STW*s) of all types of emotion tags present in that sentence. The *Sense_Weight_Score* is calculated as $SWS_i = (STW_i * N_i) / (\sum_{j=1 \text{ to } 7} STW_j * N_j) \mid i \in j$ where SWS_i is the sentence level *Sense_Weight_Score* for the emotion tag ***i*** and ***Ni*** is the number of occurrences of that emotion tag in the sentence. STW_i and STW_j are the *Sense_Tag_Weights* for the emotion tags ***i*** and ***j*** respectively.

On the other hand, the polarity information (i.e. *positive* and *negative*) that is associated with the word level *Sense_Tag_Weights* (*STWs*) is carried forward to the sentence level *Sense_Weight_Scores* (*SWSs*) for each emotion types. Therefore, we preserve the information of *Sense_Weight_Scores* (*SWSs*) to utilize them in calculating the *Affect Scores* (*ASs*) of each comment section or paragraph. Though we have considered all the annotated sentential emotions as the default emotions of its corresponding paragraph, but the system produces the *Affect Scores* (*ASs*) for individual emotional state by summing up the sentential *Sense_Weight_Scores* (*SWSs*) with respect to each of the six emotion types.

V. Affect Scoring

The *Affect Scores* (*ASs*) for each of the six emotions are preserved to identify the emotional states of an individual blogger in its corresponding Referential Informative Chain (*RIC*). By assuming each of the comment sections as an interactive system with respect to bloggers' communication, in addition to *Affect Score* (*AS*), two other types of *Affect Scores* (*ASs*) are also used to determine the six *Emotion Score* (*ES*) of each blogger who participated in that interactive system. The *Self Affect Scores* (*SASs*) are used to measure the effect of the blogger's own previous emotions in identifying its present emotional state whereas the *Influential Affect Score* (*IASs*) are considered for measuring the previous impact of all other bloggers' emotions in identifying a blogger's present emotional state. The two *Affect Scores* (*ASs*), *Self Affect Score* (*SAS*) and *Influential Affect Score* (*IAS*) are calculated based on Ekman's six basic emotion classes. *Emotion Score* (*ES*) for each of the six emotion classes is the summation of the three *Affect Scores* (*ASs*) corresponding to that emotion class, $ES_i = AS_i + SAS_i + IAS_i \mid i \in$ Ekman's six emotion classes.

If the target blogger is a beginner or blog starter or who first starts commenting, the *Emotion Scores* (*ESs*) of the blogger are then the present *Affect Scores* (*ASs*), i.e., $ES_{t0}=AS_{t0}$, where $t0$ is the first timestamp associated with the first node of the target blogger's *RIC*. In other cases, *Self Affect Scores* (*SASs*) of a target blogger is the *Emotion Scores* (*ESs*) at its immediate previous timestamp and the *Influential Affect Score* (*IAS*) is the cumulative summation of the *Emotion Scores* (*ESs*) of all other participated bloggers who commented between the target blogger's immediate previous timestamp and present timestamp. The *Self Affect Scores* (*SASs*) as well as the *Influential Affect Scores* (*IASs*) for an initiator target blogger at the start timestamp are considered as zero. For a non initiator target blogger, only the *Self Affect Scores* (*SASs*) are considered as zero at the start timestamp as it first starts commenting from that timestamp. The emotions of the target blogger are calculated at each timestamp by assuming that the emotions of other participating bloggers are not biased by other bloggers' emotions including the target till that timestamp. Hence, in case of calculating the *Influential Affect Score* (*IAS*) of a target blogger, we have hypothesized that the *Emotion Scores* (*ES*) of other participating bloggers are to be calculated independently without considering the inclusion of the self and influential affects for these participating bloggers.

For example, if we consider the following snapshot from Figure 1 to calculate the *Emotion Scores* (*ESs*) of the blogger with *uid=1*at different timestamps or assign the *Emotion Scores* (*ESs*) to the emotional states or nodes of its *RIC*, we need to consider both the *Self Affect Score* (*SAS*) as well as the *Influential Affect Score* (*IAS*). As this target blogger is an initiator of its corresponding comment section, at timestamp *t1*, the six *Emotion Scores* (*ESs*) of the target blogger are equal to the six *Affect Scores* (*AS*) that are calculated from the sentences of the corresponding comment section with *secid=UC1*. The first node of the target blogger's *RIC* is also assigned with the six *Emotion Scores* (*ES*). At timestamp *t4*, the second node of the blogger's *RIC* is assigned with the six *Emotion Scores* (*ESs*) that are calculated based on the *Self Affect Scores* (*SASs*) as well as the *Influential Affect Scores* (*IASs*). Not only the six *Affect Scores* (*ASs*) that are calculated from the sentences of the comment section (*secid=UC1.2.1*) but the immediate previous *Emotion Scores* (*ESs*) (in this case the *Affect Scores* (*ASs*) of the target blogger's previous comment section *secid=UC1*) are also added as the *Self Affect Scores* (*SASs*) to calculate the six *Emotion Scores* (*ESs*) at *t4*. The *Self Affect Scores* (*SASs*) are included in the present task to consider the emotional impact of a blogger's previous emotions into its present emotional state.

In addition to *Affect Scores* (*ASs*) and *Self Affect Scores* (*SASs*), the six *Influential Affect Scores* (*IASs*) are also added. The *Influential Affect Scores* (*IASs*) are included to consider the emotional impact of other previous participating bloggers into the present emotional state. For each of the six emotion classes, the *Influential Affect Score* (*IAS*) is the cumulative summation of the *Emotion Scores* (*ESs*) of all other participating bloggers between the timestamps *t1* and *t4*. In this case, the *Influential Affect Score* (*IAS*) for each of the six emotion types is calculated based on the *Emotion Scores* (*ESs*) of other participating bloggers (with *uid=2* and *uid=3*) at timestamp *t2* and *t3* (as *t1<t2*, *t3<t4*). The *Emotion Score* (*ES*) of the blogger (*uid=1*) at timestamp *t4* with respect to a particular emotion class is defined as follows:

$ES_{t4} = AS_{t4} + SAS_{t4} + IAS_{(t1, t4)}$ where $SAS_{t4} = ES_{t1}$ and $IAS_{(t1, t4)} = ES_{t2}$ (*uid=2*) + ES_{t3} (*uid=3*)

-<U *uid*=1, tid=t1, secid=UC1>….
-<U *uid*=2, tid=t2, secid=UC1.1>…</U>
-<U *uid*3 tid=t3, secid=UC1.2>…</U>
-<U *uid*=1 tid=t4, secid=UC1.2.1>…</U>…

It has to be mentioned that, as *Influential Affect Score* (*IAS*) should have some emotional impact; therefore influence of emotions on other bloggers is being taken into consideration. Each of the nodes in the blogger's RIC is assigned with the six *Emotion Scores* (*ESs*). The best three emotions based on the ordered *Emotion Scores* (*ESs*) of a node are tagged as the emotional state of the blogger at that timestamp. The tracking of a blogger's emotional states via the nodes of its corresponding *RIC* is carried out based on the associated the best three emotions. The change of a blogger's emotions has been tracked based on the emotions that are assigned to the nodes of the blogger's *RIC.*

Tracking along the time dimension is trivial because most text documents for analysis come with time when they were written, which is especially true for online documents such as product reviews, forum postings, and blogs. The challenge is still the accuracy of sentiment or emotion prediction and solving the associated problems [24]. Hence, the importance of self and influential affects in tracking results has been evaluated in two stages based on the identified emotions at each node.

VI. Evaluation

The evaluation of the emotion tracking system has been carried out at two levels to measure the performance of the system. One is the *Extrinsic* evaluation that is carried out through the standard metrics such as *precision* (P), *recall* (R) and F-Measure and the other is the *Intrinsic* evaluation that deals with respect to the agreement between manual annotation and system generated output. We have measured the Krippendorff's (2004) [21] [22] alpha coefficient, a standard metric used for inter-annotator reliability studies to consider the evaluator disagreements. The metric is also used in counseling and survey research, psychological testing, observational studies etc. The metrics for *Extrinsic* evaluation gives the coarse-grained statistics whereas the motivation of selecting the metric for *Intrinsic* evaluation is to identify the fine grained clarity that exists among the emotions present in the emotional states or nodes of an individual blogger's Referential Informative Chain (*RIC*).

The evaluation has been carried out with respect to a total of 193 emotional states of 20 bloggers from a collection of 34 blog documents. In case of *Extrinsic* evaluation, the system generated emotions are evaluated against the annotated emotions. We have not considered the emotions of individual states or nodes of the *RICs*. For the very reason, the *precision* (P), *recall* (R) and F-Measure have been calculated with respect to all emotion classes. It is found that the incorporation of *Self Affect Score* (*SAS*) and *Influential Affect Score* (*IAS*) improve both *precision* (P), *recall* (R) metrics. The results are shown in Table I.

Though the coarse grained evaluation shows satisfactory performance of the system, the fine grained clarity of assigned emotions in each of the states or nodes of the bloggers' *RICs* is evaluated by measuring the inter-rater agreement using

TABLE I
Extrinsic evaluation

Techniques	Precision (P)	Recall (R)	F-Measure
Affect Scoring (AS)	45.87	56.05	51.34
Self Affect Scoring (SAS)	16.22	28.12	22.10
Influential Affect Scoring (IAS)	20.63	32.98	26.72
AS+SAS	55.44	63.04	59.25
AS+IAS	59.66	65.35	62.82
AS+SAS+IAS	61.05	69.81	65.13

Krippendorff's alpha [21]. This evaluation technique is also termed as *Intrinsic* evaluation. It is a statistical measure of the agreement achieved when coding a set of units of analysis in terms of the values of a variable. The equation of alpha is as follows:

$$_{metric}\alpha = 1 - \frac{D_o}{D_e} = 1 - \frac{\sum_{c=1,k=1}^{v} o_{ck\,metric}\delta_{ck}^2}{\frac{1}{n-1}\sum_{c=1,k=1}^{v} n_c n_k \;_{metric}\delta_{ck}^2}$$

where D_o and D_e are the observed and expected disagreements, $_{metric}\delta_{ck}^2$, a difference function between values c and k reflect the metric properties (Levels of Measurement) of the variable and O_{ck}, a coincidence matrix that cross tabulates the n pair-able values from the canonical form of the reliability data into a v-by-v square matrix, where v is the number of values available in a variable. The definition of n_c, n_k and n are as follows:

$$n_c = \sum_{k=1}^{v} o_{ck}, \quad n_k = \sum_{c=1}^{v} o_{ck}, \text{ and } \quad n = \sum_{c=1,k=1}^{v} o_{ck}$$

We have considered two metrics, *Nominal Alpha* (N**α**) and *Interval Alpha* (I**α**) for measuring the agreement. Hence, the following equations corresponding to their difference function has been considered to accomplish the goal:

$$_{nominal}\delta_{ck}^2 = \begin{cases} 0 & \text{iff } c = k \\ 1 & \text{iff } c \neq k \end{cases} \quad \text{and} \quad _{interval}\delta_{ck}^2 = (c-k)^2$$

On the other hand, the Krippendorff's alpha is applicable to any number of coders, each assigning one value to one unit of analysis, to incomplete (missing) data, to any number of values available for coding a variable, to binary, nominal, ordinal, interval, ratio, polar, and circular metrics (Levels of Measurement), and it adjusts itself to small sample sizes of the reliability data.

Thus, we have considered the emotional states of any blogger's *RIC* as the set of units to be analyzed. The four values out of which three correspond to the number of emotions assigned to the state and one value corresponds to the undetermined emotion. It has been observed that the system assigns the six *Emotion Scores* (*ESs*) of value zero to some nodes in the *RICs*. These states are termed as *Undetermined Emotional States* (*UESs*). The evaluation technique considers the *Undetermined Emotional States* (*UESs*) as the incomplete or missing data. Otherwise, the best one or two or three emotions are assigned to the nodes based

on the availability of ordered *Emotion Scores* (*ESs*) with values greater than zero as obtained by the affect scoring technique.

The *Self Affect Scores* (*SASs*) and *Influential Affect Scores* (*IASs*) have been introduced to minimize the production of *Undetermined Emotional States* (*UESs*) assuming each current state of the blogger contains more or less affections from its previous emotional states. The requirement of normalizing the produced *Emotion Scores* (*ESs*) for each blogger was not faced as the average length of the *RICs* is not very large and the *Emotion Scores* (*ESs*) of other bloggers are independently considered during the calculation of *Influential Affect Scores* (*IASs*). On the other hand, the identified emotions for each node of a *RIC* are stored in a vector. One manual rater has been appointed to carry out the assignment of the values to each unit based on the identified emotions and their correctness. The manual assignment has been performed against the annotated assignment. The annotated states also contain one or multiple or no emotions and similarly these gold standard emotions are represented using a vector. Each of the vectors for each node of a blogger's *RIC* is considered as a unit and any of the four values is assigned to each unit based on the number of emotions present in annotated vector and system generated vector.

Each of the nodes or emotional state of the bloggers' *RICs* is assigned the numeric values of the two raters. The *Nominal Alpha* (N**α**) and *Interval Alpha* (I**α**) are calculated for each of the bloggers' RICs. The number of emotional states or nodes of a blogger's *RIC* represents the number of units for analysis. We have shown the average scores in Table II with respect to a total of 193 emotional states of 20 bloggers. It is observed that the agreements have produced the alpha scores of 0.67 and 0.71 for *Nominal Alpha* (N**α**) and *Interval Alpha* (I**α**) respectively. **Iα > Nα** signifies that the disagreements happen not largely among the neighboring values but the primary observation suggests that the incorporation of the two affect scores, *Self Affect Scores* (*SASs*) and *Influential Affect Scores* (*IASs*) substantially reduces the disagreement among the raters.

VII. Conclusion

In this paper, we have reported our work on identification and tracking of bloggers' emotions from structured Bengali blog documents. The present system shows the effectiveness of utilizing the previous information regarding self emotions and other bloggers' influential emotions in case of identifying present emotion of any blogger. Both of the *Extrinsic* and *Intrinsic* evaluations show significant improvement in the performance of the system. Our future plan is to employ the system in identifying emotional changes from lengthy referential chains to analyze the potential reasoning behind the change. The dependency among the bloggers' emotions is to be analyzed for capturing the reasons of emotional changes too. The hypothesis of the present model will be used in future for developing a topic driven emotion tracking model.

TABLE II
Intrinsic evaluation

Techniques	Nominal Alpha (Nα)	Interval Alpha (Iα)
Affect Scoring (AS)	0.35	0.43
Self Affect Scoring (SAS)	0.15	0.24
Influential Affect Scoring (IAS)	0.23	0.31
AS+SAS	0.56	0.58
AS+IAS	0.59	0.62
AS+SAS+IAS	0.67	0.72

Acknowledgements

The work reported in this paper was supported by a grant from the India-Japan Cooperative Programme (DSTJST) 2009 Research project entitled "Sentiment Analysis where AI meets Psychology" funded by Department of Science and Technology (DST), Government of India.

References

[1] C. Quan and F. Ren, "Construction of a Blog Emotion Corpus for Chinese Emotional Expression Analysis," in *Empirical Method in Natural Language Processing – Association for Computational Linguistics*, pp. 1446-1454, 2009.

[2] Y. Zhang, Z. Li, F. Ren and S. Kuroiwa, "A Preliminary Research of Chinese Emotion Classification Model," in *IJCSNS International Journal of Computer Science and Network Security*, vol. 8(11), pp. 127-132, 2008.

[3] C. Yang, K. H. Y. Lin and H.H Chen, "Writer Meets Reader: Emotion Analysis of Social Media from both the Writer's and Reader's Perspectives," in *009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, pp. 287-290, 2009.

[4] R. Quirk, S. Greenbaum, G. Leech and J. Svartvik, *A comprehensive Grammar of the English Language*, Longman, New York, 1985.

[5] P.D. Turney, "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews," in *Annual Meeting of the Association for Computational Linguistics*, pp. 417- 424, 2002.

[6] K. H.-Y. Lin, C. Yang, and H.-H. Chen, "What Emotions News Articles Trigger in Their Readers?" in *SIGIR*, pp. 733-734, 2007.

[7] G. Mishne and M. de Rijke, "MoodViews: Tools for Blog Mood Analysis," in *AAAI 2006 Spring Symposium on Computational Approaches to analyzing* Weblogs, 2006.

[8] Soo-Min Kim and E. Hovy, "Extracting Opinions, Opinion Holders, and Topics Expressed in Online News Media Text," in *ACL*, 2006.

[9] T. Fukuhara, H. Nakagawa and T. Nishida, "Understanding Sentiment of People from News Articles: Temporal Sentiment Analysis of Social Events," in *ICWSM'2007*, Boulder, Colorado, USA, 2007.

[10] D. Das and S. Bandyopadhyay, "Labeling Emotion in Bengali Blog Corpus – A Fine Grained Tagging at Sentence Level," in *8th Worshop on Asian Language Resources (ALR8), COLING*, pp. 47-55, 2010.

[11] D. Das and S. Bandyopadhyay, "Word to Sentence Level Emotion Tagging for Bengali Blogs" in *ACL-IJCNLP*, pp. 149-152, 2009.

[12] D. Das and S. Bandyopadhyay, "Sentence Level Emotion Tagging on Blog and News Corpora," *Journal of Intelligent System (JIS)*, vol. 19(2), pp. 125-134, 2010.

[13] D. Das and S. Bandyopadhyay, "Developing Bengali WordNet Affect for Analyzing Emotion," in *International Conference on the Computer Processing of Oriental Languages*, pp. 35-40, 2010.

[14] S. Sood and L. Vasserman, "ESSE: Exploring Mood on the Web," in *AAAI Conference on Weblogs and Social Media (ICWSM) Data Challenge Workshop*, 2009.

[15] S. Havre, E. Hetzler, P. Whitney, and L. Nowell, "ThemeRiver: Visualizing Thematic Changes in Large Document Collections," in *IEEE Transactions on Visualization and Computer Graphics*, vol. 8(1), pp. 9-20, 2002.

[16] A. Ekbal and S. Bandyopadhyay, "A Web-based Bengali News Corpus for Named Entity Recognition," in *Language Resources and Evaluation* vol. 42(2), pp. 173-182, 2008.
[17] B. Pang and L. Lee, "Opinion Mining and Sentiment analysis," *Foundations and Trends in Information Retrieval*, vol. 2(1-2), pp. 1-135, 2008.
[18] D. Das and S. Bandyopadhyay, "Emotion Tagging – A Comparative Study on Bengali and English Blogs," in *International Conference on Natural Language Processing*, pp. 177-184, 2009.
[19] J. Lafferty, A.K. McCallum and F. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," in *International Conference on Machine Learning*, 2001.
[20] A. Esuli and F. Sebastiani, "SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining," in *LREC,* 2006.
[21] K. Krippendorff, "Estimating the reliability, systematic error, and random error of interval data," *Educational and Psychological Measurement*, vol. 30 (1), pp. 61-70, 1970.
[22] K. Krippendorff, *Content analysis: An introduction to its methodology*, Thousand Oaks, CA: Sage, 2004.
[23] P. Ekman, "Facial expression and emotion," *American Psychologist*, vol. 48(4), pp. 384-392, 1993.
[24] B. Liu, "The challenge is still the accuracy of sentiment prediction and solving the associated problems," in *5th Annual Text Analytics Summit*, 2009.

Knowledge Vertices in XUNL

Ronaldo Martins

***Abstract*—This paper addresses some lexical issues in the development of XUNL – a knowledge representation language descendent from and alternative to the Universal Networking Language (UNL). We present the current structure and the role of Universal Words (UW) in UNL and claim that the syntax and the semantics of UWs demand a thorough revision in order to accomplish the requirements of language, culture and human independency. We draw some guidelines for XUNL and argue that its vertices should be represented by Arabic numerals; should be equivalent to sets of synonyms; should consist of generative lexical roots; should correspond to the elementary particles of meaning; and should not bear any non-relational meaning.**

***Index Terms*—Knowledge representation, UNL, lexical resources, semantic networks.**

I. Introduction

THE XUNL (eXtended UNL) is a knowledge representation language descendent from, and alternative to, the Universal Networking Language (UNL). It was inspired by the UNLX Project [1] – an intensive experience of using UNL for natural language analysis and generation – and by the resulting conviction that some principles of UNL should be thoroughly reformulated in order to cope with multilingual knowledge representation.

In short, XUNL departs from four axioms:

I. Knowledge can be represented as the directed hyper-graph K = (V,H), where V is the set of knowledge vertices, and H – which stands for knowledge hyper-edges – is a set of non-empty subsets of V.

II. A hyper-graph K = (V,H) is isomorphic (≃) to the hyper-graph K' = (V',H') if there exists a bijection $\phi: V \rightarrow V'$ and a permutation π $\varphi(hi) = f\pi(i)$, where h_i⊆H.

III. A natural language utterance describes a sub-hyper-graph of K induced by a subset A of V, such as KA ⊆K and KA = (A, (H ∩ A)*).

IV. Two natural language utterances KA and KB are interchangeable if and only if they are isomorphic (K_A≃K_B).

In this paper, we address the first premise, which states that human knowledge is discrete and relational: it would consist of "knowledge vertices" linked by "knowledge hyper-edges". Knowledge vertices (KV) would stand for atomic facts of human thought; and knowledge hyper-edges (KH) would represent directed relations between knowledge vertices or, recursively, between other knowledge edges.

In order to define the nature and the role of KVs, which roughly correspond, in the UNL approach, to the Universal Words (UWs), we revise the structure and the function of UWs in UNL, and demonstrate that the syntax and the semantics of the KVs should be rather different, in order to accomplish the requirements of language, culture and human independency. We draw some lexical guidelines for XUNL and prescribe the structure of KVs.

The paper is organized as follows: Section 2 presents briefly the Universal Networking Language; Section 3 brings the current syntax and semantics of UWs; Section 4 analyzes the concept of Master Definition (MD) and the structure of the UNL Knowledge Base (the UNLKB); Section 5 explores some problems and shortcomings of the current version of the UNL KB; and finally Section 6 introduces the guidelines for KVs in XNUL out of our experience with UNL.

II. UNL

The Universal Networking Language (UNL) is an "electronic language for computers to express and exchange every kind of information" [2]. It can be defined as a knowledge representation technique expected to figure either as a pivot language in multilingual machine translation systems or as a representation scheme in information retrieval applications. It has been developed since 1996, first by the Institute of Advanced Studies at the United Nations University, in Tokyo, Japan, and more recently by the UNDL Foundation, in Geneva, Switzerland.

Formally, UNL is a semantic network believed to be logically precise, humanly readable and computationally tractable. In the UNL approach, information conveyed by natural language is represented, sentence by sentence, as a hyper-graph composed of a set of directed binary labeled links (referred to as "relations") between nodes or hyper-nodes (the "Universal Words", or simply "UW"), which stand for concepts. UWs can also be annotated with "attributes" representing mainly modality.

As a matter of example, the English sentence 'Peter kissed Mary?!' can be represented in UNL as follows:

```
[S]
{unl}
agt(kiss(agt>person,obj>person).@entry.@past.
      @interrogative.@exclamative, Peter(iof>person))
obj(kiss(agt>person,obj>person).@entry.@past.
      @interrogative.@exclamative, Mary(iof>person))
{/unl}
[/S]
```

Manuscript received on October 31, 2011, manuscript accepted on December 9, 2011.

Ronaldo Martins is with UNDL Foundation, 48 Route de Chancy, Petit-Lancy, CH-1312 Geneva, Switzerland (e-mail:r.martins@undlfoundation.org).

Differently from other semantic networks (such as conceptual graphs [3, 4] and the RDF [5]), UNL is not only a formalism; it is an entire language, enclosing a lexicon (the set of UWs) and a grammar (the set of relations and attributes). As of the version 2005 UNL Specifications [6], the set of relations, which is supposed to be closed and permanent, consists of 44 binary relations (such as agent, object, instrument, and so on); the set of attributes consists of 72 elements (interrogative, imperative, polite, etc.); and the set of UWs, which is open and subject to increase, consists of more than 63,000 entries.

Under the UNL Program, natural language analysis and understanding is referred to as a process of "enconverting" from natural language (NL) into UNL. This enconverting process, which has been carried out in a computer-aided human basis, is said to be not only a mere encoding, but truly a translation from the source sentence into a new target language – UNL – which is claimed to be as comprehensive as any NL. As a matter of fact, and at least for the time being, UNL has been mainly used for multilingual document generation, through a process referred to as "deconverting", which consists in automatically providing NL outputs to hand-coded UNL graphs.

III. UW

Universal Words, the words of UNL, are composed of a root (usually referred to, in UNL Specifications, as "headword") and a suffix ("the constraint list"). The latter comes between parentheses and is used mainly to disambiguate the former. Examples of UWs are presented below:

(1a) 'Universal Word'
(1b) 'UW(equ>Universal Word)'
(1c) 'Peter(iof>person)'
(1d) 'apple(icl>fruit)'
(1e) 'kiss(agt>person,obj>person)'
(1f) 'explain(icl>express(agt>thing,gol>person,obj>thing))'
(1g) 'Manyoshu(icl>Japanese poem)'

In order to be mnemonic and humanly readable, roots and suffixes are labeled out of English words, except for culture-dependent concepts (1g above). The UNL Center does not take that to be a language bias and claims that UWs are only arbitrary labels: they are but unique strings of characters used to refer to concepts. The meaning of a UW would be entirely derived from the so-called UNL Knowledge Base (or simply UNLKB), a huge network where nodes are interconnected as to emulate the structure of human cognition.

As a matter of example, the meaning of "apple(icl>fruit)"should be defined by a set of binary relations such as those indicated by (2) below :

(2a) icl(apple(icl>fruit),fruit(pof>plant))=1;
(2b) obj(eat(agt>thing,obj>thing), apple(icl>fruit))=1;
(2c) aoj(round(aoj>thing), apple(icl>fruit))=1;
(2d) pof(apple(icl>fruit), apple tree(icl>tree))=1;

Actually, in the UNL Program, there seems to be at least two different representational levels for defining UWs. The first is related to the UNLKB itself and targets the (alleged) systematic part of the meaning, in a sense very close to the one intended by the concept of "semantic markers" [7]. On the other hand, the unsystematic part of meaning (the "distinguishers") is treated in the UNL Encyclopedia, which is a huge UNL document base, also organized as a network, where idiosyncrasies and additional information on UWs are expected to be stored. Here we will focus only on the UNLKB structure**.**

IV. UNL KNOWLEDGE BASE

The UNLKB is a semantic network in which entries have the structure exemplified in (2) above. They comprise a binary directed relation (extracted from the UNL relation set) between two UWs, along with a degree of certainty, which can range from 0 (completely false) to 255 (completely true). Any UNL relation can hold between UWs in the UNLKB, and a single UW may receive and assign many different relations from and to other UWs. However, in order to guarantee inference and cross-reference inside the network, every UW should be linked to another one by at least one of three ontological relations: "icl" (a-kind-of), "iof" (an-instance-of) or "equ" (equal-to).

Linking one UW to another by means of "icl", "iof" or "equ" is to compose a sort of thesaurus, the UW Ontology, which is part of the UNLKB. Inside the UNL System, this subnetwork has been referred to as the "UW System", and constitutes a lattice structure, given that a single child-node may have many different parent-nodes. This hierarchical network also comprises an inheritance mechanism, so that all information assigned to a given parent-node can be directly inherited by its children-nodes. In this sense, if (3) below had been stated in the UNLKB, there would be no need for (2b), provided that it could be easily inferred from (2a):

(3) obj(eat(agt>thing,obj>thing), fruit(pof>plant))=1;
(2a) icl(apple(icl>fruit),fruit(pof>plant))=1;
(2b) obj(eat(agt>thing,obj>thing), apple(icl>fruit))=1;

The need for the UNLKB has been subject to criticism inside the UNL Project, but it should be observed that knowledge-based MT systems have proved to provide better results than those that are only language-based [8]. Inside the UNL System, the UNLKB is intended to assure robustness and precision both to the NL-UNL enconverting and to the UNL-NL deconverting. In the former case, the UNLKB would be used as a sort of word sense disambiguation device; in the latter, the UNLKB, through replacement operations, would allow for the deconversion of UWs not enclosed in the target language dictionaries. Additionally, the power of the UNLKB for intelligent search and semantic reasoning should never be underestimated.

In order to discipline and organize the creation of UWs, the UNL Center has proposed a particular technique for both naming and defining a UW in a single movement: the Master Definition (MD), introduced in 2000. The MD for naming the UW "apple(icl>fruit)" and defining it in the UNLKB (through an "icl" relation to the UW "fruit(pof>plant)") is presented in (4) below:

(4) apple(icl>fruit{pof>plant})

The MD is said to facilitate (and regulate) the labeling of a UW, which would derive its suffix (the constraint list) from its definition in the UNLKB. The name of the UW would simply be the same as the MD without the strings inside the curly braces.

However, it should be noticed that the concept of MD brings itself at least two serious shortcomings: 1) due to the simplification of syntax, the MD is not capable of conveying any degree of certainty other than 1; and 2) MDs can only be used to define the UW by means of 'icl', 'equ' or 'iof'; any richer definition would require longer strings and more expensive strategies. Nevertheless, and at least for the time being, the UNLKB has been entirely defined as a hierarchy of MDs.

V. Problems

We shall here concentrate on three main problems concerning the set of UWs in the current status of the UNL Program. The first is language-dependency; the second is culture-dependency; the third is human-dependency. In all those cases, we will address the set of UWs available at www.undl.org as of July, 2011. Some changes have been provided since then, but the problems remain basically the same.

A. Language-dependency

As for language-dependency, we claim that lexicalization of UWs has been exaggeratedly based on lexical items of English. This can be attested by the extensive presence of English idiosyncrasies in the set of UWs.

For instance, one will find, in the UNLKB, both "behavior(icl>action)" and "behaviour(icl>action)". The difference between them is not semantic, but strictly orthographic, and there is no reason for cataloging such kind of spelling difference in a semantic database.

The same should apply for pairs of antonyms such as give/receive, borrow/lend, etc. These verbs are supposed to convey the same meaning in a reversed subcategorization frame: give(x,y) = receive(y,x). Once "give" and "borrow" are there, would there be any reason for including "receive" and "lend" as well?

(5a) give(agt>thing,gol>person,obj>thing)
(5b) receive(agt>thing,obj>thing,src>thing)

(6a) borrow(agt>thing,obj>thing)
(6b) lend(agt>thing,gol>person,obj>thing)

This sort of overlapping among UWs does not affect only antonyms and can be found all over the UNLKB. Let us consider two last examples: is there any real need for registering, in the same knowledge base, all the words appearing in (7) and (8) below? Are the semantic differences between them really relevant? Are they going to be preserved in languages other than English?

(7a) begin(agt>thing,obj>thing)
(7b) commence(icl>begin(agt>thing,obj>thing))
(7c) start(icl>begin(agt>thing,obj>thing))

(8a) nurse(icl>medical assistant)
(8b) nurse({icl>person>human,}icl>occupation{>work})

The examples referred to above prove that economy has not been an asset of the UNLKB. Obviously, one may claim that variation should be represented, because there is no perfect synonymy, and UNL is supposed to be as comprehensive and fine-grained as any natural language. However, in this case, we would have a problem even more severe: provided that there is no perfect lexical matching between languages, UNL would have to register every word from every language, what would not only degrade the performance and the maintenance of UNL resources, but lead UNLKB to entropy and solipsism.

B. Culture-dependency

Culture-dependency can be detected mainly in the UNLKB categorization procedures, which has involved many inconsistencies. Tigers and panthers, for instance, are normally defined as belonging to the species of felines, but, in the UNLKB, they have been categorized directly under "mammal(icl>animal)", differently from "cat(icl>feline)":

(9a) tiger(icl>mammal{>animal})
(9b) panther(icl>mammal{>animal})
(9c) cat(icl>feline{>mammal})

In the same way, specific languages and types of languages have been categorized at the same level, as indicated in (10) below:

(10a) spoken language{(icl>language>system)}
(10b) Russian(icl>language{>system})
(10c) inflectional language{(icl>language>system)}

Circularity may also be found, as in (11) and (12):

(11) thing{(icl>nominal concept)}
abstract thing{(icl>thing)}
event(icl>abstract thing{>thing})
thing(icl>event{>abstract thing})

(12) figure(icl>figure{>attribute})

The main problem concerns the lack of criteria for categorization. In (13) below, for instance, the concept conveyed by the English words "film" and "movie" is said to be linked to the concept of "abstract thing". Why that? Why not "concrete thing"? Or why not "functional thing"? What about instances of films, such as "Gone with the wind"? Would they also be considered a kind of "abstract thing"?

(13) abstract thing{(icl>thing)}
art(icl>abstract thing)
cinema(icl>art{>abstract thing})
film(icl>cinema{>art})
movie(icl>cinema{>art})

Such categorization turns out to be even more astonishing if we consider the case for "book", which is also located under the "abstract thing" branch of the UNLKB, as indicated in (14):

(14) abstract thing{(icl>thing)}
information{(icl>abstract thing)}
document(icl>information)

TABLE I.
ENGLISH-TO-PORTUGUESE CORRESPONDENCE FOR THE NOUN "BOOK"

English	Definition	Portuguese
1. book	a written work or composition that has been published printed on pages bound together; "I am reading a good book on economics"	livro
2. book, volume	physical objects consisting of a number of pages bound together; "he used a large book as a doorstop"	brochura
3. ledger, leger, account book, book of account, book	a record in which commercial accounts are recorded; "they got a subpoena to examine our books"	registro
4. book	a number of sheets ticket or stamps etc. bound together on one edge; "he bought a book of stamps"	álbum
5. record, record book, book	a compilation of the known facts regarding something or someone; "Al Smith used to say, `Let's look at the record'"; "his name is in all the recordbooks"	registro
6. book	a major division of a long written composition; "the book of Isaiah"	livro
7. script, book, playscript	a written version of a play or other dramatic composition; used in preparing for a performance	livro
8. book, rule book	a collection of rules or prescribed standards on the basis of which decisions are made; "they run things by the book around here"	livro
9. Koran, Quran, al-Qur'an, Book	the sacred writings of Islam revealed by God to the prophet Muhammad during his life at Mecca and Medina	Livro
10. Bible, Christian Bible, Book, Good Book, Holy Scripture, Holy Writ, Scripture, Word of God, Word	the sacred writings of the Christian religions; "he went to carry the Word to the heathen"	Livro

book(icl>document{>information})
book of general works{(icl>book>document)}
manuscript{(icl>book of general works)}
rare book{(icl>book of general works)}
book of geography{(icl>book>document)}

On the other hand, both "landscape" and "scenery", and even "beauty spot", are categorized under "concrete thing", as seen in (15):

(15) *concrete thing{(icl>thing,icl>place>thing)}*
natural world{(icl>concrete thing,icl>place>thing)}
landscape(icl>natural world)
scenery(icl>landscape{>natural world})
beauty spot(icl>scenery{>landscape})
scene(icl>scenery{>landscape})

The absence of categorization guidelines causes the UNLKB to be excessively impressionist, in the sense it contains, to a considerable extent, subjective and personal ideas towards the world and the structure of events. Although some of those decisions may sound quite reasonable from a given perspective, it is clear that they cannot be taken for granted. They are rather culture- and even individual-dependent and will be subject to an everlasting dispute. In fact, this is said to be the main reason why knowledge-based approaches have been discarded as a feasible strategy for language processing and, inside the UNL Program, this is probably the reason why there is so much resistance on adopting a more fine-grained level of lexical description.

Actually, outside the UNL Center, it has been observed a relatively flat use of the suffixes of UWs, as if their only role was to assign some part-of-speech information to the roots. As a result of that, UWs such as "book(icl>thing)" have been more frequent than "book(icl>document)", for instance. These simple UWs, however, are not trouble-free either: they are not able to totally disambiguate English words and to assure precision and robustness to both enconverting and deconverting. In the Princeton WordNet [9], for instance, the noun 'book' (presented in Table I), may take 10 different senses, some of which may not be translated, in Portuguese or in any other language, by the same single word. In those circumstances, a low-level use of suffixes would not only be insufficient, but mostly misleading. To reduce all senses of "book" to "book(icl>thing)" would be no better than declaring that "book" is a sort of "abstract thing".

Consequently, the best solution for the limitations pointed out above is not to deprive the UNLKB, restricting its power and the granularity of its representation. Actually, the answer is to keep improving the UNLKB, but in a rather different perspective, as suggested in the next section.

C. *Culture-dependency*

The third limitation to be addressed here concerns the alleged "human-readability" of UNL graphs. As indicated above, UNL is an "electronic language for computers to express and exchange every kind of information". It is a not a language for humans. The argument that UNL graphs should use English words because they would be human-friendly is not only pointless but contradictory. Only very specialized people would be capable of reading UWs, and no one would be actually able to understand them, provided that UWs are only labels whose meanings should be extracted from the UNLKB. As we have already stressed, "apple(icl>fruit)" does not stand for "apple" in the human sense; it is the set of all relations

departing from and coming to "apple(icl>fruit)" in the UNLKB.

The option for English words, therefore, is not only useless but mainly misleading and deceiving. One should never forget that, from the computer point-of-view, "apple(icl>fruit)" is nothing but a memory address which will be meaningful if, and only if, is associated to other several memory addresses. If UNL is to be taken not as a mere notation, not as a mere natural language script, but as a real language, a truly different and autonomous one, self-coherent and natural-like, which could figure either as a source or a target in machine translation systems, we should strengthen that UNL cannot be about other languages (such as English), but that UNL must be directly about what other languages are about.

VI. Solutions

In order to circumvent the problems depicted in the last section, we propose three radical modifications in the UNL approach: UNL should not imitate English; UNL should not be an ontology-based language; and UNL should not be a knowledge-based language. As they strongly deviate from the main beliefs of the UNL approach, we acknowledge that such changes will lead us to something that is no longer UNL. In order to avoid confusion and dispute, we have been using the name "eXtended UNL" (or simply "XUNL"). XUNL keeps the core idea that knowledge conveyed by natural language sentences could be represented by hyper-graphs, in which nodes would stand for concepts, and edges would consist of directed binary semantic relations. However, nodes and edges in UNL and in XUNL are remarkably different, and should be therefore differently referred to. XUNL nodes have been addressed as "Knowledge Vertices" (KV) instead of UWs; and XUNL relations have been called "Knowledge Hyper-edges" (KH). In this section, we trace some general guidelines for Knowledge Vertices.

A. Language-dependency

The first commitment – not to imitate English – can be understood in two different senses. The most easily achievable is that XUNL should no longer use English words, or that KVs should be made out of language-independent symbols, such as Arabic numerals. In this case, KVs would not be as readily legible as UWs, but would be shorter, less deceptive and actually universal. Additionally, human readability could be easily provided by editing facilities as these existing in the very computer where this text is being typed, which automatically converts Roman characters into machine-tractable codes. Indeed, there is no actual need for middle-level representations (such as UWs and MDs) in the current state of the art of human-machine interfaces.

However, the language-independency commitment must also be understood in a far much deeper and much more intricate way. It is not only a matter of labeling, but of choosing what is supposed to be labeled. Spelling differences ('color' and 'colour') and synonyms ('freedom' and 'liberty') should clearly not be represented as different lexical items in XUNL. The set of KVs should be equivalent to the set of synonyms of a given language instead of to the whole set of words of that language. In this sense, KVs would be very akin to the concept of synset devised by the WordNet [10].

Moreover, XUNL should comprise only lexical roots (monomorphemic stems), i.e., the set of atomic lexical items necessary and sufficient to generate the whole set of words of a given language. For instance, there is no need, in XUNL, for a word like "beautiful" or "beautifully", provided that we have "beauty" and some derivation rules. This is to say that the XUNL lexicon should be generative, instead of enumerative.

Finally, XUNL should include only the semantic elementary particles of lexical meaning. Natural language words should be represented as complex semantic structures to be analyzed in XUNL. Accordingly, a verb like "to fly" should be rather represented as "to travel through air", (or even more radically as "to change location through air"), and a noun like "chair" should be represented as "a seat for one person, with a support for the back". Natural language lexical items should not be simply translated in XUNL but truly defined in relation to a core minimum vocabulary, as simple and small as possible.

B. Culture-independency

There seems to be compelling evidence that human knowledge is not organized in tree-like deep hierarchies but in a quite different topology, consisting of prototypes, exemplars and family resemblance relationships that are mainly contingent (context-dependent) rather than essential or necessary [11, 12]. It is ineffective for instance, to state that "apple" is a kind of "fruit" because, in many different contexts, it is not. In sentences like (16) below, where only a part of the meaning of apple is actually activated, knowledge that apples can be considered types of fruits is not only superfluous but even misleading.

(16) This ball of yam looks like an apple.

The fact is that concepts are exceedingly complex and fluctuating structures that, due to analogical reasoning, often assume unpredictable meanings. It is hopeless, therefore, and often useless, to build static ontologies, which will never be able to portrait the markedly different structure of human knowledge. Instead, the set of KVs should constitute a huge distributed network of associations molded by experience rather than logic.

In order to avoid categorization biases and ontology shortcomings, the set of KVs should be defined, not in a knowledge base, but in an example base (a "memory") automatically extracted out of real texts. This memory could bear, in principle, the same formal structure of the UNLKB, i.e., a set of directed binary relations between KVs associated to a given degree of "certainty", which would have to be understood as "frequency of occurrence"; however, it would be formed not out of human insights on classes and classification principles, but out on actual co-occurrence in a given corpus. Additionally, this memory would have to be incremental, since the frequency of occurrence would be permanently revised as new data are processed.

The two major issues in the XUNL Example Base are the corpus and the machine learning strategies. As for the former, we shall admit that the larger the better. The ideal corpus would comprise every available document, so that relations

between KVs would be set as broadly as possible. However, it is unlikely that one would be able to process so many data in order to extract recurrent edges between KVs. A more realistic approach would recommend the notion of "archive", in a sense very close to the one intended by Foucault [13], or that of "norm", according to Coseriu [14]. In both cases, we would acknowledge the fact that some texts are more authoritative than others, and should have, therefore, a more prominent role in validation.

The second problem – related to (unsupervised) machine learning – has already received considerable attention in Artificial Intelligence and can be easily adapted to the process of mechanically extracting co-occurrence relations in a given corpus. Clustering [15] and neural networks [16] have proved to exemplify interesting possibilities for pattern extraction and classification in large amount of data.

C. *Human-independency*

At last, we should stress that XUNL will be able to embody human knowledge and to occupy the places of source and target language in machine translation systems if, and only if, XUNL is self-consistent and human-independent. In order to "mean" to a machine, XUNL should not mean to a human being, who operates in a completely different way, from a completely different structure.

XUNL should not rely on definitions derived from human comprehension or an external language that is not replicable by the machine, but, instead, should shape its own world, a purely intensional (non-mental) dimension, a sort of electronic (possible) world, which would represent the sense and the reference of XUNL words and expressions. This digital (and artificial) world, and not the human analogical one, would figure as the "aboutness" of XUNL, and would comprise the truth-condition requirements for XUNL expressions to be "meaningful".

This is to say that the set of KVs should constitute a sort of sign system where the value of a given sign should solely derive from its position in the network. At least at the lexical level, XUNL should consist of "un système où tout se tient" [17], following hence the structuralist approach that "every language is a system, all parts of which organically cohere and interact [... where] no component can be absent or even different, without transforming the whole" [18].

Accordingly, KVs would only bear a negative (relational) definition, which would not necessarily coincide with the positive normally ascribed by a human. The value of a given KV would be the sum, and nothing but the sum, of all relations in which it takes part in the XUNL Example Base.

VII. Final remarks

It may seem clear by now that this paper is rather prescriptive than descriptive. Our main goal here was not to describe the existing syntax and semantics of Knowledge Vertices, but to draw some general guidelines that should be used in building them. The rules are seven and they are the following:

I. KVs should be represented by Arabic numerals (instead of English words).

II. KVs should be equivalent to sets of synonyms (instead of individual words).

III. KVs should be equivalent to generative lexical roots (instead of inflected or derived forms).

IV. KVs should be equivalent to the elementary particles of meaning (instead of complex semantic structures).

V. KVs should not be organized (or defined) according to single human-crafted ontology.

VI. KVs should be organized (and defined) according to unsupervised machine learning procedures operating over a selective example base.

VII. KVs should not bear any positive (non-relational) meaning.

VIII. Implementing such a lexical database is the first step and the first challenge in the development of XUNL.

References

[1] www.ronaldomartins.pro.br/unlx

[2] H. Uchida, M. Zhu, and T. Della Senta, *A gift for a millennium*, IAS/UNU, Tokyo, 1999.

[3] J. F. Sowa, "Conceptual Structures: Information," in *Processing in Mind and Machine*, Addison-Wesley, Reading, MA, 1984.

[4] J. F. Sowa, *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, Brooks Cole Publishing Co., Pacific Grove, CA, 2000.

[5] O. Lassila and R.R. Swick, *Resource Description Framework (RDF): model and syntax specification,* W3C Recommendation, 1999.

[6] *UNL*, www.undl.org/unlsys/unl/unl2005/

[7] J. Katz and J. Fodor, "The structure of a semantic theory", *Language*, 39, pp. 170-210, 1963.

[8] S. Nirenburg, V. Raskin et al., "On knowledge-based machine translation," in *Proceedings of the 11th International Conference on Computational Linguistics*, Bonn, 1986.

[9] *WordNet*, wordnet.princeton.edu

[10] C. Fellbaum, *WordNet: An Electronic Lexical Database,* Cambridge, MA: MIT Press, 1998.

[11] D. L. Medin and A. Ortony, "Psychological Essentialism," in S. Vosniadou, A. Ortony, (eds.) *Similarity and Analogical Reasoning,* Cambridge: Cambridge University Press, 1989.

[12] E. Rosch, "On the Internal Structure of Perceptual and Semantic Categories", in T. E.Moore (Ed.), *Cognitive Development and Acquisition of Language*, New York: Academic Press, 1973.

[13] M. Foucault, *L'Ordre du discours,* Paris, Gallimard, 1971.

[14] E. Coseriu, *Teoría del lenguaje y lingüística general*, Madrid, Gredos, 1962.

[15] S. Kotsiantis and P. Pintelas, "Recent Advances in Clustering: A Brief Survey," *WSEAS Transactions on Information Science and Applications*, Vol 1, No 1 (73-81), 2004.

[16] G. Hinton and T. Sejnowski (eds.), *Unsupervised Learning and Map Formation: Foundations of Neural Computation*, MIT Press, 1999.

[17] F. de Saussure, *Cours de Linguistique Générale*., Paris, Payot, 4 ed. Trad. bras. São Paulo, Cultrix, 1969.

[18] H. G. Gabelenz, *Die Sprachwissenschaft*, 1901.

Extracción automática de los patrones de rección de verbos de los diccionarios explicativos

Noé Alejandro Castro-Sánchez y Grigori Sidorov

***Resumen*—En este trabajo se propone el uso de métodos simbólicos para la extracción de las valencias semánticas de verbos describiéndolas bajo el concepto de patrones de rección de la teoría Significado ⇔ Texto. El método se basa en el procesamiento automático de las definiciones de verbos contenidas en diccionarios explicativos y en el análisis de relaciones semánticas, principalmente de inclusión y de sinonimia, establecidas entre ellos. Partimos de la hipótesis de que las definiciones lexicográficas existentes en diccionarios explicativos deben proporcionar la suficiente información para identificar los actantes de verbos. Los resultados obtenidos demuestran que, a pesar de que en muchas de las definiciones no es posible encontrar información relativa a la estructura argumental de los verbos, es posible deducirla identificando y analizando las definiciones con las que existan relaciones sinonímicas y de inclusión.**

***Palabras clave*—Actantes, sinónimos, marcos de subcategorización, valencias, diccionarios explicativos.**

Automatic Extraction of Semantic Valences of Verbs from Explanatory Dictionaries

Noé Alejandro Castro-Sánchez and Grigori Sidorov

***Abstract*—In this work we propose the application of symbolic methods for extraction of semantic valences of the verbs describing them under the Government Pattern concept of the Meaning ⇔ Text Theory. The method is based on the automatic processing of the definitions of verbs used in Explanatory Dictionaries and the analysis of semantic relationships, as inclusion and synonymy, given among them. We believe that lexicographic definitions of Explanatory Dictionaries supply enough information for identifying verb actants. The obtained results show that even when it is not possible to find information related to the argument structure of verbs in the definitions, it is possible to deduce it identifying and analyzing other definitions which semantic relationships are established.**

***Index terms*—Actants, synonyms, subcategorization frames, valences, explanatory dictionaries.**

Manuscrito recibido el 14 de febrero de 2012, manuscrito aceptado el 7 de mayo de 2012.

Los autores trabajan en el Centro de Investigación en Computación, Instituto Politécnico Nacional, México DF (email: noe.acastro@gmail.com, sidorov@cic.ipn.mx).

I. Introducción

La gramática tradicional considera a la oración como una estructura bimembre, formado por sujeto y predicado. Sin embargo, el lingüista francés Lucien Tesnière propuso en 1959 [25] representar a la oración como una estructura jerárquica, y no binaria, donde el verbo ocupa la posición central, determinando los papeles que desempeñan el resto de elementos en la oración.

Todas las palabras que conforman la oración, establecen relaciones en donde algunas de ellas establecen o determinan propiedades de otras. Estas relaciones son denominadas "relaciones de rección". Los elementos regidos por un (o dependientes del) verbo se consideran complementos en la construcción del significado del verbo.

El hecho de regir o requerir una o varias palabras, se le denomina "régimen". De esta manera tenemos el régimen verbal, y el régimen preposicional: el primero hace referencia a la exigencia del verbo de ir o no acompañado por un elemento subordinado (régimen transitivo, y régimen intransitivo respectivamente), y el segundo señala la exigencia de una forma específica de la preposición a utilizar: "inducir a", "convertirse en", "depender de", etc.

La manera de nombrar a estos elementos que se espera acompañen a un verbo para lograr construir una oración gramatical e inteligible, varía de acuerdo al formalismo teórico que los procesa. En el enfoque teórico de constituyentes, se conocen más ampliamente con el nombre de 'marcos de subcategorización' (en inglés, *subcategorization frames* o SCF). Dentro del formalismo de dependencias, se conocen como "actantes". Bajo este formalismo pero en la "Teoría Significado ⇔ Texto" son conocidos bajo el nombre de "patrones de rección" [14].

En este trabajo de investigación identificamos de manera automática los actantes de los verbos a través del procesamiento automático de las definiciones contenidas en diccionarios explicativos y del análisis de las relaciones semánticas que ocurren entre éstos, apoyándonos en el enfoque teórico de la teoría 'Significado ⇔ Texto'.

En las secciones II y III haremos una revisión sobre los trabajos que se han desarrollado para la identificación automática de la valencia verbal y explicamos la metodología general que se ha utilizado. En la sección IV explicamos en qué se basa el método y en las secciones V, VI y VII abordamos a grandes rasgos los algoritmos que implementamos. Finalmente en la sección VIII hacemos una

descripción de los resultados obtenidos. Al final presentamos las conclusiones.

II. Trabajos relacionados

La recopilación de información de los complementos de los verbos fue una idea originalmente sugerida por el lingüista Noam Chomsky, y que se ha ido implementado por las teorías sintácticas subsecuentes.

El diseño pionero de la extracción automática de esta información, corresponde a Michael Brent [4], quien propone el desarrollo de un programa que toma texto de un corpus no etiquetado como única entrada para identificar SCF, extrayendo primeramente los verbos contenidos en él, y a continuación, frases que representen a los argumentos de los verbos.

En este trabajo Brent identificó cinco SCF, utilizando una técnica basada en el "Filtro de Casos de Rouvret y Vergnaud". A través de este filtro se identificaron los verbos potenciales, buscando, por ejemplo, palabras que contengan o carezcan del sufijo *–ing* (equivalente en español del gerundio *–ando* y *-endo*) o que sigan a un determinante o una preposición diferente a *to*. Por ejemplo, *was walking* (*estaba caminando*) se puede considerar como verbo, pero *a talk* (*una plática*) no.

En un segundo trabajo [5], identificó seis marcos sintácticos. En éste Brent incorporó un modelo estadístico en el cual se mide la frecuencia de aparición de claves con los verbos para cada uno de los marcos, así como el número de veces que cada verbo ocurre.

Posteriormente Ushioda [26] propone hacer uso de sentencias parseadas no completamente, derivadas de un corpus etiquetado. El sistema que elaboró es capaz de reconocer y calcular las frecuencias relativas de 6 marcos de subcategorización, los mismos trabajados por Brent. El proceso consiste en extraer del Corpus etiquetado las sentencias que contienen un verbo y dividir el sintagma nominal en pequeños fragmentos (*chunks*) utilizando un parseador de estados finitos, así como el resto de palabras usando un conjunto de 16 símbolos y categorías frasales. A estas sentencias les es aplicado un conjunto de reglas de extracción de marcos de subcategorización. Estas reglas están escritas como expresiones regulares y se obtienen a través de la extracción de ocurrencias de una pequeña muestra de verbos en un texto de entrenamiento.

Manning [16] propone un sistema más ambicioso capaz de reconocer 19 marcos sintácticos diferentes. Los marcos sintácticos se obtienen a través de un programa que procesa la salida de un etiquetador estocástico de partes de la oración (*part-of-speech tagger*) ejecutado sobre el corpus a analizar. El programa consta de dos partes: un parseador de estados finitos que analiza el texto etiquetado buscando un verbo, y que al encontrarlo, divide toda la información que lo sigue en pequeños componentes o chunks, hasta encontrar algún elemento reconocido como terminador de argumentos subcategorizados.

La segunda parte del programa, consiste en la reducción del ruido, para lo cual se utilizó el mismo filtro estadístico usado por Brent: el ruido (o pistas falsas), puede ser eliminado observando qué marcos aparecen con un verbo en una frecuencia razonablemente superior a la que pudiera considerarse casualidad (adjuntos) o errores en la detección.

Monedero *et al*. [19], inspirados en el trabajo de Brent y Manning, desarrollaron una herramienta para obtener marcos sintácticos de verbos en español.

El trabajo realizado, denominado SOAMAS, consistió en generar tres gramáticas: la primera de ellas encargada de identificar verbos principales y auxiliares, así como posibles conjunciones y preposiciones. La segunda realizada con el fin de reconocer sintagmas nominales, adjetivos y preposicionales. La tercera consistió en ser la encargada de identificar los complementos verbales.

El principal problema enfrentado para entonces, consistió en la carencia de corpus etiquetados para el español suficientemente extensos (dispusieron sólo de 10,000 palabras etiquetadas), lo que imposibilitó llegar a resultados confiables.

III. Metodología usada en trabajos previos

Los trabajos antes mencionados siguen una metodología de procesamiento como la expuesta en [7] y [22], en la que es posible distinguir los siguientes puntos:

1. *Selección y preparación del corpus*: indica la elección del corpus en el que se va a realizar la identificación de SCF, y, en caso de no estar anotado, el tipo de etiquetado que se le realizará (gramatical, sintáctico, etc).
2. *Detección de marcos*: establece el método computacional a seguir para identificar los SCF.
3. *Filtrado estadístico*: determina el método para eliminar el posible ruido obtenido en el paso previo.

En la *selección y preparación del corpus* se trabaja en considerar tanto el tipo como el tamaño de los corpus a procesar, pues estos factores pueden provocar variaciones en cuanto a los resultados que se obtienen. En general, los investigadores prefieren contar con la mayor cantidad de información (texto) posible, ya que de esta manera aseguran una muestra más representativa del idioma en el que se esté trabajando.

En [20] y [21] se expone cómo diferentes géneros de corpus provocan variaciones en las frecuencias de SCF. En [21] se estudiaron cinco corpus diferentes, dos de los cuales fueron obtenidos de fuentes psicológicas (caracterizados principalmente por contener sentencias aisladas), y los tres restantes fueron el "Brown corpus", "Wall Street Journal corpus" y el "Switchboard corpus". Las diferencias reportadas se encontraron tanto en los tipos de SCF como las frecuencias de los tipos de SCF.

La presentación del corpus tocante a la anotación de información lingüística, determinará la manera en que se procederá para ejecutar la tarea de extracción de SCF. Brent utiliza un corpus no anotado al cual aplica claves morfosintácticas para detectar verbos y sus posibles marcos. Ushioda propone utilizar sentencias parseadas sólo parcialmente, derivadas de un corpus ya etiquetado, y a las

cuales les es aplicado reglas escritas como expresiones regulares. Manning aplica un etiquetador estocástico sobre el corpus a analizar y así extraer todas aquellos componentes de la oración que tengan elementos reconocidos como terminadores de marcos. Gahl extrae subcorporas a través de la ejecución de expresiones regulares sobre el BNC para detectar en ellos a los posibles marcos.

La *detección de marcos* en general se ha realizado a través del *emparejamiento de patrones*, que consiste en definir a priori información gramatical que pudiera considerarse relevante para identificar alguna combinación de elementos léxicos como candidados a SCF. Posteriormente se busca en el corpus información que pudiera emparejarse con los patrones predefinidos.

La adquisición de los posibles marcos realizada por el proceso previo, no está exenta de errores, como es de esperarse. La información obtenida contiene ruido que puede derivarse de errores en la fase de etiquetado gramatical, por ejemplo, o incluso, errores en la fase de detección de SCF provocada por una ineficiencia en la discriminación de adjuntos.

Para remover toda la información no deseada, se realiza un procesamiento estadístico. En suma, se busca determinar si un candidato a SCF de un verbo en particular debe realmente considerarse como tal o no. Los métodos estadísticos para realizar el filtrado de información se hacen usualmente con la "prueba de hipótesis" (*hypothesis test*). Esta prueba consiste en establecer una hipótesis nula H_0, como verdadera, a menos que los datos sugieran lo contrario, lo cual provoca que se rechace la hipótesis y entonces se acepta como verdadera una hipótesis alternativa H_1. En el contexto de la adquisición de SCF, H_0 se considera como una falta de asociación entre un determinado verbo y un SCF, y H_1 como la afirmación a dicha asociación. Se establece la prueba como de *una cola*, dado de que la hipótesis alternativa establece una dirección, en este caso la correlación positiva entre el verbo y el marco. En seguida se calcula el valor estadístico de prueba con los datos de la muestra, lo que sirve para decidir si H_0 es verdadera o falsa. Esto se realiza comparando la probabilidad esperada de que exista correlación si H_0 es verdadera, con la probabilidad observada de coocurrencia. Si esta última es mayor que la primera, la hipótesis H_0 es rechazada.

IV. Método propuesto

La identificación de actantes de verbos se basa comúnmente en la aplicación de métodos estadísticos aplicados a corpus, analizando patrones de ocurrencia de eventos de acuerdo a la frecuencia de uso en el lenguaje.

En este trabajo se propone el uso del diccionario explicativo para su procesamiento, empleando una serie de heurísticas basadas en observaciones a priori de la naturaleza y comportamiento de los datos contenidos en las definiciones lexicográficas para la identificación de la valencia verbal. De la variedad de diccionarios que existen, nos enfocamos en aquellos dirigidos a los hablantes nativos de un idioma (monolingües), que no presentan restricciones de dominio (generales) y que presentan la definición semántica de los vocablos que contienen (explicativos). En particular se eligió el Diccionario de la Real Academia de la Lengua Española (DRAE), considerado como el de mayor resonancia en países hispanohablantes.

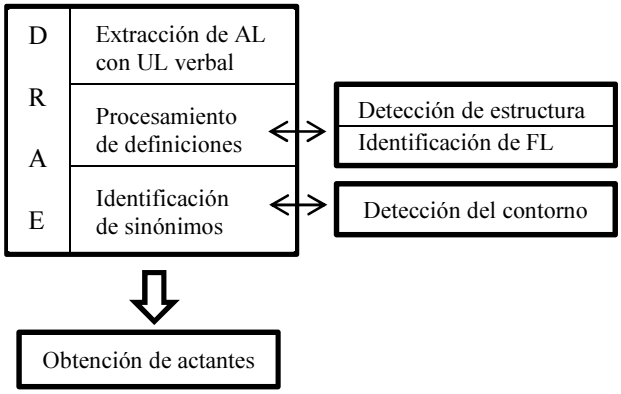


Fig. 1. Arquitectura del proyecto.

En general, los diccionarios presentan secciones textuales dispuestas ordenadamente denominadas "Artículos Lexicográficos" (AL), conformados por una entrada o "Unidad Léxica" (UL) y la información que la define o describe. Esta información se presenta ya sea como definición propia o perifrástica, cuando expresa el significado de las entradas en cuanto a su contenido léxico-semántico, o como definición impropia, cuando se utiliza para describir o explicar el funcionamiento y empleo de palabras funcionales, debido a su falta de un verdadero significado léxico, véase fig. 1.

La estructura de las definiciones propias suele seguir la norma establecida por la llamada definición aristotélica, la cual consiste de un enunciado encabezado por un término genérico o hiperónimo inmediato (genus), seguido de una diferencia específica (differentia), o conjunto de rasgos y características que permiten distinguir el término definido de otros que se agrupan bajo el mismo hiperónimo.

Esta posición predecible de los elementos que conforman las definiciones propias, genus + differentia, permite utilizar heurísticas que puedan aplicarse para identificarlos de manera automática. Además de estos elementos, se sabe ([6]) que en algunas definiciones lexicográficas es posible identificar elementos que proporcionan información sobre la estructura argumental de las UL's, relacionada con restricciones contextuales o algunos usos sintácticos. Esta información se denomina "contorno de la definición".

El contorno no siempre es señalado explícitamente en los diccionarios lexicográficos, posiblemente porque resultaría redundante para los nativos del idioma, aunque suele ser importante para saber hacer un uso correcto de la UL definida. El diccionario "Salamanca de la Lengua Española", por ejemplo, hace un señalamiento del contorno en sus definiciones:

Derivar. *Ser < una cosa > consecuencia de [otra cosa]*

Los sujetos son rodeados con los signos mayor y menor que, y los complementos entre corchetes.

Por otro lado, en el DRAE el contorno se indica encabezando la definición con la fórmula "Dicho de":

***Convalecer**. Dicho de una persona: salir del estado de postración o peligro...*

Esto indica que el verbo *Convalecer* selecciona como sujeto alguna *persona*. Lamentablemente, en este diccionario no todas las definiciones de verbos vienen acompañadas por la especificación del sujeto, y definitivamente no es posible encontrar indicaciones sobre el resto de complementos en las definiciones. Por ejemplo:

***Bajar.** Poner **algo** en un lugar inferior a aquel en que estaba*

En este caso, el objeto directo del verbo (*algo*) aparece en la definición, pero no es acompañado por alguna marca que logre identificarlo como complemento del verbo.

En algunos otros casos, algunos o todos los complementos del verbo definido no forman parte de la redacción de la definición:

***Controlar.** Ejercer el control*

Este tipo de casos representan un verdadero reto para la identificación automática del contorno en las definiciones.

V. Identificación de los componentes de las definiciones

En primer lugar se realizó un preprocesamiento de datos que consistió en extraer únicamente los artículos lexicográficos que eran de relevancia para este trabajo, es decir, unidades léxicas verbales y sus respectivas definiciones. Se utilizó la herramienta de análisis de texto de código abierto para varios idiomas, Freeling [2], como etiquetador de partes de la oración (POST, por sus siglas en inglés) para conocer la categoría gramatical de cada palabra de los datos seleccionados.

Con esta información fue posible hacer un primer intento para identificar el genus de la diferencia específica. Se hizo un primer análisis de tipo manual para identificar algunos patrones que pudieran ayudar a automatizar el proceso para atender todos los casos del dicionário. Esto arrojó las diferentes maneras en que es posible encontrar el genus, lo cual puede resumirse en lo siguiente:

1) *Verbos individuales:*
   a. Con un solo verbo. Ejemplo:
      ***Cotizar**. Pagar una cuota.*
   b. Dos o más verbos enlazados por conjunciones y/o disyunciones. Ejemplo:
      ***Armonizar**. Escoger y escribir los acordes correspondientes a una melodía.*
      ***Aballar**. Amortiguar, desvanecer o esfumar las líneas y colores de una pintura.*
2) *Como cláusula subordinada en infinitivo cumpliendo la función de complemento directo. Ejemplo:*
   ***Gallear**. Pretender sobresalir entre otros con presunción o jactancia.*
3) *Como Función Léxica. Ejemplo:*
   ***Anunciar**. Dar publicidad a algo con fines de propaganda comercial.*

Cada caso particular requiere un tratamiento diferente que permita su correcta identificación. En el caso 1 y 2, todo verbo existente como cabecera de la definición se considera genus de la UL definida. En 3) se requiere un procesamiento más complejo: los verbos que vienen acompañados por un sustantivo son Funciones Léxicas (FL) potenciales. Las FL se definen como una función que asocia una palabra denominada "base", la cual aporta su significado literal a la expresión, a otra llamada "colocador", que adquiere un significado diferente de su significado típico, de tal manera que el significado del conjunto incluye el significado de una de las palabras (base), pero no del otro (colocador) [12]. De esta manera, el genus en una definición que es encabezada por una FL no puede ser el colocador.

Siendo posible identificar el genus en la definición, el resto de elementos que la constituyen automáticamente son tomados como parte de la diferencia específica.

VI. Procesamiento del contorno

Observaciones de las definiciones mostraron que el contorno se conforma por sustantivos comunes (NC) y pronombres indefinidos (PI), lo que condujo a la elaboración de una heurística que identifica los segmentos de las definiciones constituidos por palabras con estas categorías gramaticales.

El algoritmo desarrollado se basa en una serie de reglas que reflejan la estructura básica de las definiciones, más concretamente, de la diferencia específica, que permiten capturar incluso el contexto sintáctico que delimita cada elemento del contorno, ayudando a conocer por ejemplo las preposiciones con las que puede acompañarse.

Las reglas quedan definidas de la siguiente manera:

1) *La nomenclatura utilizada se define en la tabla I.*

Tabla I. Definición de los símbolos utilizados en la gramática

Símbolo utilizado	Significado
Diff	Differentia
Cont	Contorno
Nuc	Núcleo del contorno (PI ó NC)
EleIzq	Elementos a la izquierda
EleDer	Elementos a la derecha
EIzq	Elemento izquierdo
EDer	Elemento derecho
DA, DI, DO, DP, CS, RG, Z, AQ, RN, CC, FC	Etiquetas asignadas a palabras para indicar su información morfológica, propuestas por el grupo EAGLES para la anotación morfosintáctica de lexicones y corpus

2) *El lado izquierda de la primera producción, es el símbolo inicial.*

3) *Reglas:*
Diff → Cont
Cont→ Nuc | EleIzq Nuc | EleIzq Nuc EleDer | Nuc EleDer | Cont Liga Cont
Nuc → PI | NC
EleIzq → EIzq | EIzq EleIzq
EleDer→ EDer | EDer EleDer
EIzq→ DO | DA | DI | DP | DD | SP | CS | RG | Z | AQ
EDer → AQ | RN
Liga → CC | FC

Estas reglas no se utilizan en la producción de oraciones (pues podrían generar oraciones agramaticales como un nombre común acompañado por una sucesión ininterrumpida de preposiciones), sino en la segmentación de las definiciones, donde cada segmento está conformado por un único candidato a elemento del contorno.

Un ejemplo de la aplicación de estas reglas en las definiciones, es el siguiente:

Poner. *Colocar en un sitio o lugar a alguien o algo*
Segmentación: *en un sitio o lugar* | *a alguien o algo*

VII. Adquisición de sinónimos

Para redactar las definiciones de verbos, probablemente los lexicógrafos no toman un criterio unificado sobre el uso o no del contorno asociado a los verbos, ni sobre el número de elementos del contorno que se puedan utilizarse en las definiciones. Es decir, existirán definiciones que aporten mayor información en este rubro, que otras. Debido a esto, lo que hemos propuesto es utilizar las definiciones de otros verbos para complementar la información faltante en casos donde sea necesario. Esta selección de verbos no se realiza de manera aleatoria, sino que se basa en las relaciones semánticas dadas entre verbos, como la sinonimia y las relaciones de inclusión.

La identificación de los verbos relacionados entre sí por sinonimia, se realiza de la siguiente manera: el diccionario de la RAE emplea el tipo de definición sinonímica recurrentemente, la cual consiste en utilizar como definición una o varias palabras con la misma categoría gramatical que la UL definida. Por ejemplo, el verbo "Coger" se define como:

Coger. *Asir, agarrar o tomar*

Lo que significa que la definición de "Coger" puede encontrarse en la definición de los verbos "asir", "agarrar" o "tomar". Este tipo de definiciones puede provocar círculos viciosos, lo cual es considerado como un defecto por los lexicógrafos, sin embargo, este comportamiento beneficia nuestra tarea. Un ejemplo de círculo vicioso es el conformado entre los verbos "coger, asir, agarrar y tomar", mostrado en la siguiente gráfica. El inicio de cada flecha indica la UL definida, y el nodo al que apunta la UL que se utiliza como sinónimo en su definición.

Vemos un ejemplo de los círculos viciosos. Las definiciones que componen cada verbo, son las siguientes:

- ***Coger***. *Asir, agarrar o tomar*
- ***Agarrar***. *Coger, tomar.*
- ***Tomar***. *Coger o asir con la mano algo.*
- ***Asir***. *Tomar o coger con la mano, y, en general, tomar, coger, prender.*

Al ser considerados estos verbos como sinónimos, significa que pueden sustituirse indistintamente en al menos algún sentido de los varios que tienen atribuidos. Siendo así, se deberían cumplir los siguientes dos supuestos:

1. El número de actantes de cada verbo es el mismo para cada uno de sus sinónimos (en al menos un sentido),
2. Las restricciones semánticas que un verbo impone a sus actantes, son las mismas que el resto de sus sinónimos (en al menos un sentido).

De cumplirse los puntos previos, permitiría subsanar en la medida de lo posible la falta de información referente al contorno que suele existir en las definiciones de verbos en el diccionario de la RAE, combinando el contorno de las definiciones que aparecen en un conjunto de sinónimos. Para ello, en primer lugar, se debe distinguir qué sentido en específico logra la relación sinonímica de los verbos. Por ejemplo, el verbo "abatir" en el sentido 6 incluye como sinónimos en su definición los verbos "desarmar" y "descomponer". Ambos verbos disponen de varios sentidos, de entre los cuales es necesario distinguir cuáles son los que los relacionan como sinónimos. La solución que en este trabajo se implementó consiste en buscar en las definiciones algún hiperónimo común a los verbos, lo que indicaría que existe relación semántica en ese sentido en específico.

En las tablas II y III, se muestran los hiperónimos de los primeros 5 sentidos de los verbos "desarmar" y "descomponer", respectivamente. Se observa que el sentido

TABLA II.HIPERÓNIMOS DEL VERBO *DESARMAR*

Num. sentido	Hiperónimo
1	*Quitar, hacer entregar*
2	*Desnudar o desceñir*
3	*Reducir*
4	*Dejar*
5	*Desunir, separar*

TABLA III. HIPERÓNIMOS DEL VERBO *DESCOMPONER*

Num. sentido	Hiperónimo
1	*Desordenar y desbaratar*
2	*Separar*
3	*Indisponer*
4	*Averiar, estropear, deteriorar*
5	*Corromperse*
…	…

5 de “desarmar” y el sentido 2 de “descomponer” comparten el mismo hiperónimo.

Teniendo identificados los sentidos relacionados semánticamente, pueden combinarse los contornos de las definiciones para complementar la información faltante en algunas de ellas. En esta tarea pueden identificarse los siguientes casos:

1) No existe información alguna del contorno en alguna definición, pero sí en las otras. Retomando las definiciones de los verbos “coger”, “agarrar”, “asir” y “tomar”, observamos que la definición del verbo “coger” sólo incluye sinónimos, sin hacer mención alguna del contorno. Sin embargo, la definición del verbo “tomar” incluye dicha información. El resultado de la obtención de segmentos de la definición es:

Tomar. *Coger o asir con la mano algo*
Segmentación*: con la mano* | *algo*

Por lo tanto, el contorno del verbo “tomar” se considera también perteneciente al verbo “coger”.

2) Algunas definiciones incluyen segmentos que no pertenecen al contorno. Este es el caso más común, y es complicado lograr una correcta discriminación de segmentos. Por ejemplo:

Llevar. *Conducir algo desde un lugar a otro alejado de aquel en que se habla o se sitúa mentalmente la persona que emplea este verbo.*
Segmentación: *algo* | *desde un lugar* | *a otro* | *mentalmente la persona* | *este verbo*

En esta definición, los segmentos “mentalmente la persona” y “este verbo”, no son elementos que formen parte del contorno y que por lo tanto reflejen a los actantes del verbo.

VIII. Resultados experimenatales

Después de procesar todas las definiciones de verbos encontramos poco más de 6,000 definiciones sinonímicas. Estas 6,000 definiciones se procesaron para identificar si existía algún genus común a las definiciones de los verbos agrupados y así precisar el número del sentido en que se relacionaban. Esto llevó a la identificación de un aproximado de 6,500 grupos de sinónimos en donde se identificaron explícitamente los sentidos. Por ejemplo, el verbo “amparar” en su sentido 4 se define como: “Defenderse, guarecerse”. Estos verbos usados en la definición, ambos en su sentido 2, se definen como:

Defender (2): *Mantener, conservar, sostener algo contra el dictamen ajeno.*
Guarecer (2): *Guardar, conservar y asegurar algo*

Ambas definiciones comparten el verbo *conservar*, por lo que en ese sentido en particular conforman un grupo de sinónimos con sentido identificado. Sin embargo, observamos también que *defender* en su sentido 1 y *guarecer* en su sentido 4 se definen como:

Defender (1): *Amparar, librar, proteger*
Guarecer (4): *Socorrer, amparar, ayudar.*

Conformarían un nuevo grupo en dichos sentidos bajo el verbo *amparar*. Del ahora total aproximado de 6,500 grupos conformados por verbos en un sentido en particular, en 3,000 agrupaciones no se lograron identificar los sentidos que relacionaban a los verbos siguiendo el criterio del genus común. De los 6,500 grupos, cerca de 500 grupos no ofrecen ningún candidato a contorno. Por ejemplo:

Abrasar (3): *Calentar demasiado.*
Quemar (2): *Calentar mucho.*

Por otro lado, considerando que no todos los sustantivos comunes y pronombres indefinidos que aparecen en una definición pueden ser catalogados como elementos del contorno (ver apartado 5.1), decidimos procesar aquellas definiciones cuyos candidatos a elementos del contorno estuvieran conformados únicamente por los pronombres indefinidos “algo, alguien”, y los sustantivos comunes “cosa, persona, animal, lugar” y “parte”, ya que al realizar una medición de las categorías gramaticales de palabras funcionales más frecuentemente utilizadas en las definiciones, las palabras antes mencionadas tuvieron mayor presencia (Tabla IV).

Tabla IV. Elementos de controno más frecuentes

Palabra	Frecuencia
Algo	3,000
Alguien	2,000
Otro	900
Cosa	800
Parte	500
Persona	400
Lugar	350
Cuerpo, acción, fuerza, agua, tierra, …	< 300

Por otro lado, estas palabras representarían en cualquier ontología el nivel más alto o abstracto de los grupos que la componen.

El procesamiento de estos datos nos arrojó un total de 420 grupos de sinónimos que contienen dichas palabras en sus funciones.

La cantidad de verbos que se lograron detectar en este último grupo, fue de 280 verbos, y de estos, se lograron identificar 390 sentidos en total.

Varios grupos incluyen el mismo sentido de algún verbo. Al existir intersección entre ellos, podemos proceder a la unión de grupos, y así muy posiblemente, complementar de manera más precisa la información de los diferentes verbos y sobre todo de su contorno.

TABLA V. ESTADÍSTICAS DE SINÓNIMOS

Elemento evaluado	Cantidad
Definiciones sinonímicas	6,000
Grupos de sinónimos con sentidos de verbos identificados	6,500
Grupos de sinónimos donde no se identificaron los sentidos de verbos	3,000
Grupos de sinónimos donde no se identificaron candidatos a contorno	500
Grupos de sinónimos con candidatos a contorno más abstractos	420

Por ejemplo, consideremos el siguiente grupo de sinónimos tomados de la definición del verbo "maliciar" en su primer sentido:

Maliciar (1): *Recelar, sospechar, presumir algo con malicia*

Los verbos "recelar" y "sospechar" coinciden en usar el mismo genus en sus sentidos 1 y 2 respectivamente:

Recelar (1): *Temer,* ***desconfiar*** *y sospechar*
Sospechar (2): Desconfiar*, dudar, recelar de alguien*

Combinamos las definiciones de ambos verbos en los sentidos antes indicados y el contorno resultante es "de alguien".

Por otro lado, "recelar" y "dudar" son también sinónimos según el segundo sentido de "sospechar". Ambos verbos son definidos en los sentidos abajo indicados, nuevamente bajo el genus "desconfiar", de la siguiente manera:

Recelar (1): *Temer,* ***desconfiar*** *y sospechar*
Dudar (2)*:* ***Desconfiar****, sospechar de alguien o algo*

La identificación del contorno en ambas definiciones sería "de alguien o algo". Como ambos grupos de sinónimos incorporan el verbo "recelar" en un mismo sentido (1), entonces los unimos para conformar un solo grupo. De esta manera, tenemos que los verbos "recelar" (en 1), "sospechar" (en 2) y "dudar" (en 2) comparten el contorno "de alguien o algo".

En suma, por cada verbo en un sentido en particular, unimos todos los grupos de sinónimos que lo incluían y combinamos los contornos identificados.

La evaluación manual de los resultados dio 83% de precisión del método. Se evaluaron manualmente los contornos de 115 verbos.

IX. CONCLUSIONES

En este trabajo propusimos un método para la extracción de los actantes de verbos para el idioma español, basándonos en el análisis de las definiciones en diccionarios explicativos. Dado que la redacción de los artículos lexicográficos se apega a estructuras bien establecidas, es posible crear heurísticas para el análisis y extracción de información de ellos. Cada uno de los elementos que conforman estas estructuras, aportó datos relevantes para el cumplimiento de los objetivos propuestos.

En particular, el contorno de las definiciones de los verbos, al indicar condiciones sintagmáticas del verbo y recoger las restricciones de tipo semántico que sus argumentos requieren, se consideran una imagen de la valencia verbal. Así, la extracción del contorno se traduce en la obtención de información sobre los actantes del verbo.

La falta de una especificación rigurosa del contorno en la mayoría de las definiciones de los verbos, imposibilita conocer de manera certera sus valencias. Sin embargo, encontramos un recurso para complementar esta escasa información apoyándonos en las definiciones de otros verbos. Esto se hizo atendiendo las relaciones léxicas de inclusión (hiperonimia/hiponimia) establecidas entre los genus y los artículos lexicográficos y las relaciones de sinonimia que pueden encontrarse en las llamadas definiciones sinonímicas de las que hace uso el diccionario. A través de estas relaciones pudimos identificar qué sentidos de los verbos establecían relaciones de sinonimia con otros.

Considerando que los sinónimos pueden sustituirse mutuamente en cualquier contexto (bajo sentidos en específico) fue posible afirmar que bajo estas condiciones existe una coincidencia en la valencia verbal. Esta obtención de ciclos nos ayudó a completar la lista de actantes de cada verbo complementando la información que cada definición manejaba.

AGRADECIMIENTOS

El trabajo fue realizado con el apoyo parcial del gobierno de México (proyectos CONACYT 50206-H y 83270, SNI) e Instituto Politécnico Nacional, México (proyectos SIP 20111146, 20113295, 20120418, COFAA, PIFI), Gobierno del DF (ICYT-DF proyecto PICCO10-120) y la Comisión Europea (proyecto 269180).

REFERENCIAS

[1] Ch. Aone and D. MacKee, "Acquiring Predicate-Argument Mapping Information from Multilingual Texts," in *Corpus processing for lexical acquisition*, pp. 191–202, 1996.

[2] J. Atserias, B. Casas, E. Comelles, M. Gonzáles, and L. Padró, "FreeLing 1.3: Syntactic and Semantic Services in an Open-Source NLP Library," in *Fifth international conference on Language Resources and Evaluation*, Genoa, Italy nlp/freeling, http://www.lsi.upc.edu/nlp/freeling, 2006.

[3] I. Bolshakov, A. Gelbukh, *Computational Linguistics: Models, Resources, Applications*, 2004.

[4] M. Brent, "Automatic acquisition of subcategorization frames from untagged text," in *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, Berkeley, CA., pp. 209-214, 1991.

[5] M. Brent, "From grammar to lexicon: unsupervised learning of lexical syntax," *Computational Linguistics* 19(3): 243-262, 1993.

[6] M. Cordero, "Diccionario de la lengua española secundaria (DILES): Planta para su elaboración con algunos apuntes básicos de metalexicografía," *Káñina, Rev. Artes y Letras*, Univ. Costa Rica. XXXI (1): 167-195, ISSN: 0378-0473, 2007.

[7] R. Dale, H. Moisl, and H. Somers. *Handbook of Natural Language Processing*, ISBN: 0-8247-9000-6, 2000.

[8] *Diccionario de la Lengua Española*, Edición vigésimo segunda. www.rae.es, 2001.

[9] J. Fernández, *Rektion. Rección/Régimen.* http://culturitalia.uibk.ac.at, Hispanoteca, 2002.
[10] S. Fujita and F. Bond, "An Automatic Method of Creating Valency Entries using Plain Bilingual Dictionaries," in *The tenth conference on theoretical and methodological issues in machine translation*, Baltimore, Maryland, pp. 55-64, 2004.
[11] S. Gahl, "Automatic extraction of subcorpora based on subcategorization frames from a part-of-speech tagged corpus," in *Proc. of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, Montreal, Canada., pp. 428-432, 1998.
[12] A. Gelbukh, O. Kolesnikova. *Semantic Analysis of Verbal Collocations with Lexical Functions*. Studies in Computational Intelligence, N 414, Springer, 2012.
[13] D. Ienco, S. Villata., and C. Bosco, "Automatic Extraction of Subcategorization Frames for Italian," in *International Conference on Language Resources and Evaluation LREC*, 2008.
[14] S. Kahane, "Meaning-text theory," in Ágel, Vilmos et al. (eds.): *Dependency and Valency. An International Handbook of Contemporary Research*. Berlin, 2003.
[15] D. Kawahara and S. Kurohashi, "Case frame compilation from the web using high-performance computing," in *Proceedings of LREC2006*, 2006.
[16] C. Manning, "Automatic acquisition of a large subcategorization dictionary from corpora," in *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, Columbus, Ohio, pp. 235- 242, 1993.
[17] S. Marinov and C. Hamming, *Automatic Extraction of Subcategorization Frames from the Bulgarian Tree Bank*, 2004.
[18] A. Mendikoetxea, "En busca de los primitivos léxicos y su realización sintáctica: del léxico a la sintaxis y viceversa," *2º Xarxa Temàtica de Gramàtica Teòrica*, Barcelona, UAB, 2004.
[19] J. Monedero, J. González, J. Goñi, C. Iglesias, and A. Nieto, "Obtención automática de marcos de subcategorización verbal a partir de texto etiquetado: el sistema SOAMAS," *Procesamiento del lenguaje natural*, boletín 17, 1995.
[20] D. Roland, D. Jurafsky, "How Verb Subcategorization Frequencies Are Affected By Corpus Choice," in *Proc. of COLING/ACL-98*, pp. 1122-1128, 1998.
[21] D. Roland and D. Jurafsky, "Verb Sense and Verb Subcategorization Probabilities," in Stevenson, Suzanne, and Paola Merlo (eds.), *The Lexical Basis of Sentence Processing: Formal, Computational, and Experimental Issues*. Amsterdam: John Benjamins, pp. 325-346, 2002.
[22] S. Sabine, "The Induction of Verb Frames and Verb Classes from Corpora," in *Corpus Linguistics. An International Handbook*. Anke Lüdeling and Merja Kytö (eds). Mouton de Gruyter, Berlin, pp. 952–972. eBook ISBN: 978-3-11-021388-1. Print ISBN: 978-3-11-020733-0, 2009.
[23] A. Sarkar and D. Zeman, "Automatic Extraction of Subcategorization Frames for Czech," in *Proc. of the 18th International Conference on Computational Linguistics*, 2000.
[24] A. Séreny, E. Simon, and A. Babarczy, "Automatic Acquisition of Hungarian Subcategorization Frames," in *9th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics CINTI*, 2008.
[25] L. Tesnière, *Éléments de syntaxe structurale* (*Elementos de sintaxis estructural*) 1959.
[26] A. Ushioda, D. Evans, T. Gibson, and A. Waibel, "The automatic acquisition of frequencies of verb subcategorization frames from tagged corpora," in Boguraev, B. and Pustejovsky, J. eds. *SIGLEX ACL Workshop on the Acquisition of Lexical Knowledge from Text*. Columbus, Ohio, pp. 95-106, 1993.
[27] E. Uzun, Y. Kılıçaslan, H.V. Agun, and E. Uçar, "Web-based Acquisition of Subcategorization Frames for Turkish," in *Computational Intelligence: Methods and Applications*, IEEE Computational Intelligence Society, 2008.

Using Continuations to Account for Plural Quantification and Anaphora Binding

Anca Dinu

***Abstract*—We give in this paper an explicit formal account of plural semantics in the framework of continuation semantics introduced in [1] and extended in [4]. We deal with aspects of plural dynamic semantics such as plural quantification, plural anaphora, conjunction and disjunction, distributivity and maximality conditions. Those phenomena need no extra stipulations to be accounted for in this framework, because continuation semantics provides a unified account of scope-taking.**

***Index Terms*—Discourse semantics, continuations, plural anaphora, plural quantifiers' scope.**

I. Introduction

THE formal study of plurals is a major and complex undertaking. It should address plural quantification, plural anaphora, conjunction and disjunction, distributivity, maximality, among other problems. Ideally, plural linguistic phenomena should be parallel to their singular counterpart. Unfortunately, it is not so: plurality introduces a number of complications not present in the analysis of singular. In this paper we will only give some directions for further study of plural semantics (our analysis is far from being exhaustive), with emphasis on plural anaphora, in the framework of continuation semantics.

The programming language concept of continuations was successfully used by Barker and Shan in a series of articles to analyze intra-sentential linguistic phenomena such as focus fronting, donkey anaphora, presuppositions, crossover or superiority [2, 3, 19, 18, 1]. Moreover, [9] proposed an elegant discourse semantics based on continuations. Continuations are a standard tool in computer science, used to control side effects of computation (such as evaluation order, print or passing values). The basic idea of continuizing a grammar is to provide subexpressions with direct access to their own continuations (future context), so subexpressions are modified to take a continuation as an argument. For instance, if we take the local context to be restricted to the sentence, when computing the meaning of the sentence *John saw Mary*, the default future of the value denoted by the subject is that it is destined to have the property of seeing Mary predicated of it. In symbols, the continuation of the subject denotation $\boldsymbol{j}$ is the function $\lambda x.saw\ \boldsymbol{m}\ x$. Similarly, the default future of the object denotation $\boldsymbol{m}$ is the property of being seen by John, the function $\lambda y.\ saw\ y\ \boldsymbol{j}$; the continuation of the transitive verb denotation *saw* is the function $\lambda R.R\ \boldsymbol{m}\ \boldsymbol{j}$; and the continuation of the VP *saw Mary* is the function $\lambda P.P\ \boldsymbol{j}$. This simple example illustrates two important aspects of continuations: every meaningful subexpression has a continuation; and the continuation of an expression is always relative to some larger expression containing it. Thus, when *John* occurs in the sentence *John left yesterday*, its continuation is the property $\lambda x.\mathbf{yesterday}\ \mathbf{left}\ x$; when it occurs in *Mary thought John left*, its continuation is the property $\lambda x.thought\ (left\ x)\ \boldsymbol{m}$ and so on.

In what follows we will give a short survey of the continuations semantic framework in which we will analyze plurals.

One of the main challenges of interpreting a discourse (giving it a compositional semantics) is interpreting cross-sentential anaphora. Assigning a first order logical representation to a discourse like *A man came. He whistled* is problematic. How can we get from the two first order representations in (1) and (2) the representation in (3), i.e. obtaining the bound variable *whistled(x)* in (3) from the free one in (1)?

$$\exists x.(man(x) \wedge came(x)) \quad (1)$$

$$whistled(x) \quad (2)$$

$$\exists x.((man(x) \wedge came(x)) \wedge whistled(x)) \quad (3)$$

Various dynamic semantic theories that handle this were proposed, for instance in Discourse Representation Theory [12], File Change Semantics [10], Dynamic Montague Grammar [7], Dynamic Predicate Logic [8], Jacobson's variable free semantics [11]. All these theories have also something to say about the complex semantics of plurals (for instance about the plural version of the above discourse: *Some man came. They whistled*), complications included. We have chosen to treat plural semantics in the framework of continuation semantics because it provides a unified account of scope-taking (quantification and binding employ the same mechanism), and thus an elegant treatment of anaphora (be it singular or plural).

We will use Barker's tower notation for a given expression, which consists of three levels: the top level specifies the syntactic category of the expression coached in categorical grammar (the categories act either as functions or as attributes), the middle level is the expression itself and the bottom level is the semantic value.

Manuscript received on November 1, 2011, manuscript accepted on December 9, 2011.

The author is with the Centre for Computational Linguistics, University of Bucharest, Romania (e-mail: anca_d_dinu@yahoo.com).

$$\begin{matrix} syntactic\ category \\ expression \\ semantic\ value \end{matrix}$$

The syntactic categories are written $\frac{C|B}{A}$, where A, B and C can be any categories. We read it counter clockwise: the expression functions as category A in local context, takes scope at an expression of category B to form an expression of category C.

The semantic value $\lambda k.f[k(x)]$ is equivalently written vertically as $\frac{f[\,]}{x}$ omitting the future context (continuation) *k*. Here, *x* can be any expression, and *f[ ]* can be any expression with a gap *[ ]*. Free variables in *x* can be bound by binders in *f [ ]*. This notational convention is meant to make easier (more visual) then in linear notation the combination process of two expressions. [1] gives the two possible modes of combination:

$$\begin{pmatrix} \frac{C|D}{A/B} & \frac{D|E}{B} \\ left-exp & right-exp \\ \frac{g[\,]}{f} & \frac{h[\,]}{x} \end{pmatrix} = \begin{matrix} \frac{C|E}{A} \\ left-exp\ right-exp \\ \frac{g[h[\,]]}{f(x)} \end{matrix}$$

$$\begin{pmatrix} \frac{C|D}{B} & \frac{D|E}{B\backslash A} \\ left-exp & right-exp \\ \frac{g[\,]}{x} & \frac{h[\,]}{f} \end{pmatrix} = \begin{matrix} \frac{C|E}{A} \\ left-exp\ right-exp \\ \frac{g[h[\,]]}{f(x)} \end{matrix}$$

Below the horizontal lines, combination proceeds simply as in combinatory categorical grammar: in the syntax, B combines with *A/B* or *B\A* to form *A*; in the semantics, x combines with f to form *f(x)*. Above the lines is where the combination machinery for continuations is employed. The syntax combines the two pairs of categories by cancellation: the *D* on the left cancels with the *D* on the right. The semantics combines the two expressions with gaps by composition: we plug *h[ ]* to the right into the gap of *g[ ]* to the left, to form *g[h[ ]]*. The expression with a gap on the left, *g[ ]*, always surrounds the expression with a gap on the right, *h[ ]*, no matter which side supplies the function or the argument below the lines. This fact expresses the generalization that the default order of semantic evaluation is left-to-right.

When there is no quantification or anaphora involved, a simple sentence like *John came* is derived as follows:

$$\begin{pmatrix} DP & DP\backslash S \\ John & came \\ j & came \end{pmatrix} = \begin{matrix} S \\ John\ came \\ came\ j \end{matrix}$$

In the syntactic layer, as usual in categorical grammar, the category under slash (here DP) cancels with the category of the argument expression; the semantics is function application.

Quantificational expressions have extra layers on top of their syntactic category and on top of their semantic value, making essential use of the powerful mechanism of continuations in ways proper names or definite descriptions do not. For example, below is the derivation for *A man came*:

$$\begin{pmatrix} \frac{S|S}{DP}/N & N & \frac{S|S}{DP\backslash S} \\ a & man & came \\ \lambda P.\frac{\exists x.P(x)\wedge[\,]}{x} & man & \frac{[\,]}{came} \end{pmatrix} = \begin{matrix} \frac{S|S}{S} \\ a\ man\ came \\ \frac{\exists x.man(x)\wedge[\,]}{came\ x} \end{matrix}$$

Comparing the analysis above of *John came* with that of *A man came* reveals that *came* has been given two distinct values. The first, simpler value is the basic lexical entry, the more complex value being derived through the standard type-shifter *Lift*, proposed by [15] and many others:

$$\begin{matrix} A \\ expression \\ x \end{matrix} \overset{Lift}{\Longrightarrow} \begin{matrix} \frac{B|B}{A} \\ expression \\ \frac{[\,]}{x} \end{matrix}$$

Syntactically, *Lift* adds a layer with arbitrary (but matching) syntactic categories. Semantically, it adds a layer with empty brackets. In linear notation we have: $x \overset{Lift}{\Longrightarrow} \lambda k.k(x)$.

To derive the syntactic category and a semantic value with no horizontal line, [1] introduce the type-shifter Lower. In general, for any category *A*, any value *x*, and any semantic expression *f [ ]* with a gap, the following type-shifter is available.

$$\begin{matrix} \frac{A|S}{S} \\ expression \\ \frac{f[\,]}{x} \end{matrix} \overset{Lower}{\Longrightarrow} \begin{matrix} A \\ expression \\ f[x] \end{matrix}$$

Syntactically, *Lower* cancels an *S* above the line to the right with an *S* below the line. Semantically, *Lower* collapses a two-level meaning into a single level by plugging the value *x* below the line into the gap *[ ]* in the expression *f [ ]* above the line. *Lower* is equivalent to identity function application.

The third and the last type shifter we need is the one that treats binding. Binding is a term used both in logics and in linguistics with analog (but not identical) meaning. In logics, a variable is said to be bound by an operator (as the universal or existential operators) if the variable is inside the scope of the operator. If a variable is not in the scope of any operator then the variable is said to be free. In linguistics, a binder may be a constituent such as a proper name (*John*), an indefinite common noun (*a book)*, an event or a situation. Anaphoric expressions such as pronouns (*he, she, it, him, himself*, etc), definite common nouns (*the book*, *the book that John read*), demonstrative pronouns (like *this, that*), etc act as variables that take the value of (are bind by) a previous binder. We adopt the idea (in line with [1]) that the mechanism of binding is the same as the mechanism of scope taking.

In order to give a proper account of anaphoric relations in discourse, we need to formulate an explicit semantics for both the binder and the anaphoric expressions to be bound. Any DP may act as a binder, as the *Bind* rule from [1] explicitly states:

$$\begin{matrix} \frac{A|B}{DP} \\ expression \\ \frac{f[\,]}{x} \end{matrix} \overset{Bind}{\Longrightarrow} \begin{matrix} \frac{A|DP \rhd B}{DP} \\ expression \\ \frac{f([\,]x)}{x} \end{matrix}$$

At the syntactic level, *Bind* says that an expression that functions in local context as a DP may look to the right to bind an anaphoric expression (encoded by the sign ▷). At the semantic level, the expression transmits the value of the variable *x*. In linear notation, the semantic part of *Bind* looks like: $\lambda k.\, f[k(x)] \overset{Bind}{\Longrightarrow} \lambda k.\, f([k(x)]x)$

As for the elements that may be bound, [1] give for instance the following lexical entry for singular pronoun *he*:

$$\frac{\frac{DP \triangleright S|S}{DP}}{\frac{he}{\frac{\lambda y.[\,]}{y}}}$$

Distinct scope-taking levels correspond to different binders, layers playing the role of indices: a binder and the pronoun it binds must take effect at the same layer in the compositional tower. A superior level takes scope at inferior levels and left expressions take scope at right expressions, to account for left-to-right natural language order of processing.

In order to account for discourse phenomena, [4] gives the semantics of the dot as a function that take two sentence denotations and returns a sentence denotation (their conjunction):

$$\frac{S\backslash(S/S)}{\lambda p\lambda q.\, p \wedge q}$$

II. Plural Quantificational Determiners

There is a vast literature on representing plurals. We will only refer to two of the most influential existing approaches: the proposals of Scha [17] and of Link [13]. The most well-known and largely accepted view of natural language quantification is the generalized quantifiers view (Montague [14] and many others). The generalized quantifier type <<<e,t>,t>,t> is exactly the type of quantificational determiners in continuation-based semantics. This is by no means a coincidence, generalized quantifiers approach only continuizes the noun phrase meanings rather than continuizing uniformly throughout the grammar as it is done in continuation-based semantics.

A tradition going back at least to Evans [5] says that the scope of all quantifiers is clause bounded. An E-type (or donkey) pronoun is a pronoun that lies outside the restrictor of a quantifier or outside the antecedent of a conditional, yet co-varies with some quantificational element inside it, usually an indefinite. Here there are some of the famous donkey sentences examples:

If a farmer owns a donkey, he beats it.
Every farmer who owns a donkey beats it.

Evans made it standard to assume that the indefinite *a donkey* cannot take scope over the pronoun *it*, and therefore cannot bind it, at least not in the ordinary sense of binding. To the contrary, as [1] put forward, the relationship between *a donkey* and *it* in the above examples seems like binding because it *is* just binding: the scope of the indefinite a donkey stretches over the consequent of the conditional and binds the pronoun *it*. In what follows, manipulating the scope of plural quantificational determiners in a similar liberal manner (in which we do not restrict it to clause boundaries) will allow us to account for a wide range of linguistic data involving plural anaphora binding.

Plural quantificational determiners take as arguments plural common nouns. Among other constructions such as coordinated DPs or singular DPs, they introduce plural referring variables. From a technical point of view, a plural referring variable notated with upper letters (*X, Y, Z,..*) is a set of entities.

We will give lexical entries for the plural quantificational determiners *some* (not to be confused with its singular counterpart), *all* and *most,* in the continuation semantics framework. *Some* and *most* need special care when the plural variable they introduce binds some subsequent anaphora, due to the so-called maximality constraint. While *Some kids came*. (with no other subsequent anaphora that refers to the kids that came) means that there is a set of any cardinal of kids that came, the discourse *Some kids came. They played*. means that there is a *maximal* set of kids who came and that maximal set played. So, there is a maximality operator that blocks further transmission of arbitrary sets, much like the negation blocks transmission of values of indefinites in direct object position to subsequent anaphora. The two uses of *some* have different truth-conditions. When *some* is used in the first, weak sense, we take it to have the following lexical entry:

$$\frac{\frac{S|S}{DP^{pl}/N^{pl}}}{\frac{some}{\lambda P.\frac{\exists X.(|X| \geq 1 \wedge P(X) \wedge [\,]}{X}}}$$

Then, we have to force the scope closing of the variable *X* in the usual way by applying *Lower*, in order to forbid it to bind subsequent anaphora (transmit a non-maximal value).

When used in the second, maximal sense, that exports a maximal set to bind a subsequent anaphora (such as t*hey*), we take *some* to have the alternative lexical entry:

$$\frac{\frac{\frac{S|DP^{pl} \triangleright S}{\frac{S|S}{DP^{pl}/N^{pl}}}}{\frac{some}{\exists X.|X| \geq 1 \wedge [\,]X}}}{\lambda P.\frac{X = \underset{Y}{argmax}|\{Y; P(Y) \wedge [\,]\}|}{Y}}$$

Note that we could have not used in this case the regular Bind rule, because of the intervening level that contains *argmax*. This level blocs the transmission of variable *Y* and only lets the maximal variable *X* to bind subsequent anaphora.

For the same reasons, we similarly treat the quantificational determiner *most*, for which we propose the following two alternative lexical entries, one for the weak sense, and one for strong (maximal) sense, respectively:

$$\lambda P.\frac{\frac{\frac{S|S}{DP^{pl}}}{N^{pl}}}{X}\ \frac{most}{\exists X.\, P(X) \wedge [\,] \wedge 2|X| \geq |\{x; Px\}|}$$

$$\lambda P.\frac{\frac{\frac{S|DP^{pl}}{\frac{S|S}{DP^{pl}}/N^{pl}}}{X}\ \frac{most}{\exists X.2|X| \geq |\{x; Px\}| \wedge [\,]X}}{Y}\ \frac{}{X = \underset{Y}{argmax}|\{Y; P(Y) \wedge [\,]\}|}$$

For the quantificational determiner *all*, the maximality condition has limited scope only over the restrictor *P*, thus we can give it a single lexical entry:

$$\lambda P.\frac{\frac{\frac{S|S}{DP^{pl}}/N^{pl}}{X}\ \frac{all}{\exists X.\,(X = \underset{Y}{argmax}|\{Y; P(Y)\}|) \wedge [\,]|)}$$

It has been argued that *all* is not a quantificational determiner proper, but more like a modifier. It may be for that reason that it behaves differently compared to genuine quantificational determiners.

We turn now to the problem of compositionally obtaining the meaning of bare plurals. Bare plurals are plurals without overt article. Sentences with bare plurals can have either existential readings (*John gave Mary flowers*), or universal (generic) readings (*Flowers are beautiful*). We propose that the existential reading is accounted for by a silent quantificational determiner that has the same semantics as *some* (i.e. both weak and maximal senses). The universal reading is accounted for by a similar silent quantificational determiner, having the semantics of *all*:

$$\lambda P.\frac{\frac{\frac{S|S}{DP^{pl}}/N^{pl}}{X}\ \frac{\Phi}{\exists X.\,(X = \underset{Y}{argmax}|\{Y; P(Y)\}|) \wedge [\,]|)}$$

We take predicates to be distributive or collective: in *John gave Mary flowers*, *gave* is used in its collective sense for its second argument; in *Flowers are beautiful*, *is beautiful* is used in its distributive sense in its first argument.

Cardinal determiners have two built-in meaning components: an existential component and a cardinality one. We propose the following two alternative lexical entries for card, one for the referential (weak) meaning *there are card Ps…*, the other for the strong meaning *there are exactly card Ps…*:

$$\lambda P.\frac{\frac{\frac{S|S}{DP^{pl}}}{N^{pl}}}{X}\ \frac{card}{\exists X.\, |X| = card \wedge P(X) \wedge [\,}$$

$$\lambda P.\frac{\frac{\frac{S|S}{DP^{pl}}/N^{pl}}{X}\ \frac{card}{argmax|\{X; P(X) \wedge [\,]\}| = card}}{X}$$

The semantic ambiguity between the two lexical entries of a cardinal *card* is determined by whether the scope of the following context (continuation) lies inside the scope of the cardinality (as in the second entry) or not (as in the first entry).

Both these weak and strong meaning of cardinal may bind subsequent anaphora (as opposed to the case of *some* that can bind only with its maximal meaning). For the weak meaning, we can just use the regular *Bind* rule, whereas for the strong meaning (exactly *card*), one cannot use Bind because that would bring the continuation into the scope of *argmax*, altering the truth conditions. Thus, we have to force the scope closing of *argmax* immediately after the interpretation of the cardinal's minimal clause by applying *Lower*. To allow the strong meaning of *card* to bind, we have to give it jet another lexical entry:

$$\lambda P.\frac{\frac{\frac{S|DP^{pl}}{\frac{S|S}{DP^{pl}}/N^{pl}}}{X}\ \frac{card}{\exists X.\,[\,]}}{Y}\ \frac{}{X = \underset{Y}{argmax}|\{Y; P(Y) \wedge [\,]X\}| \wedge |X| = card}$$

These representations are not completely satisfying because the lexical ambiguity of plural quantificational determiners generates an explosion of ambiguous representation of the discourse in which the determiners are used. We leave the problem of finding a more general solution for a unitar representation of the plural quantificational determiners *some*, *most* and cardinals to further research.

III. Distributive vs. Collective Reading

We will consider two of the most influential existing strategies to deal with plurals and their associated ambiguities (collective, distributive or cumulative readings): Scha [17] and Link [13]. Scha and Link locate the source for the ambiguity of plural sentences differently. According to Scha the ambiguity between collective, distributive and possibly other readings is located in the plural noun phrase or more precisely in the determiner. According to Link, noun phrases are unambiguous and the readings should be generated within the verb phrase. A third strategy proposes that readings of complex sentences are a result of the whole structure or as Roberts [16] puts it: "Distributivity is a property of predications, combinations of a subject and a predicate."The readings can be triggered by different elements of a sentence; there is a functional interplay between the different categories."

We will take *predicates*, not nouns to be distributive, collective, or ambiguous. We will not commit ourselves to whether the distributivity comes as a feature from the lexical semantics, or it is entailed from the world knowledge and the sense of the predicate itself [16]. Here are some examples:

Sue and Mary are pregnant. (*be pregnant* is a distributive predicate)

John and Bill moved the piano. (*moved* is an ambiguous between distributive and collective predicate)

The students gathered in the square. (*gathered* is a collective predicate)

As a general rule, we posit that a distributive predicate P_{dist} is true of a plural referring variable $X=\{x_1, x_2, \ldots x_n\}$ iff $P_{dist}(x_1) \wedge P_{dist}(x_2) \wedge \ldots \wedge P_{dist}(x_n)$. And a collective predicate P_{call} is true of a plural referring variable $X=\{x_1, x_2, \ldots x_n\}$ iff $P_{call}(x_1 \wedge x_2 \ldots \wedge x_n)$. Note that a predicate may have multiple arguments (subject and direct object, for instance). So a predicate may be distributive or collective in each of the arguments.

IV. Coordonation: Conjunciton and Disjunction

The work [2] gives the following lexical entry for *or*:

$$\lambda R \lambda L. \frac{(\lambda k.\, k(L) \vee k(R))(\lambda x.\, [\,])}{\begin{array}{c}(A\backslash\frac{S|S}{A})/A \\ or \\ x\end{array}}$$

The lexical entry for *or* is polymorphic: *A* can be any category, such as *DP, DP\S* (verb phrases), *DP\DP* (adjectives) or *S* (sentence). Partee and Rooth [15] are the first to suggest allowing phrases like *John* or *Bill* to introduce new variables.

We point that disjunction may introduce only singular variables:

John owns a donkey or a goat. He beats it/ them.*
*John or Bill called. He/*They hang up.*

We straightforwardly extend the semantic representation for disjunction from [2] to conjunction:

$$\lambda R \lambda L. \frac{(\lambda k.\, k(L) \wedge k(R))(\lambda x.\, [\,])}{\begin{array}{c}(A\backslash\frac{S|S}{A})/A \\ and \\ x\end{array}}$$

Note that *and* is also polymorphic. Thus it may account for discourses like: *John drinks and talks. He does this for hours,* where *this* is anaphoric to plural events, provided only we modify the binding rule to allow categories other then *DP* (like *DP\S*) to bind subsequent pronouns. Note also that conjoined *DP*s have the power to introduce variables that may be further referred by plural pronouns, a power disjoint *DP*s do not have:

*John owns a donkey and a goat. He beats *it/ them.*
*John and Bill called. *He/They hang up.*

V. Plural Pronominal Anaphora

Ideally, singular and plural pronominal anaphora should behave similarly and be parallel phenomena. Unfortunately, at a closer look, there are striking differences between the anaphoric properties of singular and plural pronouns. On the one hand, only singular *DP*s have the power to introduce singular variables that could bind subsequent singular pronominal anaphora. On the other, plural variables may be introduced not only by plural *DP*s, but also by: two or more singular *DP*s, coordinated (*John and Mary came. They whistled*) or not (*John took Mary to Acapulco. They had a lousy time*), or by quantificational singular *DP*s (*Every man came. They whistle* or *A kid climbed every tree. They were full of energy*).

We first treat the simplest case, that of plural entities introduction by plural *DP*s (analogous to singular entity introduction). Plural *DP*s are formed of plural quantificational determiners such as *some, all* or *most* and a plural common noun required as argument by the determiner. We take singular common nouns to be functions (properties) of individual variables *x*, while plural common nouns expect a plural individual variable *X*. Thus, for such (non-specific) antecedents of *they*, we may use the following lexical entry:

$$\frac{\begin{array}{c}DP^{pl} \rhd S|S \\ \hline DP^{pl} \\ they \\ \lambda Y.[\,] \\ \hline Y\end{array}}{}$$

Here is the derivation for *Some kids came. They played.*:

$$\frac{\begin{array}{c}\frac{S|DP^{pl} \rhd S}{\frac{S|S}{DP^{pl}/N^{pl}}} \\ some \\ \exists X.\,|X| \geq 1 \wedge [\,]X \\ \hline X = \underset{Y}{argmax}|\{Y: P(Y) \wedge [\,]\}| \\ \hline Y\end{array}}{\lambda P.} \quad \frac{\begin{array}{c}S|S \\ \hline N^{pl} \\ kids \\ [\,] \\ \hline kids\end{array}}{} \quad \frac{\begin{array}{c}DP^{pl} \rhd S|DP^{pl} \rhd S \\ \hline \frac{S|S}{DP^{pl}\backslash S} \\ came \\ [\,] \\ \hline [\,] \\ \hline came\end{array}}{}$$

$$= \frac{\begin{array}{c}\frac{S|DP^{pl} \rhd S}{\frac{S|S}{S}} \\ some\ kids\ came \\ \exists X.\,|X| \geq 1 \wedge [\,]X \\ \hline X = \underset{Y}{argmax}|\{Y: kids(Y) \wedge [\,]\}| \\ \hline came\ Y\end{array}}{} \Longrightarrow^{Lower}$$

$$\frac{\begin{array}{c}\frac{S|DP^{pl} \rhd S}{S} \\ some\ kids\ came \\ \exists X.\,|X| \geq 1 \wedge [\,]X \\ \hline X = \underset{Y}{argmax}|\{Y: kids(Y) \wedge came\ Y\}|\end{array}}{}$$

$$\frac{\begin{array}{c}\frac{S|DP^{pl} \rhd S}{S} \\ some\ kids\ came \\ \exists X.\,|X| \geq 1 \wedge [\,]X \\ \hline X = \underset{Y}{argmax}|\{Y: kids(Y) \wedge came\ Y\}|\end{array}}{} \quad \frac{\begin{array}{c}DP^{pl} \rhd S|DP^{pl} \rhd S \\ \hline S\backslash(S/S) \\ [\,] \\ \hline \lambda p \lambda q.p \wedge q\end{array}}{} \quad \frac{\begin{array}{c}DP^{pl} \rhd S|S \\ \hline S \\ they\ played \\ \lambda Z.[\,] \\ \hline played\ Z\end{array}}{}$$

$$= \frac{\begin{array}{c}\frac{S|S}{S} \\ some\ kids\ came.they\ played \\ \exists X.\,|X| \geq 1 \wedge \lambda Z.[\,]X \\ \hline X = \underset{Y}{argmax}|\{Y: kids(Y) \wedge came\ Y\}| \wedge played\ Z\end{array}}{} \Longrightarrow^{Lower}$$

$$\frac{S}{\text{some kids came. they played}}$$
$$\exists X.\,|X| \geq 1 \wedge \lambda Z.\left[X = \underset{Y}{\text{argmax}}\,|\{Y : \textbf{kids}(Y) \wedge \textbf{came}\,Y\}| \wedge \textbf{played}\,Z\right]X$$

$$= \frac{S}{\text{some kids came. they played}}$$
$$\exists X.\,|X| \geq 1 \wedge X = \underset{Y}{\text{argmax}}\,|\{Y : \textbf{kids}(Y) \wedge \textbf{came}\,Y\}| \wedge \textbf{played}\,X$$

which amounts to saying that there is a plural entity *X* of cardinality at least one, formed of all the kids that came and that plural entity *X* played. In a similar manner one obtains the derivation for *Most kids came. They played* and for *All kids came. They played.*

As for the plural anaphora introduced by cardinal determiners, consider the following two examples: *Five men walk in the park. They watch the birds.* (preferred reading: there are some context relevant five men and they walk in the park and they watch the birds; there could be other not contextually important men walking and watching); *Five men walk in the park and watch the birds.* (preferred reading: there are exactly five man who are in the park and watch the birds). We take both examples to be semantically ambiguous between two readings which correspond to the two scope-distinct lexical entries for the cardinal determiner five. Pragmatic reasons dictate the preferred reading in each case. We give the interpretations of these preferred readings (and skip the not preferred ones, though semantically possible), ignoring the full interpretation of *walk in the park* and of *watch the birds*:

$$\frac{S}{\text{five men walk in the park. they watch the birds}}$$
$$\exists X.\,|X| = 5 \wedge \textbf{men}(X) \wedge \textbf{walk in the park}\,X \wedge \textbf{watch the birds}\,X$$

$$\frac{S}{\text{five men walk in the park and watch the birds}}$$
$$\underset{X}{\text{argmax}}\,|\{X : \textbf{men}(X) \wedge \textbf{walk in the park}\,X \wedge \textbf{watch the birds}\,X\}| = 5$$

We turn now to the case of introducing plural entities by coordination (conjunction or disjunction). The lexical entry for conjunction obviously gives right truth conditions and offers an antecedent for subsequent anaphora, as in, for example: *John and Mary came. They whistled.* In such cases, where more than one specific antecedent is present in the discourse, the lexical entry for *they* needs to search left for two (or three, or another number) *DP*s, for instance:

$$\frac{\dfrac{DP \rhd S|S}{DP \rhd S|S}}{\dfrac{DP^{pl}}{\dfrac{\textit{they}}{\dfrac{\lambda y.[\,]}{\dfrac{\lambda x.[\,]}{\{x, y\}}}}}}$$

$$\frac{\dfrac{DP \rhd S|S}{DP \rhd S|S}}{\dfrac{DP^{pl}}{\dfrac{\textit{they}}{\dfrac{\lambda z.[\,]}{\dfrac{\lambda y.[\,]}{\{y, z\}}}}}} \quad \frac{\dfrac{S|S}{S|S}}{\dfrac{DP^{pl}\backslash S}{\dfrac{\textit{whistled}}{\dfrac{[\,]}{[\,]}}}} = \quad \frac{\dfrac{DP \rhd S|S}{DP \rhd S|S}}{\dfrac{S}{\dfrac{\textit{they whistled}}{\dfrac{\lambda z.[\,]}{\dfrac{\lambda y.[\,]}{\textbf{whistled}\,\{y, z\}}}}}} = \quad \frac{\dfrac{DP \rhd S|S}{DP \rhd S|S}}{\dfrac{S}{\dfrac{\textit{they whistled}}{\dfrac{\lambda z.[\,]}{\dfrac{\lambda y.[\,]}{\textbf{whistled}\,(y) \wedge \textbf{whistled}\,(z)}}}}}$$

$$\frac{\dfrac{S|DP \rhd S}{S|DP \rhd S}}{\dfrac{S}{\dfrac{\textit{John and Mary came}}{\dfrac{[\,]j}{\dfrac{[\,]m}{\textbf{came}\,(j) \wedge \textbf{came}\,(m)}}}}} \quad \frac{\dfrac{DP \rhd S|DP \rhd S}{DP \rhd S|DP \rhd S}}{\dfrac{S\backslash(S/S)}{\dfrac{[\,]}{\dfrac{[\,]}{\lambda p \lambda q.p \wedge q}}}} \quad \frac{\dfrac{DP \rhd S|S}{DP \rhd S|S}}{\dfrac{S}{\dfrac{\textit{They whistled}}{\dfrac{\lambda z.[\,]}{\dfrac{\lambda y.[\,]}{\textbf{whistled}\,(y) \wedge \textbf{whistled}\,(z)}}}}}$$

$$= \frac{S}{\text{John and Mary came. They whistled}}$$
$$\textbf{came}\,(j) \wedge \textbf{came}\,(m) \wedge \textbf{whistled}\,(m) \wedge \textbf{whistled}\,(j)$$

The mechanism of transmitting more than one value of the antecedent to the plural anaphoric pronoun *they* is the same for referring to determiner phrases that are not in a coordination relation (by conjunction) like: *John met Mary. They smiled.*

An open problem still remains: how to block *John or Bill called. *They hang up*?

The third case, the case of introducing plural entities by singular *DP*s is the most difficult. We will stipulate that a singular *DP* may bind a plural entity (introduced by a pronoun or a definite) if and only if it is logically a plural, that is: either the singular *DP* is bound by universal quantifier (as in *Every man came. They whistled*), or the singular *DP* is embedded inside an expression in which it co-varies with a variable bound by the universal quantifier (the so-called structural dependency), as in:

A kid climbed every tree. He was full of energy. or *They were full of energy.*

The first sentence has two distinct readings, one in which *a* takes scope over *every* and one in which *every* takes scope over *a*. If the first sentence is continued by the second, then the only possible reading in natural language becomes that with *a* taking scope over *every*:

$$\frac{S}{\text{a kid climbed every tree. he be full of energy}}$$
$$\exists x.\,\textbf{kid}\,(x) \wedge \neg\exists y.\,\textbf{tree}(y) \wedge \neg[\,\textbf{climbed}\,y\,x] \wedge \textbf{be full of energy}\,x$$

If the first sentence is continued by the third, the only possible reading in natural language becomes that with *every* taking scope over *a*, both with its general scope and its nuclear scope:

$$\frac{S}{\text{a kid climbed every tree. they were full of energy}}$$
$$\neg\exists y.\,\textbf{tree}(y) \wedge \neg\exists x.\,\textbf{kid}(x) \wedge \textbf{climbed}\,y\,x \wedge \textbf{were full of energy}\,x$$

VI. Conclusions

We gave in this paper an explicit formal account of plural semantics based on the continuation semantics in [2] and [4]. We accounted for some aspects of plural semantics such as plural quantification, plural anaphora, conjunction and disjunction, distributivity and maximality conditions. Those phenomena needed no extra stipulations to be accounted for in this framework, because continuation based semantics provides a unified account of scope-taking.

Acknowledgments

Research supported by the CNCS, IDEI – PCE project 311/2011, "The Structure and Interpretation of the Romanian Nominal Phrase in Discourse Representation Theory: the Determiners."

REFERENCES

[1] C. Barker and Chung-chieh Shan, "Donkey anaphora is in-scope binding," *Semantics & Pragmatics*, Volume 1, pp. 1–46, 2008.

[2] C. Barker, "Continuations and the nature of quantification," *Natural Language Semantics,* 10(3). 211–242, 2002.

[3] C. Barker, "Continuations in natural language," in *Proceedings of the fourth ACM SIGPLAN workshop on continuations*, Hayo Thielecke (ed.), pp. 55–64, 2004.

[4] A. Dinu, "Versatility of continuations in discourse semantics," *Fundamenta Informaticae*, 18 pp., 2011.

[5] G. Evans, "Pronouns, Quantifiers and Relative Clauses I & II," *Canadian Journal of Philosophy,* 7, 1977.

[6] M. Felleisen, "The theory and practice of first-class prompts," in *Proceedings of the Fifteenth Annual ACM Symposium on Principles of Programming Languages*, J. Ferrante and P. Mager (eds), p. 180-190, San Diego, California, ACM Press, 1988.

[7] J. A.G. Groenendijk and M.B.J. Stokhof, "Dynamic Montague Grammar," in *Papers from the Second Symposium on Logic and Language*, L. Kalman and L. Polos (eds.), Budapest: Akadémiai Kiadó, 1990.

[8] J. Groenendijk and M. Stokhof, "Dynamic predicate logic," *Linguistics and Philosophy,* 14(1) 39–100, 1991.

[9] P. de Groote, "Towards a Montagovian account of dynamics," *Semantics and Linguistic Theory XVI*, 2006.

[10] I. Heim, "File change semantics and the familiarity theory of definiteness," in *Meaning, Use and the Interpretation of Language*, Rainer Bauerle, Cristoph Schwarze, and Arnim von Stechow (eds), Walter de Gruyter & Co., 1983.

[11] P. Jacobson, "Towards a variable-free semantics," *Linguistics and Philosophy,* 22(2). 117–185, 1999.

[12] H. Kamp and U. Reyle, *From Discourse to Logic*, Kluwer Academic Publishers, 1993.

[13] Link, G.: The logical analysis of plurals and mass terms: A lattice-theoretical approach. In *Meaning, use and the interpretation of language*, eds. R. Bäuerle, C. Schwarze and A. von Stechow, 303-323. Berlin, New York: Walter de Gruyter, 1983.

[14] R. Montague, "The Proper Treatment of Quantification in English," in *Formal Philosophy: Selected Papers of Richard Montague*, R. Thomason (ed.), pp. 247–270. New Haven:Yale, 1970.

[15] B. H. Partee and M. Rooth, "Generalized conjunction and type ambiguity," in *Meaning, use and interpretation of language*, Rainer Bäuerle, C. Schwarze, and A. von Stechow (eds.), pp. 361–383, de Gruyter, 1983.

[16] C. Roberts, *Modal subordination, anaphora and distributivity*. PhD dissertation, UMass. Amherst, 1987.

[17] R. Scha, "Distributive, collective and cumulative quantification," in *Formal Methods in the Study of Language,* J. Groenendijk, M. Stokhof, and T.M.V. Janssen (eds.), Mathematisch Centrum, Amsterdam, 1981.

[18] Chung-chieh Shan and C. Barker, "Explaining crossover and superiority as left-to-right evaluation," *Linguistics and Philosophy,* 29.1:91–134, 2006.

[19] Chung-chieh Shan, *Linguistic side effects,* Ph.D. thesis, Harvard University, 2005.

Demodulation of Interferograms based on Particle Swarm Optimization

Julio Jiménez, Humberto Sossa, Francisco Cuevas, and Laura Gómez

***Abstract*—A parametric method to carry out fringe pattern demodulation by means of a particle swarm optimization is presented. The phase is approximated by the parametric estimation of an *n*th-grade polynomial so that no further unwrapping is required. A particle swarm is used to optimize the input parameters of the function that estimates the phase. A fitness function is established to evaluate the particles, which considers: (a) the closeness between the observed fringes and the recovered fringes, (b) the phase smoothness and c) the prior knowledge of the object, such as its shape and size. The swarm of particles evolves until a fitness average threshold is obtained. We demonstrate that the method is able to successfully demodulate fringe patterns and even a one-image closed-fringe pattern.**

***Index Terms*—Phase retrieval; fringe analysis; optical metrology; evolutionary technique.**

I. Introduction

INTERFEROMETRY is an non-destructive optical technique: It is used to measure physical variables (stress, temperature, acceleration, curvature, and so on), and this with a high degree of resolution, as it follows from the wavelength magnitude used by the light [1]. A typical interferometer splits a laser beam using a beam divisor. Beam A is called reference, and it is projected directly onto a film or a CCD camera using mirrors or optical fiber; beam B interacts with the physical phenomenon to be measured. The interaction modifies the optic path of beam B; which is then projected onto the same film or CCD camera as beam A. A diagram of the Michelson interferometer is shown in Fig. 1, and the basic operation of the interferometer is as follows. Light from a light source is split into two parts, with one part of the light travelling a different path length than the other. After traversing these different path lengths, the two parts of the light beam are brought together to interfere with each other and the interference pattern can be seen on a screen.

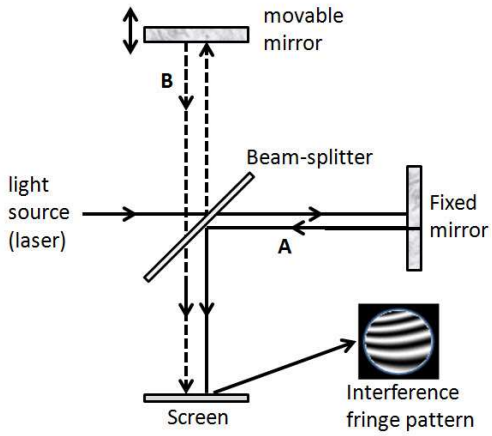
In optical metrology, it is well known that in a fringe pattern can be represented through total irradiance, using the following mathematical expression:

$$I(x,y) = a(x,y) + b(x,y)\cos\left[\phi(x,y)\right], \qquad (1)$$

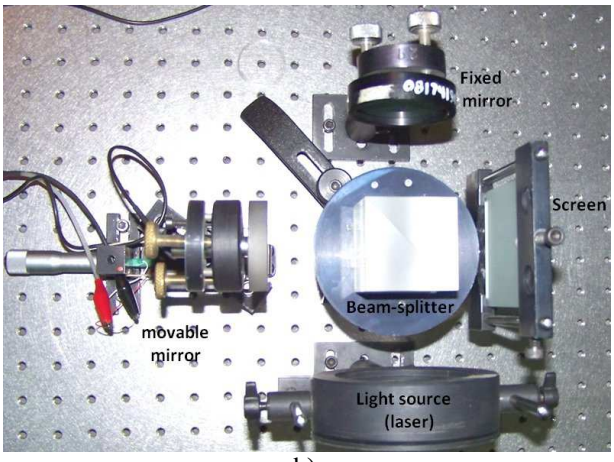
where x, y are integer values representing coordinates of the pixel location in the fringe image, $a(x,y)$ is the background illumination, $b(x,y)$ is the amplitude modulation (e.g., this factor is related with the surface reflectance), and $\phi(x,y)$ is the phase term related to the physical quantity being measured, and is the most important term for optical metrology.

The purpose of any interferometric technique is to determine the phase term, which is related to the physical quantity being measured. Fig. 2(a) shows an interferogram, with its and its associated phase term $\phi(x,y)$ in Fig. 2(b).

One way to calculate the phase term $\phi(x,y)$ is by means of the phase-shifting technique (PST), as described in [2, 3, 4, 5].



a)



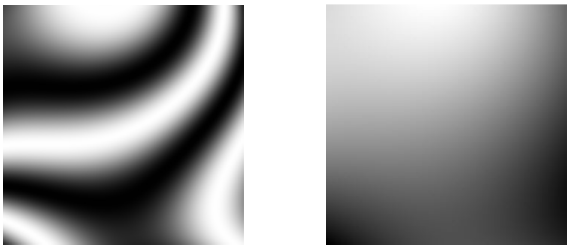
b)

Fig. 1. a) Schematic illustration of a Michelson interferometer, and b) real schematic.

Manuscript received on November 11, 2011, manuscript accepted on December 15, 2011.

Julio Jimenez, Humerto Sossa and Laura Gomez are with Centro de Investigación en Computación, Instituto Politécnico Nacional, México DF., 07738, Mexico (email: jfvielma@cio.mx,hsossa@cic.ipn.mx, lenis45@hotmail).

Francisco Cuevas is with Centro de Investigaciones en Óptica, León, Guanajuato, 37150, Mexico (email: fjcuevas@cio.mx).



(a) (b)

Fig. 2. (a) Fringe pattern, and (b) its phase map.

A drawback is that it requires at least three interferograms with the phase-shifted. The phase shift among interferograms must be known and experimentally controlled. This technique can be used when mechanical conditions are met throughout the interferometric experiment, one of these conditions and the most important refer the object of study, this must be static, to project the phased shifted into it, these because they must have at least three images.

On the other hand, when the mentioned stability conditions are not satisfied, for example objects or experiments that change over time (also called transients), and for that measurement which is only possible to obtain a single image; other techniques can be used to estimate the phase term (or also known as demodulation) from a single fringe pattern; for example in [6] and [17], authors use the Fourier Transform method, while in [8], the Synchronous method is introduced, and the phase locked loop method (PLL) in [9]. However, these techniques work well only if the analyzed interferogram has a carrier frequency and a narrow bandwidth, and the signal has low noise; moreover, these methods do not perform well for phase calculation in a closed-fringe pattern. Additionally, the Fourier and synchronous methods estimate the wrapped phase due to use of an arctangent function during the phase calculation, so an additional unwrapping procedure is required [10]. The unwrapping process is difficult when the fringe pattern includes high amplitude noise, which causes differences greater than 2π radians between adjacent pixels ([11], [12] and [13]). In the PLL technique, the phase is estimated by following the phase changes of the input signal by varying the phase of a computer simulated oscillator (VCO), such that the phase error between the fringe pattern and VCO's signal vanishes.

Recent techniques make use of soft computing algorithms like neural networks and genetic algorithms (GA). In the neural network technique [14] and [15], a multi-layer neural network (MLNN) is trained by using fringe patterns, and the phase gradients associated with them, from calibrated objects. After the training, the MLNN can estimate the phase gradient when the fringe pattern is presented in the MLNN input.

The first method based on GAs to applied to the phase demodulation problem, was proposed by Cuevas et al. in [16], proposed the use of a fitness function based on a phase estimate by creating a surface through by the adjustment of the coefficients of a polynomial of order four, and in [17] was using the fitness function created by Cuevas et al. [16], the

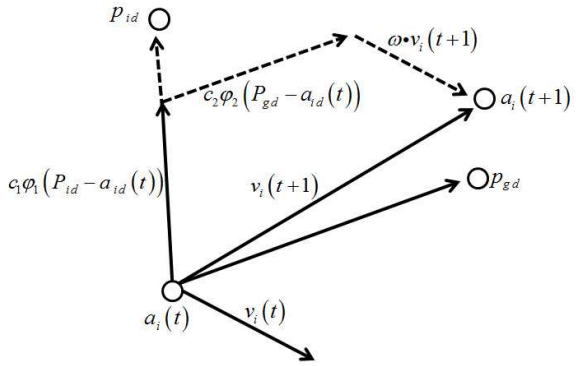


Fig. 3. Update particle.

adjustment of the surface was changed by a Zernike polynomial. To demodulate more complicated interferograms were created methods based on partition of the image, this method is called Window Fringe Pattern Demodulating (WFPD) technique, and it was proposed in [18], and used in [19], where each window is demodulated by a genetic algorithm, and these windows are slightly overlapping. The functions can be Bessel in the case of fringes coming from a vibrating plate experiment, or Zernike polynomials, in an optical testing experiment. In the case when not much information is known about the experiment, a set of low degree polynomials $p(a,x,y)$ can be used. A population of chromosomes is codified with the function parameters that estimate the phase. A fitness function is established to evaluate the chromosomes, and it considers the same aspects as the cost function in a regularization technique. The population of chromosomes evolves until a fitness average threshold is obtained. The method can demodulate noisy, closed fringe patterns and so, no further unwrapping is needed.

In this paper, we present a variation of the WFPD method introduced by Cuevas et al. in [19]. The new proposal is applied to demodulate complex fringe patterns using a particle swarm optimization technique (PSO) to fit a polynomial; it also allows one to create an automatic fringe counting based on digital image processing. In addition, we use low resolution versions of the interferogram for the recovery of the phase; in other words, we use subsampled images. Results using closed and under-sampled computer generated fringe patterns are presented.

II. Particle Swarm Optimization

Particle swarm optimization has been used to solve many optimization problems since it was proposed by Kennedy and Eberhart in [20] and [21]. After that, they published the book in [22] and several papers on this topic ([23], [24] and [25]), one of which made a study on its performance using four non-linear functions, which has been adopted as a benchmark by many researchers in this area. In PSO, each particle moves in the search space with a velocity that is in accordance with its own previous best solution and its group's previous best solution. The dimension of the search space can be any

positive integer. Following Eberhart and Kennedy's naming conventions, D is the dimension of the search space. The i^{th} particle is represented as $A_i = (a_{i1}, a_{i2}, \ldots, a_{iD})$, and the best particle of the swarm, i.e. the particle with the lowest function value, is denoted by index *g*. The best previous position (i.e. the position corresponding to the best function value) of the i^{th} particle is recorded and represented as $P_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$, and the position change (velocity) of the i^{th} particle is $v_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$. Each particle updates its position with the following two equations:

$$v_{id}(t+1) = \omega a_{id} + c_1\varphi_1\left(p_{id} - a_{id}(t)\right) + c_2\varphi_2\left(p_{gd} - a_{id}(t)\right) \quad (2)$$

$$a_{id}(t+1) = a_{id}(t) + v_{id}(t+1), \quad (3)$$

where for each particle i, a_i is the position, v_{id} the velocity, P_{id} the best position of a particle, P_{gd} the best position within the swarm, and c_1 and c_2 are positive constants containing the balance factors between the effect of self-knowledge and social knowledge in moving the particle towards the target; in literature, a value of 2 is usually suggested for the sum of both factors, φ_1 and φ_2 are random numbers between 0 and 1, and ω is inertia weight. Within the update of the particles, the velocity is denoted as the *momentum* with which the force is pulling the particle to continue in its current direction. The best position of a particle is the *cognitive component,* and this force emerges from the particle's tendency to return to its own best solution found so far, while the best position of a swarm is the *social component,* this is the force emerging from the attraction of the best solution found so far in its neighborhood. These features are shown in Fig. 3.

III. PSO Applied to Phase Recovery

As described by Eberhart and Kennedy, the PSO algorithm is an adaptive algorithm based on a social-psychological metaphor; a population of individuals (referred to as particles) adapts by returning stochastically toward previously successful regions. The fringe demodulation problem is a difficult problem to solve when the noise in the fringe pattern is high, since many solutions are possible even for a single noiseless fringe pattern. Besides, the complexity of the problem is increased when a carrier frequency does not exist (closed fringes are present).

Given that for a closed fringe interferogram there are multiple phase functions for the same pattern, the problem is stated as an *ill-posed* problem in the Hadamard sense, since a unique solution cannot be obtained [26]. It is clear that the image of a fringe pattern $I(x,y)$ will not change if $\phi(x,y)$ in (1) is replaced with another phase function $\hat{\phi}(x,y)$ given by:

$$\hat{\phi}(x,y) = \begin{cases} -\phi(x,y) + 2\pi k & (x,y) \in R \\ \phi(x,y) & (x,y) \notin R \end{cases}, \quad (4)$$

where *R* is an arbitrary region, and *k* is an integer. In this work, PSO is presented to carry out the optimization process, where a parametric estimation of a non-linear function is proposed to fit the phase of a fringe pattern. Then, the PSO technique fits a global non-linear function instead of a local plane to each pixel, just as it is done in regularization techniques [27] and [28]. The fitting function is chosen depending on prior knowledge of the demodulation problem, such as object shape, carrier frequency, pupil size, etc; when no prior information about the shape of $\phi(x,y)$ is known, a polynomial fitting is recommended. In this paper, the authors have used a polynomial fitting to show how the method works.

The purpose in any application of PSO is to evolve a particle swarm of size P (which codifies P possible solutions to the problem) using the update velocity and position of each particle, with the goal of optimizing a fitness function that solves the problem.

In phase demodulation from fringe patterns, the phase data can be approximated by choosing from one of several fitting functions. The fitness function is modeled by the following considerations: a) the similarity between the original fringe image and the genetic generated fringe image, and b) the smoothness in the first and second derivatives of the fitting function.

A. Fitness function

The fitness function U that was utilized in this paper to evaluate the p^{th} particle a^p in the swarm, used an *r-degree* approximation, and is given by:

$$p_r(a,x,y) = a_0 + a_1x + a_2y + a_3x^2 + a_4y^2 + a_5xy + a_6x^2y + a_7xy^2 + \cdots + a_{\left(\frac{(r+1)(r+2)}{2}\right)}y^r \quad (5)$$

Many ways to quantify the quality of fitness function U can be used. We decided to use a term that compares the RMS error between the original fringe pattern and the fringe pattern obtained from the estimated phase:

$$U(a^p) = \sum_{y=1}^{R-1}\sum_{x=1}^{C-1}\left[I_N(x,y) - \cos\left(f(a^p,x,y)\right)\right]^2 \quad (6)$$

where *x, y* are integer values representing indexes of the pixel location in the fringe image. Super-index p is an integer index value between 1 and p, which indicates the number of particles in the swarm. $I_N(x,y)$ is the normalized version of the detected irradiance at point (x,y).

The data from the inferferogram were normalized in the range [-1,1], $R \times C$ is the image resolution whose fringe intensity values are known, and $f(a^p x, y) = p(a^p x, y)$. Additional terms are added to the fitness functions; in this

case, the restrictions for the phase. The fitness function used by Cuevas et al. in [19] incorporates three criteria: similarity, smoothness and overlapped phase similarity with a previously estimated phase. Similarity between fringe patterns is given by equation (6), while smoothness and overlapped phase similarity are expressed by the following equation:

$$R\left(a^p\right)=\sum_{y=1}^{R-1}\sum_{x=1}^{C-1}\begin{Bmatrix}\lambda\Big[\left(f\left(a^p,x,y\right)-f\left(a^p,x-1,y\right)\right)^2\\+\left(f\left(a^p x,y\right)-f\left(a^p x,y-1\right)\right)^2\Big]m\left(x,y\right)\end{Bmatrix} \tag{7}$$

where $R\left(a^p\right)$ is the total amount of restrictions added to the fitness function for a given window whose origin is $\left(r,c\right)$; $m\left(x,y\right)$ is a mask that indicates where the fringe pattern appears inside the image, and λ is a smoothness weight factor (it should be clear for the reader that a higher value of parameter λ implies a smoother function to be fitted).

The third criterion is eliminated in order to simplify the fitness function to get a robust retrieval in just one window. This way, the phase in different windows can be demodulated in parallel. The phase segments are sequentially overlapped. Noise filtering and fringe normalization are solved by using alternative low-pass filtering techniques. We assume smooth phase continuity distributed in first and second derivatives.

The new fitness function can thus be written as:

$$\begin{aligned}U\left(a^p\right)=\alpha-\sum_{y=1}^{R-1}\sum_{x=1}^{C-1}\Bigg\{&\left(I_N\left(x,y\right)-\cos\left(f\left(a^p,x,y\right)\right)\right)^2\\&+\lambda\Big[\left(f\left(a^p,x,y\right)-f\left(a^p,x-1,y\right)\right)^2\\&+\left(f\left(a^p,x,y\right)-f\left(a^p,x,y-1\right)\right)^2\Big]\Bigg\}m\left(x,y\right)\end{aligned} \tag{8}$$

Parameter α must be set to the maximum value of the second term in equation (8). This is done with the aim of converting the problem from a minimal to a maximal optimization question, since a fitness function for PSO is considered to be a non-negative image of merit and profit; this is:

$$\begin{aligned}\alpha=\max_p\Bigg(\sum_{y=1}^{R-1}\sum_{x=1}^{C-1}\Bigg\{&\left(I_N\left(x,y\right)-\cos\left(f\left(a^p,x,y\right)\right)\right)^2\\&+\lambda\Big[\left(f\left(a^p,x,y\right)-f\left(a^p,x-1,y\right)\right)^2\\&+\left(f\left(a^p,x,y\right)-f\left(a^p,x,y-1\right)\right)^2\Big]\Bigg\}\Bigg)m\left(x,y\right)\end{aligned} \tag{9}$$

The first term inside the double summation in equation (9) attempts to keep the local fringe model close to the observed irradiances in the least-squares sense, while the second term is a local discrete difference, which enforces the assumption of smoothness and continuity of the detected phase.

B. *Decoding particles*

As it was said earlier, PSO is used to find the function parameters; in this case, vector a. If we use this function, the particle can be represented as:

$$a=\left[a_0\ a_1\ \ldots\ a_q\right] \tag{10}$$

A *k-bit* bit-string is used to codify a particle value; then, the particle has $q\times k$ bits in length. We define the search space for these parameters. The bit-string codifies a range within the limits of each parameter. The decoded value of the a_i parameter will use the methodology introduced by Toledo and Cuevas in [18], and is:

$$a_i=L_i^B+\frac{L_i^U-L_i^B}{2^k-1}N_i \tag{11}$$

where a_i is the i^{th} parameter real value, L_i^B is the i^{th} bottom limit, L_i^U is the i^{th} upper limit, and N_i is the decimal basis value. These maximum values can be expressed as:

$$L_0^B=-\pi,\quad L_0^U=\pi \tag{12}$$

$$L_i^U=-L_i^B \tag{13}$$

$$L_i^U=\frac{4\pi F}{R1_i^m C1_i^n} \tag{14}$$

where F is twice the maximum number of fringes on the window; the equation is expressed in [18]:

$$F=2\times\max\left(F_x,F_y,\sqrt{F_x^2+F_y^2}\right) \tag{15}$$

F_x and F_y are the maximum fringe numbers in the x and y directions. Finding the value for F automatically is not an easy problem to solve; to our knowledge, there are several algorithms that perform this count, ranging from manual counting by an expert, using a priori knowledge of the phenomenon being measured, even those based on image processing, so in this paper, to get the maximum number of fringes in an image, we propose combining image thresholding described in [29] and connected component labeling, as described in [31] and [32]:

Image thresholding: To binarize the fringe image, we have used Otsu's technique [29], which is known to be based on discriminated analysis. The threshold value t obtained by this method allows partitioning the image into two classes: C_0 and C_1 (i.e., the foreground and background). In other words: $C_0=\{0,1,2,\ldots,t\}$ and $C_0=\{t+1,t+2,\ldots,L-1\}$, where L is the number of gray levels. For an example of the application of Otsu's procedure onto an image, refer to Fig. 5. Fig. 6(a) shows a simple fringe image, while Fig. 6(b) shows the corresponding binary version obtained by Otsu's method.

Connected component labeling: Segmenting a binary image by means of connected component labeling is a standard procedure found in literature. A connected component (CC) is a region of foreground pixels for which a connected path can be found for any two pixels belonging to the region. Finding the connected components in a binary image can be done in many ways ([30], [31] and [32]). The simplest method consists in iteratively replacing each label with the minimum of its 8-connected neighborhood [31]. The algorithm begins with an initial labelling of all 1-pixels, and ends when no more replacements can be made.

In this work, we use the following methodology. Taking as input the binary image, for example the image showen in Fig. 5(a), the algorithm makes a journey through the image from left to right and top to bottom. At each position, a 2×2 neighborhood is analyzed. The positions of the pixels in the neighborhood are: $a_{(i,j)}$, $a_{(i+1,j)}$, $a_{(i,j+1)}$ and $a_{(i+1,j+1)}$ (see Fig. 4). Under 8-connectivity, it is guaranteed that the four pixels are connected.

$a_{i,j}$	$a_{i,j+1}$
$a_{i+1,j}$	$a_{i+1,j+1}$

Fig. 4. 2x2 neighborhood scheme.

To assign the subset that corresponds to each pixel, the following steps are applied:

1. Check the validity of $a_{(i,j)}$, $a_{(i+1,j)}$, $a_{(i,j+1)}$ and $a_{(i+1,j+1)}$. A pixel is valid if $I(i,j)=1$ (it belongs to the foreground), or zero if $I(i,j)=0$ (it belongs to the background).
2. Of the pixels that are valid, check whether one of them has been previously assigned to a given neighborhood. If one or more of the valid pixels have been assigned to a neighborhood, then search for the pixel with the highest number of elements. This is done by using a vector T, which contains all the subsets that have already been assigned, as well as the number of elements in each subset. This facilitates the search.
3. Among the pixels of the neighbourhood that are valid, we search for the pixel whose subset has more elements. To this subset, the other pixels will be assigned.
4. If none of the pixels is assigned to a group, then assign them to a new subset and update the value of the tag in the vector T.
5. Update the values of the subsets and advance one pixel to repeat the steps above.
6. Repeat these steps all over the image.

The result of applying this methodology to an image is shown in Fig. 6(c). The four connected foreground regions appear in different colors. The number of connected components found is a good approximation of the maximum number of fringe patterns in the image.

Alternative way to compute the maximum number of fringes in an image: Another way to find the maximum number of fringes F is as follows. Starting from the central pixel of the fringe image, scan it horizontally, vertically and diagonally, in both directions, as shown in Fig. (7). As an example, in this figure, when we go from the central point to the right, we find a transition from the fringe to the background; as we continue we find a second transition from the background to another fringe. We have thus two fringes. To get the final number, we take into account the considerations given in [1]; by using the interference order for each fringe, we arrive at the end of the swapping that in the example image there are four fringes. This value F can now be used in equation (14) to compute L_i^U. From Equation (13), we can compute L_i^B. Finally, we can substitute these two values in Equation (11) to estimate each a_i. This constitutes an original and very simple procedure to find the components of vector a.

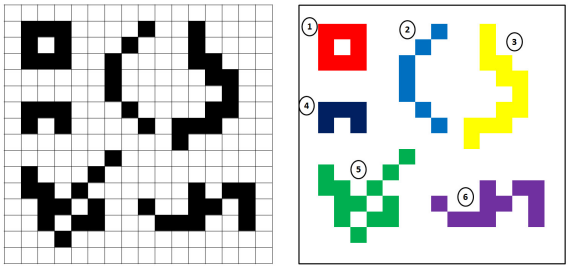


Fig. 5. Example of connected component labelling, a) original image, b) labelling image.

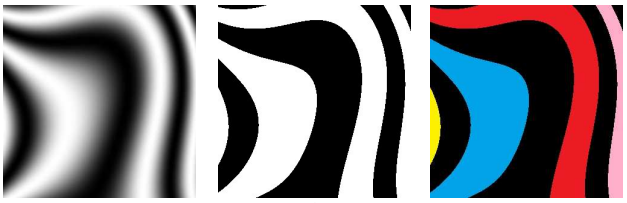


Fig. 6. (a) image of fringe patterns, (b) binary image using Otsu method, (c) labelling image with the result of 4 fringes in the image.

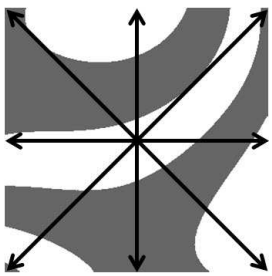


Fig. 7. Direction of sweeps for the location of fringes.

TABLE I
TABLE OF INERTIA AND VELOCITY PARAMETERS

Inertia	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.0001	2.870	3.432	3.612	3.505	3.839	3.277	2.916	2.777	2.395
0.0002	3.007	3.044	3.210	3.083	2.725	2.680	1.688	1.801	2.366
0.0003	1.665	1.875	2.565	2.559	1.576	1.708	**1.151**	1.945	2.469
0.0004	2.170	**1.738**	2.777	1.912	**1.290**	2.171	1.806	**0.567**	1.946
0.0005	1.883	1.860	2.838	1.686	1.701	2.063	1.969	0.791	1.792
0.0006	2.106	2.134	2.900	**1.086**	2.318	1.705	1.645	1.399	2.343
0.0007	1.928	1.993	**0.853**	1.168	2.019	2.270	1.772	1.428	1.828
0.0008	**0.893**	1.938	1.350	1.531	2.019	2.632	1.373	1.373	2.260
0.0009	1.536	1.911	1.436	1.773	2.407	**0.313**	1.902	0.779	**1.523**

For the special case a_0 (i=0), the limits are between $-\pi$ and $+\pi$. a_0 is eliminated from parameter vector a to redefine a new vector a':

$$a' = \left[a_1 \; a_2 \ldots a_q \right] \tag{16}$$

so $p(a,x,y)$ can be expressed as follows:

$$p(a,x,y) = p(a',x,y) + a_0 \tag{17}$$

and replacing (17) into (1):

$$I(x,y) = a(x,y) + b(x,y)\cos\left[p(a',x,y) + a_0 \right], \tag{18}$$

Additionally, a_0 can be expressed as $a_0 = 2\pi l + a_0'$, with l being an integer, and $a_0' < 2\pi$, so equation (18) becomes:

$$I(x,y) = a(x,y) + b(x,y)\cos\left[p(a',x,y) + a_0' + 2\pi l \right], \tag{19}$$

The cosine function is periodical with period 2π, so:

$$I(x,y) = a(x,y) + b(x,y)\cos\left[p(a',x,y) + a_0' \right], \tag{20}$$

In equation (20) demonstrates that limits for a_0 within a range of 2π are enough to represent the phase of the fringe pattern.

C. Convergence

PSO convergence depends mainly on swarm size. Large swarm convergence takes place in smaller number of, but processing time is increased. To stop the PSO process, different convergence measures can be employed. In this paper, we have used a relative error comparison between the fitness function value of the best vectors in the swarm and value a as follows in equation (21), which is the maximum possible value that we can get from equation (8). Thus, we can establish a relative evaluation with uncertainty to stop PSO as:

TABLE II
BEST PARTICLES

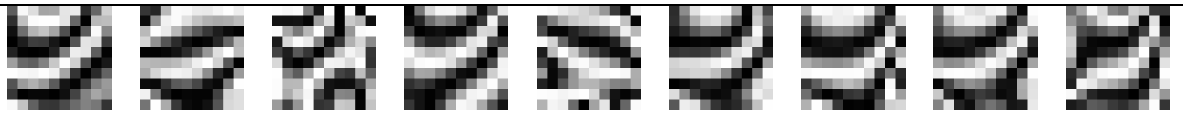
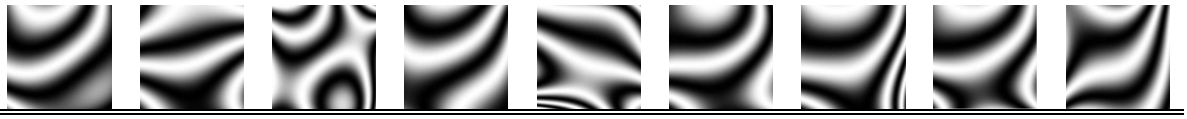
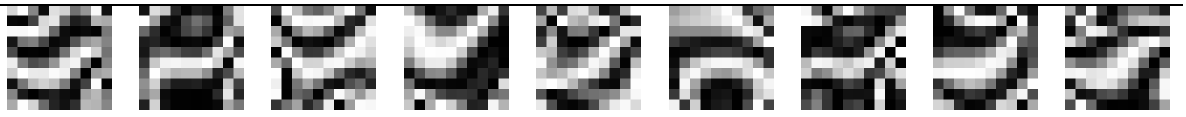
Inertia Velocity	0.1 0.0008	0.2 0.0004	0.3 0.0007	0.4 0.0006	0.5 0.0004	0.6 0.0009	0.7 0.0003	0.8 0.0004	0.9 0.0009
Low-resolution interferogram									
High resolution interferogram									

TABLE III
WORST PARTICLES

Inertia Velocity	0.1 0.0002	0.2 0.0001	0.3 0.0001	0.4 0.0001	0.5 0.0001	0.6 0.0001	0.7 0.0001	0.8 0.0001	0.9 0.0003
Low-resolution interferogram									
High resolution interferogram									

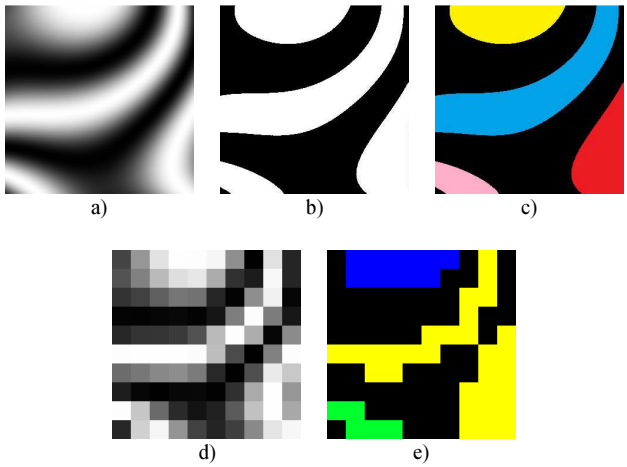


Fig. 8. (a) Image of fringe pattern in resolution , b) binary image using Otsu's method, (c) labeling image with the result of 4 fringes in the image. (d) Low resolution image with sub-Nyquist,(e) labeling image with the result of 3 fringes in the image.

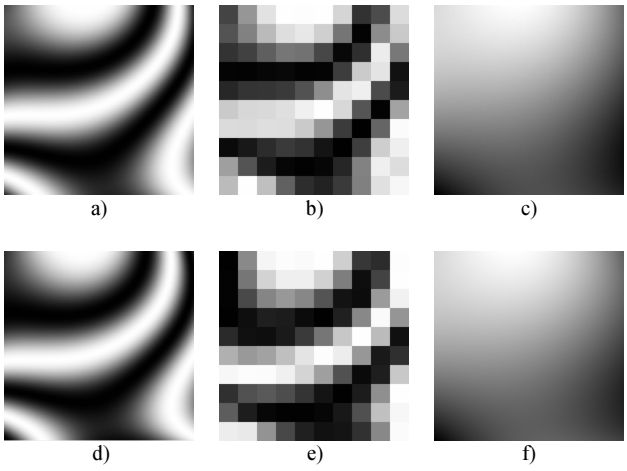


Fig. 9. (a) Observed fringe pattern, (b) Observed fringe pattern in low resolution, (c) its phase map. (d) estimated fringe pattern by PSO, (f) in low resolution and (g) its phase map.

$$\left|\frac{\alpha - U\left(a^{*}\right)}{\alpha}\right| < \varepsilon \tag{21}$$

where $U\left(a^{*}\right)$ is the fitness function value of the best vectors in the swarm in the current iteration, and ε is the relative error tolerance. Additionally, we can stop the process in a specified number of iterations if equation (21) is not satisfied.

IV. Experiments

The proposed method was applied to estimate the phase for a closed fringe pattern. We used a particle swarm size of 100, with 70 iterations, inertia was chosen in the range [0.1 to 0.9], and velocity was a number in the range [0.0001 to 0.0009]. In each particle, the coded coefficients of a fourth degree polynomial were included. The following polynomial was coded in each particle:

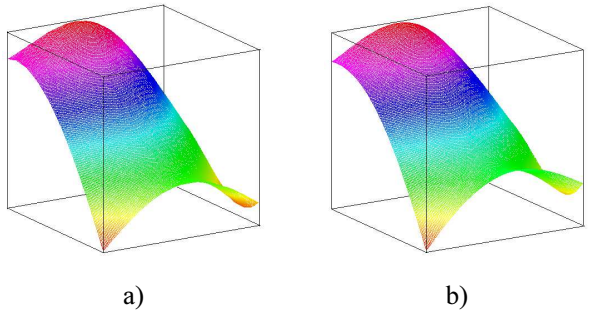


Fig. 10. Phase map observed (a), and phase map estimated by PSO (b).

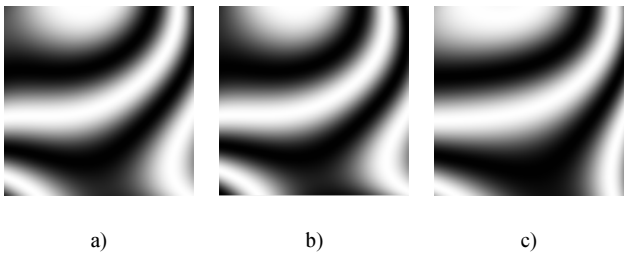


Fig. 11. (a) Observed fringe pattern, (b) estimated fringe pattern by PSO, (c) estimated fringe pattern by GA.

$$\begin{aligned} p_4(x,y) = {} & a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 xy \\ & + a_5 y^2 + a_6 x^3 + a_7 x^2 y + a_8 x y^2 \\ & + a_9 y^3 + a_{10} x^4 + a_{11} x^3 y + a_{12} x^2 y^2 \\ & + a_{13} x y^3 + a_{14} y^4 \end{aligned} \tag{22}$$

The 15 coefficients were configured in each particle inside the swarm to be evolved. As real interferograms present low contrast, and to show that our proposal performs efficiently, a low noise closed fringe pattern was generated using the following expression:

$$I(x,y) = 127 + 63\cos\left(P_4(x,y) + \eta(x,y)\right), \tag{23}$$

where

$$\begin{aligned} p_4(x,y) = {} & 0 - 0.7316x - 0.2801y + 0.0065x^2 \\ & - 0.00036xy - 0.0372y^2 + 0.00212x^3 \\ & + 0.000272x^2 y + 0.001xy^2 - 0.002y^3 \\ & + 0.000012x^4 + 0.00015x^3 y + 0.00023x^2 y^2 \\ & + 0.00011xy^3 + 0.000086y^4 \end{aligned} \tag{24}$$

and $\eta(x,y)$ is the uniform additive noise in the range [–2 radians to 2 radians]. Additionally, the fringe pattern was generated with a low resolution of 10×10 pixels. In this case, we used a parameter search range of [-1 to 1]. The swarm of particles evolved until the number of iterations reached 70, and relative error tolerance ε was 0.05 in equation (21). The fringe pattern and the binary image field of the computer generated interferogram are shown in Figs. 8(a) and 8(b), respectively, with a resolution to 512 × 512, and after applying

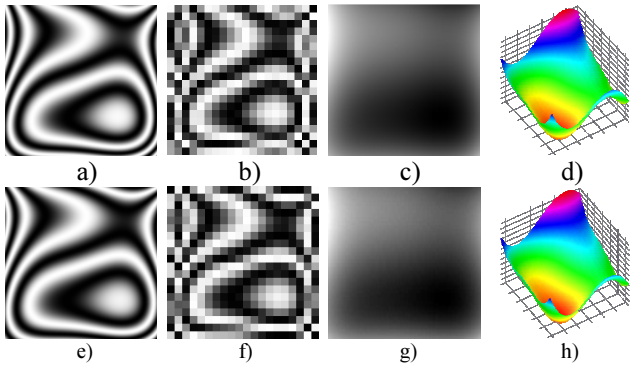


Fig. 12. (a) Observed fringe pattern, (b) observed fringe pattern in low resolution, (c) its phase map, (d) phase in 3D, e) PSO estimated fringe, (f) in low resolution and (g) its phase map and h) phase in 3D.

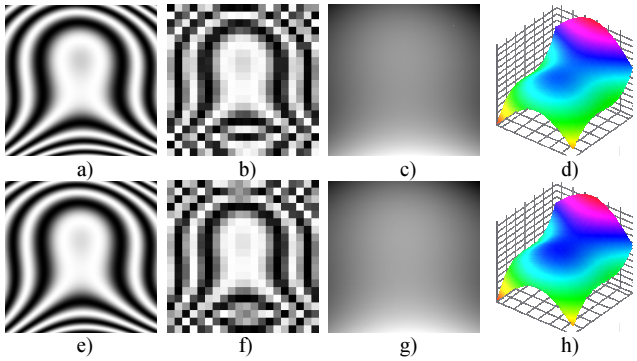


Fig. 13. (a) Observed fringe pattern, (b) observed fringe pattern in low resolution, (c) its phase, (d) phase 3d, e) estimated fringe pattern by PSO, (f) in low resolution and (g) its phase map and h) phase 3d.

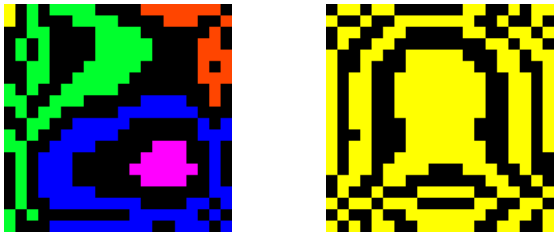


Fig. 14. Labelling images in low-resolution with the result of: (a) 5 fringes in the image show in 12(b), and 1 fringe in the image show in 13(b).

the Otsu's method, we obtain the number of fringes on the image which was 4 number of fringes, this shown in Fig. 8(c). Finally in Figs. 8(d) and Fig. 8(e) shows a sub-sampled image 10x10 and in the connected component labeled obtain the 3, number of fringes.

The fringe pattern and the phase field of the computer generated interferogram are shown in Figs. 9(a) and 9(b), respectively. The PSO technique was used to recover the phase from the fringe pattern. The fringe pattern and the phase estimated through PSO are shown in Figures 9(d), 9(e) and 9(f).

The 3D phase map observed is shown in Fig. 10(a), and the 3D phase map estimated by PSO in Fig. 10(b). Tests are shown in Table 1, the best particles for the testers are shown in Table 2, and Table 3 shows the worst particle for the testers.

Additionally, our method was compared with that proposed by Toledo and Cuevas in [18], which is based on genetic algorithms, and in which, taking into consideration the settings of GA parameters, eight parameters were initialized: number of generations, number of population, cross and mutation rate, type of selection, mutation rate and type of cross. In our case, only four parameters were initialized: iterations (generations), swarm (population), inertia and speed. Finally, during the test an error of 0.4281 was obtained with the GA-based method. With our PSO based proposal, we obtained an error of 0.313.

The Fig. 11(a) shows the original interferogram; figures 11(b) and 11(c) illustrate the result obtained through our method and the result obtained with the GA method introduced in [19]. The interferogram demodulation, in comparison, was almost identical, but the difference is that the image input used with the PSO technique with PSO was recovered from a low level image that had a serious problem of sub-Nyquist in that it no longer distinguished fringes.

The proposed methodology was applied to other images to show its performance. For this, refer to Figs. 12 and 13.

The use of a sub-sample with a high sub-Nyquist problem is something where traditional techniques (Fourier method, Synchronous method and the phase locked loop method) fail; instead, techniques that use GAs have a sub-sampling Nyquist above the limit (one fringe per pixel), as shown in Fig. 14.

Compared with other methods in literature, our method has the advantage that, using a single image, it does not apply any unwrapping module to the phase, and that the polynomial is directly the phase term; it can work with images with high sub-Nyquist, a problem that traditional methods have so far failed to solve.

Execution time is considered fast compared to methods using GAs which is due to the encoding and the image size.

V. Conclusion

A PSO based technique was applied to recover the modulating phase from closed and noisy fringe patterns. A fitness function, which considers prior knowledge about the object being tested, is established to approximate the phase data. In this work, a fourth degree polynomial was used to fit the phase.

A swarm of particles was generated to carry out the optimization process. Each particle was formed by a codified string of polynomial coefficients. Then, the swarm of particles evolved using velocity, position and inertia.

The proposal works successfully where other techniques fail (Synchronous and Fourier methods). This is the case when a noisy, wide bandwidth and/or closed fringe pattern is demodulated. Regularization techniques can be used in these cases, but this proposal has the advantage that the cost function does not depend upon the existence of derivatives and restrictive requirements of continuity (gradient descent methods). Since PSO works with a swarm of possible solutions instead of with a single solution, it avoids falling into a local

optimum. Additionally, no filters and no thresholding operators were required, in contrast with the fringe-follower regularized phase tracker technique.

PSO has the advantage that if the user has prior knowledge of the object shape, then a better suited fitting parametric function can be used instead of a general polynomial function. Additionally, due to the fact that the PSO technique gets the parameters of the fitting function, it can be used to interpolate sub-pixel values and to increase the original phase resolution or interpolate where fringes do not exist or are not valid. A drawback is the selection of the optimal initial PSO parameters (such as swarm size, inertia and velocity) that can increase convergence speed.

ACKNOWLEDGMENT

This paper has been prepared with financial support from the IPN and CONACYT under grants: SIP 20111016, SIP 20121311 and CONACYT 155014. We wish to thank the Centro de Investigación en Computación of the I.P.N. for supporting us on this major accomplishment. We would also like to take the time to give warm thanks for all the support and contribution that given to us by Centro de Investigaciones en Óptica. Additionally, special thanks to Lawrence Whitehill and Mario Ruiz for reviewing English in this document. Finally, thanks to the anonymous reviewers who have given us their constructive criticism on the improvement of this work.

REFERENCES

[1] D. Malacara, *Optical Shop Testing*, New York: Wiley, 1992.

[2] D. Malacara, M. Servin and Z. Malacara, *Interferogram Analysis for Optical Testing*, Marcel Dekker, Ed. New York: CRC Press, 1998.

[3] D.W. Robinson and G.T. Reid, *Interferogram Analysis: Digital Fringe Measurement Techniques*, London: IOP publishing, 1993.

[4] K. Creath, "Phase measurement interferometry techniques," in *Progress in Optics*, vol. 26, E. Wolf, Ed. Amsterdam: Elsevier, pp. 348-393, 1988.

[5] K. Creath, *Interferogram Analysis*, D. Robinson and G.T. Reid (Eds.) London: IOP Publishing, pp. 94, 1993.

[6] M. Takeda, H. Ina and S. Kobayashi, "Fourier-transform method of fringe-pattern analysis for computer based topography and interferometry," *J. Opt. Soc. Am.*, vol.72, pp. 156-160, 1981.

[7] X. Su, W. Chen, "Fourier transform profilometry: a review," *Opt. Laser Eng*., vol. 35, pp. 263-284, 2001.

[8] K.H. Womack, "Interferometric phase measurement using spatial synchronous detection," *Opt. Eng*., vol. 23, pp. 391-395, 1984.

[9] M. Servin and R. Rodriguez-Vera, "Two dimensional phaselocked loop demodulation of interferogram," *J. Mod. Opt.,* vol. 40, pp. 2087-2094, 1993.

[10] D.C. Ghiglia and M.D. Pritt, *Two-dimensional Phase Unwrapping*, New York: John Wiley & Sons, Inc., 1998.

[11] X. Su and L. Xue, "Phase unwrapping algorithm based on fringe frequency analysis in Fourier-transform profilometry," *Opt. Eng*., vol. 40, pp. 637-643, 2001.

[12] D.C. Ghiglia, G.A. Mastin and L.A. Romero, "Cellular automata method for phase unwrapping," *J. Opt. Soc. Am*., vol. 4, pp. 267-280, 1987.

[13] M. Servin, F.J. Cuevas, D. Malacara, J.L. Marroquin and R. Rodriguez-Vera, "Phase unwrapping through demodulation using the RPT technique," *Applied Optics*, vol. 38, pp. 1934-1940, 1999.

[14] F.J. Cuevas, M. Servin, O.N. Stavroudis and R. Rodríguez-Vera, "Multilayer neural network applied to phase and depth recovery from fringe patterns," *Optics Communications*, vol. 181, pp. 239–259, 2000.

[15] F.J. Cuevas, M. Servin and R. Rodríguez-Vera, "Depth recovery using radial basis functions," *Opt Commun*., vol. 163, pp. 270–277, 1999.

[16] F.J. Cuevas, J.H. Sossa-Azuela and M. Servin, "A parametric method applied to phase recovery from a fringe pattern based on a genetic algorithm," *Opt Commun*., vol. 203, no. 3-6, pp. 231–239, 2002.

[17] L.E. Mancilla, J.M. Carpio and F.J. Cuevas, "Demodulation of Interferograms of Closed Fringes by Zernike Polynomials using a technique of Soft Computing," *Engineering Letters,* vol. 15, no. 1, pp. 99-104, 2007.

[18] L.E. Toledo and F.J. Cuevas "Optical Metrology by Fringe Processing on Independent Windows Using a Genetic Algorithm," *Experimental mechanics*, vol. 48, pp. 559-569, 2008.

[19] F.J. Cuevas, F. Mendoza, M. Servin and J.H. Sossa-Azuela, "Window fringe pattern demodulation by multi-functional fitting using a genetic algorithm," *Opt. Commun*., vol. 261, pp. 231–239, 2006.

[20] J. Kennedy and R.C. Eberhart, "Particle swarm optimization," in *Proc. IEEE Intl. Conf. on Neural Networks*, vol. 4. Piscataway, NJ: IEEE Service Center, pp. 1942-1948, 1995.

[21] R.C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory controllers," in *Proc. Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan, Piscataway, NJ: IEEE Service Center, pp. 39-43, 1995.

[22] R.C. Eberhart and Y. Shi, "Comparison between genetic algorithms and particle swarm optimization," in *Evolutionary programming*, V.W. Porto, N. Saravanan, D. Waagen, and A. E. Eiben (Eds.), 1998.

[23] Y. Shi, and R.C. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the IEEE International Conference on Evolutionary Computation*, Piscataway, NJ: IEEE Press. 1998, pp. 69-73.

[24] Y. Shi and R.C. Eberhart, "Empirical study of particle swarm optimization," in *Proceedings of the 1999 Congress on Evolutionary Computation*, Piscataway, NJ: IEEE Service Center, 1999, pp. 1945–1950..

[25] Y. Shi and R.C. Eberhart, "Particle Swarm Optimization with Fuzzy Adaptive Inertia Weight", in *Proceedings of the Workshop on Particle Swarm Optimization*, Indianapolis, IN: Purdue School of Engineering and Technology, IUPUI press, 2011.

[26] J. Hadamard, *Sur les problems aux derivees partielles et leur signification physique,* Princeton University Bulletin, Princenton, 1902.

[27] M. Servin, J.L. Marroquin and F.J. Cuevas, "Fringe-follower regularized phase tracker for demodulation of closed-fringe interferograms," *J. Opt. Soc. Am. A*., vol. 18, pp. 689-695, 2001.

[28] M. Servin, J.L. Marroquin and F.J. Cuevas, "Demodulation of a single interferogram by use of a two-dimensional regularizad phase-tracking technique," *Appl. Opt*., vol. 36, no. 19, pp.4540-4548, 1997.

[29] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on System, Man, and Cybernetics,* SMC vol. 9, no. 1, 1979.

[30] A. Glassner, "Fill 'Er Up!" *IEEE Computer Graphics and Applications*, vol. 21, no. 1, pp. 78-85, Jan./Feb. 2001.

[31] R.M Haralick and L.G. Shapiro, *Computer and Robot Vision*, Vol. 1, Addison-Wesley, 1992.

[32] A. Rosenfeld and J.L. Pfalz, "Sequential Operations in Digital Picture Processing," *Journal of the ACM*, vol. 13, no. 4, pp. 471-494, 1966.

Journal Information and Instructions for Authors

I. Journal Information

Polibits is a half-yearly open-access research journal published since 1989 by the *Centro de Innovación y Desarrollo Tecnológico en Cómputo* (CIDETEC: Center of Innovation and Technological Development in Computing) of the *Instituto Politécnico Nacional* (IPN: National Polytechnic Institute), Mexico City, Mexico.

The journal has double-blind review procedure. It publishes papers in English and Spanish (with abstract in English). Publication has no cost for the authors.

A. Main Topics of Interest

The journal publishes research papers in all areas of computer science and computer engineering, with emphasis on applied research. The main topics of interest include, but are not limited to, the following:

- Artificial Intelligence
- Natural Language Processing
- Fuzzy Logic
- Computer Vision
- Multiagent Systems
- Bioinformatics
- Neural Networks
- Evolutionary Algorithms
- Knowledge Representation
- Expert Systems
- Intelligent Interfaces
- Multimedia and Virtual Reality
- Machine Learning
- Pattern Recognition
- Intelligent Tutoring Systems
- Semantic Web
- Robotics
- Geo-processing
- Database Systems
- Data Mining
- Software Engineering
- Web Design
- Compilers
- Formal Languages
- Operating Systems
- Distributed Systems
- Parallelism
- Real Time Systems
- Algorithm Theory
- Scientific Computing
- High-Performance Computing
- Networks and Connectivity
- Cryptography
- Informatics Security
- Digital Systems Design
- Digital Signal Processing
- Control Systems
- Virtual Instrumentation
- Computer Architectures

B. Indexing

The journal is listed in the list of excellence of the CONACYT (Mexican Ministry of Science) and indexed in the following international indices: LatIndex, SciELO, Periódica, and e-revistas.

There are currently only two Mexican computer science journals recognized by the CONACYT in its list of excellence, *Polibits* being one of them.

II. Instructions for Authors

A. Submission

Papers ready to review are received through the Web submission system on www.easychair.org/conferences/?conf=polibits1; see also updated information on the web page of the journal, www.cidetec.ipn.mx/polibits.

The papers can be written in English or Spanish. In case of Spanish, English title, author names, abstract, and keywords must be provided; in recent issues of the journal you can find examples of how it is done.

Only full papers are reviewed; abstracts are not considered as submissions. The review procedure is double-blind. Therefore, papers should be submitted without names and affiliations of the authors and without any other data that reveal the authors' identity.

For review, a PDF file is to be submitted. In case of acceptance, the authors will need to upload the source code of the paper, either Microsoft Word or TeX with all supplementary files necessary for compilation. Upon acceptance notification the authors receive further instructions on uploading the camera-ready source files.

Papers can be submitted at any moment; if accepted, the paper will be scheduled for inclusion in one of forthcoming issues, according to availability and the size of backlog. While we make every reasonable effort for fast review and publication, we cannot guarantee any specific time for this.

B. Format

The journal uses the IEEE Template for all Transactions except IEEE Transactions on Magnetics, www.ieee.org/web/publications/authors/transjnl/index.html (while the journal uses this format for submissions, it is in no way affiliated with, or endorsed by, IEEE).

There is no specific page limit: we welcome both short and long papers, provided that the quality and novelty of the paper adequately justifies its length. Usually the papers are between 10 and 20 pages; much shorter papers often do not offer sufficient detail to justify publication.

The editors keep the right to copyedit or modify the format and style of the final version of the paper if necessary.