Editorial

THIS issue of *Polibits* includes a selection of papers related to the topic of processing of semantic information. Processing of semantic information involves usage of methods and technologies that help machines to understand the meaning of information. These methods automatically perform analysis, extraction, generation, interpretation, and annotation of information contained on the Web, corpus, natural language systems, and other data.

The **special section** of this issue consists of six papers dedicated to **processing of semantic information**. The first four papers present new proposals on processing of semantic information using corpora. The fifth paper analyses opinions. The final paper of this section use classification rules for creation of conceptual graphs.

The paper "*Spoken to Spoken vs. Spoken to Written: Corpus Approach to Exploring Interpreting and Subtitling*" deals with corpora of Finnish-Russian interpreting discourse and subtitling. The software package developed for processing of the corpora includes routines specially written for studying speech transcripts rather than written text. For example, speaker statistics function calculates number of words, number of pauses, their duration, and average speech time of a certain speaker.

The paper "*Semi-Automatic Parallel Corpora Extraction from Comparable News Corpora*" develops an effective technique that extracts parallel corpus between Manipuri, a morphologically rich and resource constrained Indian language, and English from comparable news corpora collected from the Web.

The paper "*A Natural Language Dialogue System for Impression-based Music Retrieval*" evaluates a natural language dialogue system with 164 impression words, 14 comparative expressions, such as "a little more" and "more and more," and modifies the most recently used query vector through a dialogue. Also, the paper evaluates performance using 35 participants to determine the effectiveness of the proposed dialogue system.

The paper "*Retrieving Lexical Semantics from Multilingual Corpora*" proposes an unsupervised technique for building a lexical resource like *WordNet* used for annotation of parallel corpora. The reported results are for English, German, French, and Greek using the *Europarl* parallel corpus. The multilingual aspect of the approach helps in reducing the ambiguity inherent in any words/phrases in the English language.

The research presented in the paper "*Opinion Mining using Ontologies*" analyses opinions using an innovative approach based on ontology fusion and matching. The proposed method allows two enterprises to share and merge the results of opinion analyses on their own products and services.

The paper "*Learning of Chained Rules for Construction of Conceptual Graphs*" studies chained rules for generating new rules that can help to construction of conceptual graphs. The proposed supervised method is based on the inclusion of chained rules. The rules are defined on the basis of three elements: the role of dialing or holding the word in the sentence, the standard conceptual graph, and the definition of an object that functions as a black box of graphs.

The section of **regular papers** includes three papers.

The first paper "*On a Framework for Complex and ad hoc Event Management over Distributed Systems*" provides a framework for event-based communications, and at the same time new advantages with respect to the existing standards such as composition, interoperability and dynamic adaptability. The proposed framework detects general and flexible event which can be adapted to specific requirements and situations. Within the framework, the main aspects of event management over distributed systems are treated, such as event definition, detection, production, notification and history management. Other aspects such as event composition are also discussed.

The second paper "*Computer System for Analysis of Holter Cardiopathy*" describes a medical tool related to cardiopathy studies that is available and accessible to any hospital, medical center, or doctor's office, has accessible cost, is a user friendly and understandable. As a benefit for patients, this tool allows major accessibility of such studies. Also, this paper reports how professional staff can obtain in certain cases a possible diagnosis.

Finally, the paper "*Prediction of Failures in IP Networks using Artificial Neural Networks*" presents the implementation of a system for predicting timeout failures and rejection of connections in LAN, using multilayer perceptron configuration of neural networks. It describes the implementation of the system, experiments conducted for the selection of specific parameters of the neural network, training algorithm and results.

Yulia Ledeneva
Research Professor,
Autonomous University of the State of Mexico,
Mexico

Spoken to Spoken vs. Spoken to Written: Corpus Approach to Exploring Interpreting and Subtitling

Mikhail Mikhailov, Hannu Tommola, and Nina Isolahti

***Abstract*—The need for corpora of interpreting discourse in translation studies is gradually increasing. The research of AV translation is another rapidly developing sphere, thus corpora of subtitling and dubbing would also be quite useful. The main reason of the lack in such resources is the difficulty of obtaining data and the inevitability of manual data input. An interpreting corpus would be a collection of transcripts of speech in two or more languages with part of the transcripts aligned. The subtitling and dubbing corpora can be designed using the same principles. The structure of the corpus should reflect the polyphonic nature of the data. Thus, markup becomes extremely important in these types of corpora. The research presented in this paper deals with corpora of Finnish-Russian interpreting discourse and subtitling. The software package developed for processing of the corpora includes routines specially written for studying speech transcripts rather than written text. For example, speaker statistics function calculates number of words, number of pauses, their duration, average speech tempo of a certain speaker.**

***Index terms*—Interpreting, subtitling, corpora, Russian language, Finnish language.**

I. Introduction

COMPILING **written** text corpora has become a relatively easy technical task in the last decades. Some of published texts are ready available in digital form, other can be digitized with the help of scanning and OCR software. Plenty of texts of different genres written in all imaginable languages are being accumulated on the web. It is even possible to collect so called web-corpora in automated mode from the Internet (see works by Adam Kilgariff, William Fletcher, Marco Baroni, e.g. [1]). Text corpora exceeding 100 millions running words in size are quite common today[1].

As regards compiling **spoken** corpora, it remains hard, time-consuming, expensive and extremely slow work. As opposed to written resources, not many ready-made transcripts of spoken language are available (e.g. speeches of politicians, TV interviews, etc.), and most of those are adaptations of oral speech into written form and have to be matched with the recordings. As opposed to written resources, transcripts or oral speech are not subject to amateurish collecting. Recording and transcribing of oral speech remain scholars' activity. Although the quality of sound recording and possibilities for data storage have greatly improved during the last decades, speech recognition technologies are still under development, error rate is considerably high [2]. The speech recognition systems are not being developed for converting spontaneous speech into textual form, but rather for the purposes of dictation. The commercial software is still quite expensive (see e.g. http://www.enablemart.com/Voice-Recognition). Besides, if the transcribed discourse is multilingual, additional technical problems have to be solved. So, in most cases the transcribing is to be performed manually for the time being.

English language resources dominate in **spoken corpora**, which is quite predictable. It would be enough to mention Cambridge International Corpus (CANCODE, http://www.cambridge.org/elt/corpus/cancode.htm), Diachronic Corpus of Present-Day Spoken English (DCPSE, http://www.ucl.ac.uk/english-usage/projects/dcpse/index.htm), Michigan Corpus of Academic Spoken English (MICASE, http://quod.lib.umich.edu/m/micase/). A considerable list of spoken corpora can be found at http://corpus-linguistics.de/html/corp/corp_spoken.html. Compiling of non-English spoken corpora is lagging behind. The research presented in this paper deals with two languages, Finnish and Russian, which are no exception. The Russian National Corpus includes a spoken subcorpus of about 6 million running words (www.ruscorpora.ru, [3]). Transcripts of Finnish speech are available from the Finnish Broadcast Corpus (Finnish Bank of Language, http://www.csc.fi/english/research/software/fbc). Most of the other existing collections of transcripts of prepared and spontaneous speech are of modest size and with only basic search interface or no search interface at all.

Not surprisingly, the tools and methodology used in spoken corpus research are developed along the same lines as the tools for processing written language. The transcripts are regarded as a sort of written texts. Many of the spoken corpora do not even use any transcribing conventions (e.g. MICASE).

The research of interpreting is a quite important part of translation studies. However, **interpreting corpora** are still quite a new kind of language resources and thus far not much quantitative data is available. We would like to mention the European Parliament Interpreting Corpus (EPIC) as one of the

Manuscript received March 7, 2010. Manuscript accepted for publication May 31, 2010.

All authors are with the School of Modern Languages and Translation Studies of the University of Tampere, Finland (e-mail: mikhail.mikhailov@uta.fi, hannu.tommola@uta.fi, nina.isolahti@uta.fi).

[1] Of course, corpus tagging and annotation is still a problem, and corpus research is still limited by the lack of annotated corpora.

very few examples. The corpus consists of speeches at the European Parliament interpreted into English, Italian, and Spanish, and is arranged as a parallel corpus (http://sslmitdev-online.sslmit.unibo.it/corpora/corporaproject.php?path= E.P.I.C.). The lack of the language resources makes it difficult to obtain extensive research data, to say nothing of data processing facilities pertaining to interpreting. Consequently, there is a huge demand for more electronic data to be employed in interpreter training and interpreting research.

Besides conventional converting of written text in one language into written text in another language (translation) and converting of oral speech in one language into oral speech in another language (interpreting) there exist other types of translating. Written text may be interpreted impromptu, speech may be translated into a text by means of subtitling or speech-to-text reporting[2].

Subtitling is an important type of non-conventional translating especially in the countries where it is not common to dub movies and tv-programs. It is important to mention, that subtitling is in many ways different from conventional written translation, see e.g. [4]. With the growth of the market for audiovisual products, subtitling has become an object of research, and corpus data is needed. The Open Source Parallel Corpus (OPUS) includes a parallel corpus of subtitles (http://urd.let.rug.nl/tiedeman/OPUS/cwb/OpenSubtitles/frames-cqp.html) [5].

In this paper, another important kind of resources is introduced. The data is arranged as parallel corpus with speech and subtitles aligned. The Subtitling corpus we are designing presents a new type of language resource with both the speech transcripts and the subtitles included. When investigating subtitled material it is important to have access not only to a film script but to the entire audiovisual message. With that purpose in mind, this corpus should consist of an exact transcript of the film dialogue as well as the relevant information on its other auditive and visual elements. This data would be aligned with the subtitles. We know nothing about existing corpora of this kind.

II. Research Methods and Material

A. Research Methods

The research of the data supplied in the interpreting corpora shall not be confined to examining corresponding passages in the original speech and in the speech of the interpreter. The holistic approach taken would study the communication between the participants and the interpreter, the message transmission via interpreting, the communicative failures during interpreting, the extralinguistic activities of the communicants, etc. The audiovisual translation analysis would also take into account the visual channel and the pressure on the recipient, who has to read the subtitles at the same time as s/he watches the movie. Apart from methods in functional theories of translation, some directions of established linguistic theory will also be suitable in analyzing and interpreting the research results: discourse and conversation analysis, linguistic pragmatics, theory of speech acts, etc.

B. Research Material

A number of Finnish-Russian electronic corpora: the Corpus of court interpreting (CIC), the Corpus of learners' interpreting (CLI), and the Corpus of film transcripts and subtitles (FiTS), are currently being collected and placed on the web site of the project.

The structure of the database is established and a pilot version of the search engine for the spoken corpora has been developed. The data is currently stored on the server of the Russian Section of the Department of Translation Studies (https://mustikka.uta.fi/spoken/, access restricted to the members of the research team).

III. Composition of the Corpora

In institutional interpreting contexts, part of communication often takes place in one language without the help of the interpreter, who takes part in discussion when needed. Even when the interpreter does take part in the communication, the process is often not as smooth, as it might be expected. The speakers often interrupt each other, and the interpreter works under constant pressure. The interpreting discourse is thus a sophisticated mixture of verbal and non-verbal communication, part of which is mediated by the interpreter (see e.g. [6]–[8]).

The same features can be found in a film with subtitles. It is a very complicated stream of information: visual images, sounds, verbal and non-verbal communication of the characters, speech of the narrator, text as part of original film, and subtitles (see [9]). Subtitling is a very specific kind of activity, and the subtitler must be aware both of communication problems and technical issues (see [10] and [11]).

In many respects these two kinds of data – interpreting discourse and a film with subtitles – can be reproduced in corpus databases of the same structure. Such a corpus can be arranged as a hybrid of a bilingual corpus and a parallel one. Thus, an interpreting corpus would be a collection of transcripts of speech in two (or even more) languages, and some of the transcripts would be aligned [12]. A corpus of film transcripts and subtitles would be a synthesis of a spoken corpus (transcripts), a text corpus (subtitles), and a parallel corpus (aligned transcripts and subtitles).

Audio and visual components would in many cases be extremely useful additions to the corpus data. Unfortunately, it is not always possible to include them due to problems of ethical, copyright, and technical nature. However, remarks and comments are seriously considered as a part of corpus structure. All above mentioned issues make the architecture of the corpus quite sophisticated and the mark-up vitally important.

The transcripts are annotated using xml markup. The transcription is broad; however, speech is not smoothed up to

[2] A new form of communication used for communication between deafened and hearing people.

written language as it happens in many projects, which do not directly contribute to linguistic research. We mark pauses and their lengths as well as some prosodic features (logical accent, rising/falling pitch, etc.). No punctuation marks are used in the transcripts but question and exclamation marks, which make reading easier. The features relevant from the point of view of translation process are also subject to markup, these are deletions, additions, changes, etc.

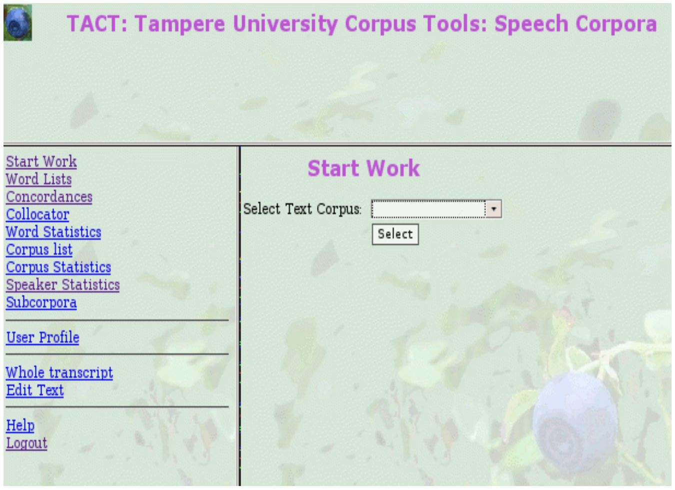


Fig. 1. TACT: User interface.

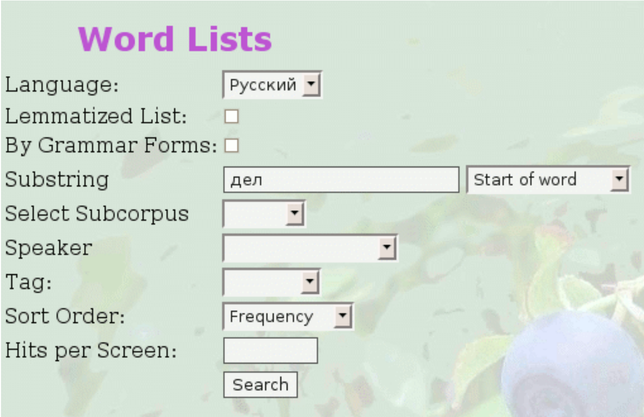
Nonetheless, xml document is not the final representation of the corpus, which is stored in a database format. The reason why transcripts are not fed into the database directly is the relative ease of markup in xml, which can be done in any word processor. It is also quite a simple task to check the consistency of the markup. So, the data extracted from xml files are uploaded to Postgresql databases (http://www.postgresql.org). The database handles many different routines like data maintenance, search, corpus users, sessions, etc. The most important for the search engine database tables are the following:

- Transcripts. Each running word, pause, tag is stored in a separate record. This makes it possible to build concordances, word lists, calculate statistics using SQL queries.
- Phrases. The start and end of each phrase is marked in Transcripts table with special tags and all the data on the phrase (speaker, timing, duration, etc.) are stored in a separate table.
- Lemmas. The lemmas of the word tokens are stored in separate tables linked to the Transcripts table. This makes the Transcripts table more transparent, saves space on disk, and simplifies generating of lemmatized lists. Lemmatization is performed after tokenization with the help of external software. English and Finnish texts are lemmatized with Connexor software (http://www.connexor.eu/technology/machinese/ machinesephrasetagger/), German with Morphy (http://www.wolfganglezius.de/ doku.php?id=cl:morphy, [13]), Russian with Rmorph (http://www.cic.ipn.mx/~sidorov/rmorph/index.html, [14]).
- Library. The information on each item of the corpus (e.g. a film, an interview, a hearing at the Court, etc.) is stored in a separate table. The data available is text code in the corpus, title, author (for written text), date of issue, as well as text statistics (number of characters, number of running words, etc.).

IV. Corpus Tools

The maintenance of the corpus database (tokenization, lemmatization, updating statistics, etc.) is performed by running php-scripts in terminal window.

The most important and frequently used search routines are included into the TACT web interface (Tampere University Corpus Tools, developed by Mikhail Mikhailov). Not surprisingly, a written-language bias in the tools and methodology of spoken corpus research is quite obvious. We mean that same tools are used for processing spoken corpora as for the written ones. Some of the spoken corpora do not even use transcribing conventions (e.g. MICASE). The TACT package also includes routines, which can be used for processing of written texts as well. However, certain functions were developed specially for spoken corpora. The software package is being constantly modified, and new functions are added to meet the requirements of the research team. Most of the functions work both with the whole corpus and with subcorpora (i.e. groups of texts defined by the user). The following research tools are currently available:



Word Lists

Language: Русский
Lemmatized List: □
By Grammar Forms: □
Substring дел Start of word
Select Subcorpus
Speaker
Tag:
Sort Order: Frequency
Hits per Screen:

Fig. 2. Word Lists. Dialog.

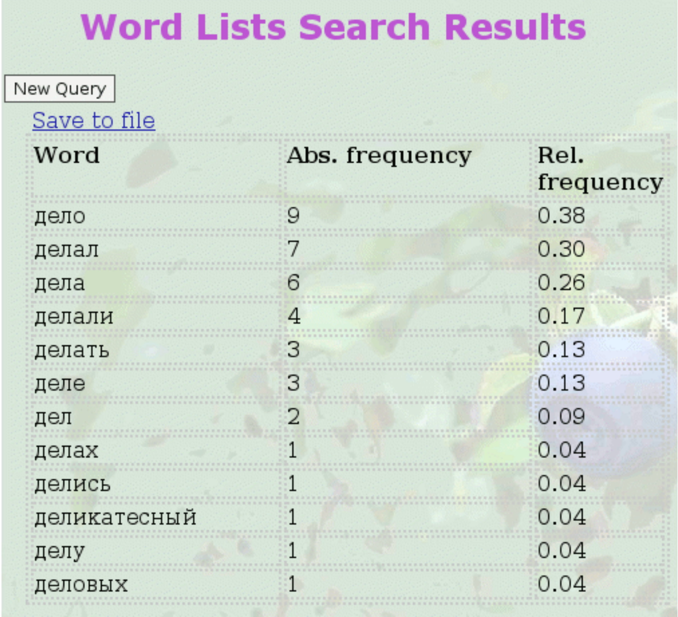
The following tools are currently available:

- Word lists;
- Concordances;
- Collocation lists (though not very helpful due to the modest size of the corpus);
- Corpus statistics;
- Speaker statistics
- etc. (see Fig. 1).

Some of the tools are more relevant for processing written texts; some have been substantially revised for the purpose of studying speech transcripts.

A. *Word Lists*

This tool is more flexible than standard applications for building word lists.



Word Lists Search Results

Word	Abs. frequency	Rel. frequency
дело	9	0.38
делал	7	0.30
дела	6	0.26
делали	4	0.17
делать	3	0.13
деле	3	0.13
дел	2	0.09
делах	1	0.04
делись	1	0.04
деликатесный	1	0.04
делу	1	0.04
деловых	1	0.04

Fig. 3. Word Lists. Search Result.

It is possible to generate frequency lists of word tokens, running words, grammar tags, or even frequency lists of tokens marked by certain tags. The utility generates frequency lists for the whole corpus or for a subcorpus. Sometimes it might be quite helpful to obtain a frequency list for a certain speaker. It is no need to waste time on generating the whole list if the researcher is interested only in most frequent words, or in the words following certain pattern. On Fig. 2 the user is requesting a Russian unlemmatized frequency list of words starting with string *del* ordered according to frequency.

The resulting frequency list is displayed on Fig. 3; in addition to the absolute frequencies the relative frequency per 1000 words is calculated and shown as well.

B. *Concordances*

It is much more difficult to present a readable concordance derived from a speech transcript than from a written text. Moreover, the interpreter's speech has to be detached from the source speech.

The solution we suggest is to use two-column presentation with source speech in the left column and interpreting in the right one (see Fig. 4). The rise and fall of the tone, emphasis and other prosodic features are also visualized. The problem of presenting speech overlapping remains unsolved; overlaps are currently marked with brackets, which is not very user-friendly.

C. *Speaker Statistics*

The most interesting research tool of the TACT application is the utility presenting speaker statistics. It calculates speaker's speech tempo, number of pauses, length of pauses and other parameters. For the interpreter the script calculates statistics separately for all languages he/she speaks during the hearings.

This tool is significant for studying interpreting, whereas it is less relevant with subtitling, although it might be of use in linguistic research of film transcripts.

V. Current State of the Project

The corpora are currently being collected at the School of Modern Languages and Translation Studies of the University of Tampere as graduate and post-graduate research.

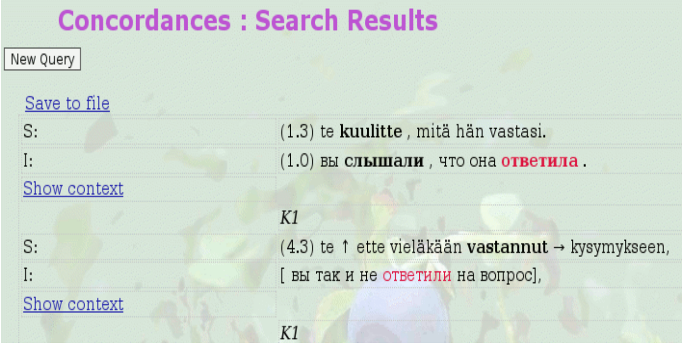
Court Interpreting Corpus. Currently nine hearings (about 48,989 running words) have been transcribed, tagged, and placed on the server.

Corpus of Film Transcripts and Subtitles: Three films (*Brat / The Brother*, *Kukushka / The Cuckoo*, and *Osobennosti natsional'noj ohoty / Peculiarities of the national hunting*) have been transcribed and aligned with the Finnish subtitles.

Corpus of Learners' Interpreting: Two talks with consecutive interpreting by three students of the Russian Translation Studies have been recorded transcribed and uploaded to the corpus database. Although the corpus is quite small, 12,531 running words, it is richly annotated with additions, deletions, and changes tagged.

Although the amount of material is modest, it is unique in many respects, and some interesting results have already been obtained. However limited the available data is, it enables some implementation of quantitative methods in studying interpreting and subtitling.

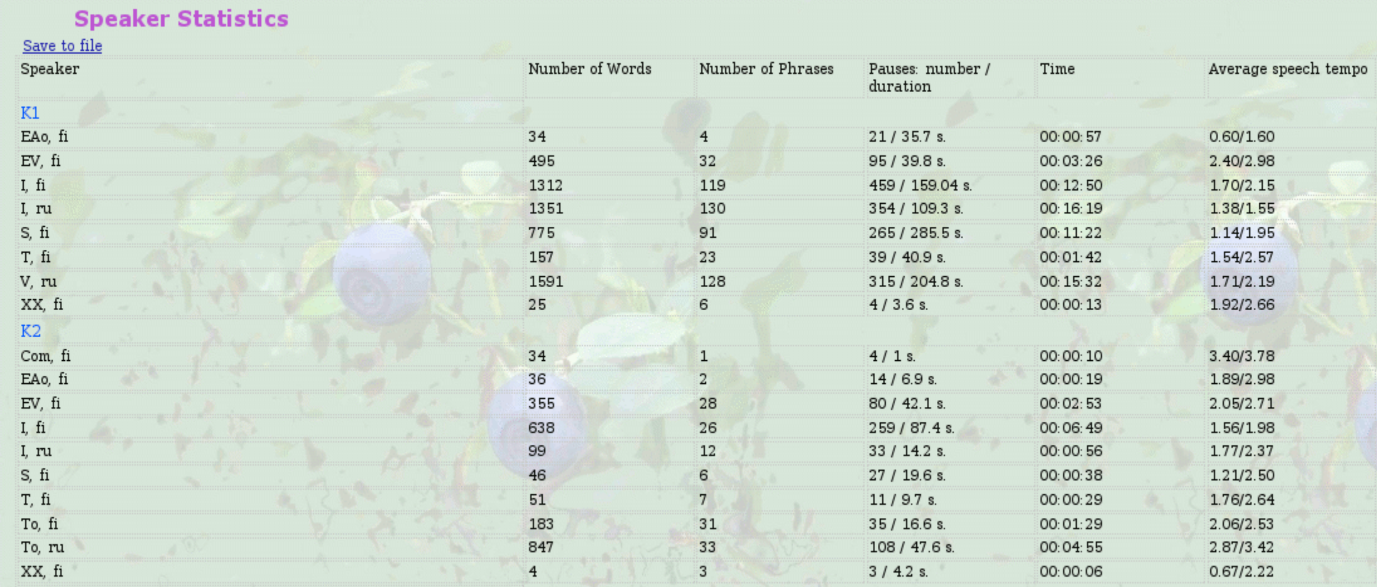
The project also sets new challenges in developing an efficient, robust and flexible search engine for processing interpreting and subtitling corpora, i.e. electronic corpora with transcripts of discourse with consecutive and/or simultaneous interpreting or subtitles.



Concordances : Search Results

S:	(1.3) te **kuulitte** , mitä hän vastasi.
I:	(1.0) вы **слышали** , что она **ответила** .
Show context	
	K1
S:	(4.3) te ↑ ette vieläkään **vastannut** → kysymykseen,
I:	[вы так и не ответили на вопрос],
Show context	
	K1

Fig. 4. Concordance.



Speaker Statistics

Save to file

Speaker	Number of Words	Number of Phrases	Pauses: number / duration	Time	Average speech tempo
K1					
EAo, fi	34	4	21 / 35.7 s.	00:00:57	0.60/1.60
EV, fi	495	32	95 / 39.8 s.	00:03:26	2.40/2.98
I, fi	1312	119	459 / 159.04 s.	00:12:50	1.70/2.15
I, ru	1351	130	354 / 109.3 s.	00:16:19	1.38/1.55
S, fi	775	91	265 / 285.5 s.	00:11:22	1.14/1.95
T, fi	157	23	39 / 40.9 s.	00:01:42	1.54/2.57
V, ru	1591	128	315 / 204.8 s.	00:15:32	1.71/2.19
XX, fi	25	6	4 / 3.6 s.	00:00:13	1.92/2.66
K2					
Com, fi	34	1	4 / 1 s.	00:00:10	3.40/3.78
EAo, fi	36	2	14 / 6.9 s.	00:00:19	1.89/2.98
EV, fi	355	28	80 / 42.1 s.	00:02:53	2.05/2.71
I, fi	638	26	259 / 87.4 s.	00:06:49	1.56/1.98
I, ru	99	12	33 / 14.2 s.	00:00:56	1.77/2.37
S, fi	46	6	27 / 19.6 s.	00:00:38	1.21/2.50
T, fi	51	7	11 / 9.7 s.	00:00:29	1.76/2.64
To, fi	183	31	35 / 16.6 s.	00:01:29	2.06/2.53
To, ru	847	33	108 / 47.6 s.	00:04:55	2.87/3.42
XX, fi	4	3	3 / 4.2 s.	00:00:06	0.67/2.22

Fig. 5. Speaker statistics.

Subtitling is a major means of inter-cultural communication and an extremely widely read text type in 'subtitling countries' such as Finland. There is a great need for systematic data to help improve subtitle quality and understand the subtitling process and audience expectations. The parallel corpus of transcripts and subtitles, which combines spoken and written data, is an entirely new type of language resource promising an important step forward in subtitling research.

We believe that the corpora can be used both directly and indirectly in Interpreting and Translation Studies: in training of interpreters and subtitlers, and in theoretical descriptions of the structure of multilingual and multimediated discourse.

REFERENCES

[1] J. Pomikálek, P. Rychlý and A. Kilgarriff, *"Scaling to Billion-plus Word Corpora,"* in *Advances in Computational Linguistics. Special Issue of Research in Computing Science,* Vol 41, Mexico City. 2009. Available: http://www.kilgarriff.co.uk/Publications/2009-PomikalekRychlyKilg-MexJournal-ScalingUp.pdf

[2] E. G. Devine, S. A. Gaehde, and A. C. Curtis, "Technology Evaluation: Comparative Evaluation of Three Continuous Speech Recognition Software" in *Packages in the Generation of Medical Reports JAMIA* 2000, pp. 462-468.

[3] E. Grišina, "Ustnaja reč v Nacional'nom korpuse russkogo jazyka," *Nacional'nyj korpus russkogo jazyka:* 2003—2005. M.: Indrik, 2005.

[4] Y. Gambier, "Challenges in research on audiovisual translation," in *Translation research projects*, Tarragona, 2009, pp. 17—27.

[5] J. Tiedemann, "Improved Sentence Alignment for Movie Subtitles," in *Proceedings of RANLP '07*, Borovets, Bulgaria, 2007. http://urd.let.rug.nl/tiedeman/OPUS/.

[6] R. González, V. F. Vásquez, H. Mikkelson, *Fundamentals of Court Interpretation. Theory, Policy, and Practice*, Durham, North Carolina: Carolina Academic Press, 1991.

[7] S. Hale, *The Discourse of Court Interpreting. Discourse practices of the law, the witness and the interpreter,* Amsterdam Philadelphia: John Benjamins, 2004.

[8] T. R. Välikoski, *The Criminal Trial as a Speech Communication Situation,* Tampere: Tampere University Press, 2004

[9] A. Rosa, "Features of Oral and Written Communication in Subtitling," *Multimedia Translation,* Y. Gambier and H. Gottlieb (eds.), John Benjamins, Amsterdam/Philadelphia, 2001.

[10] J. Heulwen, "Quality Control of Subtitles: Review or Preview," *Multimedia Translation*. Y. Gambier and H. Gottlieb (eds.), John Benjamins, Amsterdam/Philadelphia, 2001.

[11] J. Pedersen. "Scandinavian Subtitles: A comparative study of subtitling norms in Sweden and Denmark with focus on extralinguistic cultural references," Ph.D. dissertation. Stockholm: University of Stockholm. 2007.

[12] M. Mikhailov and N. Isolahti, "Korpus ustnyx perevodov kak novyj tip korpusa tekstov (The corpus of interpreting as a new type of text corpora, in Russian)," in *Dialog-2008 International Conference*, June 4th–8th, Moscow, 2008, http://www.dialog-21.ru/dialog2008/materials/html/58.htm.

[13] W Lezius, "Morphy - German Morphology, Part-of-Speech Tagging and Applications," in *Proceedings of the 9th EURALEX International Congress* Stuttgart, Germany, 2000, pp. 619-623. Available: http://www.wolfganglezius.de/lib/exe/fetch.php?media=cl:euralex2000.pdf

[14] A. Gelbukh and G. Sidorov, "Approach to construction of automatic morphological analysis systems for inflective languages with little effort. In: Computational Linguistics and Intelligent Text," *Lecture Notes in Computer Science*, N 2588, Springer-Verlag, 2003, pp. 215–220.

Semi-Automatic Parallel Corpora Extraction from Comparable News Corpora

Thoudam Doren Singh and Sivaji Bandyopadhyay

***Abstract*—The parallel corpus is a necessary resource in many multi/cross lingual natural language processing applications that include Machine Translation and Cross Lingual Information Retreival. Preparation of large scale parallel corpus takes time and also demands the linguistics skill. In the present work, a technique has been developed that extracts parallel corpus between Manipuri, a morphologically rich and resource constrained Indian language and English from a comparable news corpora collected from the web. A medium sized Manipuri-English bilingual lexicon and another list of Manipuri-English transliterated entities have been developed and used in the present work. Using morphological information for the agglutinative and inflective Manipuri language, the alignment quality based on similarity measure is further improved. A high level of performance is desirable since errors in sentence alignment cause further errors in systems that use the aligned text. The system has been evaluated and error analysis has also been carried out. The technique shows its effectiveness in Manipuri-English language pair and is extendable to other resource constrained, agglutinative and inflective Indian languages.**

***Index Terms*—arallel corpora, similarity measure, bilingual lexicon, morphology, named entity list.arallel corpora, similarity measure, bilingual lexicon, morphology, named entity list.P**

I. Introduction

IN the last few years , there has been a growing interest in the multilingual corpora. Preparation of large scale parallel corpora is a time consuming process and also demands the linguistics skill though parallel corpora for some of the major languages such as the English-French Canadian Hansards [1] and Europarl parallel corpus[1] [2] involving several European languages are available. There are several languages in the world for which this critical resource is yet to be developed. Sentence level alignment would be trivial if each sentence is translated into exactly one sentence. But generally, a sentence in one language may correspond to multiple sentences in the other; sometimes information content of several sentences is distributed across multiple translated sentences. Thus there are many to many alignments at the sentence level in a parallel corpus. Even in the multilingual and multicultural Indian context, the resource is not available in the required measure for several language pairs. In this view, a simple but effective semi-automatic technique has been devised to develop a parallel corpus between Manipuri, a morphologically rich and resource constrained Indian language and English. One of the major sources of such a resource is the web. The comparable news available between two languages can be collected and parallel corpora can be developed by proper filtering and processing from the raw comparable corpora. The Manipuri and English languages have been considered for case study in the present work.

Manipuri is a scheduled Indian language spoken mainly in the state of Manipur in India and in the neighboring countries namely Bangladesh and Myanmar approximately by three million people. It is a Tibeto-Burman language and highly agglutinative in nature, monosyllabic, influenced and enriched by the Indo-Aryan languages of Sanskrit origin and English. The affixes play the most important role in the structure of the language. A clear-cut demarcation between morphology and syntax is not possible in this language. In Manipuri, words are formed in three processes called affixation, derivation and compounding. The majority of the roots found in the language are bound and the affixes are the determining factor of the class of the words in the language. Annotated corpus, bilingual dictionaries, name dictionaries, WordNet, morphological analyzers, POS taggers, spell checkers etc. are not yet available in Manipuri in the required measure. Recently, manual development of sentence aligned parallel corpora in tourism domain between English and six different Indian languages, namely, Hindi, Bengali, Marathi, Oriya, Urdu and Tamil has been started under the Government of India, Department of Information Technology sponsored consortium project ”Development of English to Indian Languages Machine Translation (EILMT) Systems”. Manual alignment unduly constrains the volume of aligned sentences which can be retrieved given limited time and resource. There is no parallel corpus for many other Indian languages and Manipuri is one of them. In this background, an attempt has been made to extract sentence aligned parallel Manipuri-English corpora from comparable news corpora collected from the web.

The rest of the paper is organized as follows. Related works are discussed in section 2 and collection of comparable news corpora from the web is described in section 3. The preprocessing of the collected comparable news corpora and lexicon preparation are detailed in section 4. The paragraph and sentence level alignment processes are described in section

Manuscript received February 15, 2010. Manuscript accepted for publication May 31, 2010.

Authors are with Computer Science and Engineering Department, Jadavpur University, Kolkata, India (thoudam.doren@gmail.com, sivaji_cse_ju@yahoo.com).

[1] http://www.statmt.org/europarl/

5. The proposed techniques are evaluated in section 6 and the conclusion is drawn in section 7.

II. Related works

There are three kinds of sentence alignment approaches: the lexical approach, the statistical approach and the combinations of them. The performance tends to deteriorate significantly when these approaches are applied to complex corpora that are widely different from the training corpus and/or includes less literal/lexical translation. The major advantage of statistical measures is language independence. The major problem, however, is the proper selection of text units for the consideration. The chosen text units have to be comparable in their semantic complexity; otherwise statistical measures produce incorrect and incomplete results. The string similarity approach aims to extract closely related word pairs. The method is applicable to related language pairs only. Several sentence alignment techniques have been proposed that are mainly based on word correspondence, sentence length, and hybrid approaches. Word correspondence was used by Kay [3] and is based on the idea that words that are translations of each other will have similar distributions in the source (SL) and target language (TL) texts. Sentence length methods are based on the intuition that the length of a translated sentence is likely to be similar to that of the source sentence. Brown, Lai and Mercer [4] used word count as the sentence length, whereas Gale and Church [1] used character count. Brown, Lai and Mercer [4] assumed prior alignment of paragraphs. Gale and Church [1] relied on some previously aligned sentences as ‘anchors’. Word correspondence was further developed in the IBM Model-1 [5] for statistical machine translation. Simard and Plamondon [6] used a composite method in which the first pass aligns at the character level as in [7] (itself based on cognate matching) and the second pass uses IBM Model-1, following Chen [8] . Composite methods are used so that different approaches can complement each other. The Gale and Church [1] algorithm is similar to the Brown [4] algorithm except that the former works at the character level while the later works at the word level. Dynamic programming is applied to search for the best alignment. It is assumed that large corpora is already subdivided into smaller chunks. News articles alignment based on Cross Lingual Information Retrieval (CLIR) are reported in [9] and [10] . Alignment of Japanese-English articles and sentences is discussed in [11] . Comparison, selection and use of sentence alignment algorithms for new language pairs are discussed in Singh [12] . Bilingual text matching using bilingual dictionary and statistics are discussed in [13] .

III. Collection of comparable news corpora from the web

The Manipuri-English comparable news corpora is collected from news available in both Manipuri and English from the website http://www.thesangaiexpress.com/ covering the period from May 2008 to November 2008 on daily basis since there is no repository maintained in the website. The corpora is comparable in nature as identical news events are discussed in both Manipuri and English news stories but these stories are not aligned either at article or sentence level. The available news covers national and international news, brief news, editorial, letter to editor, articles, sports etc. The local news coverage is more than the national and international news. The Manipuri side of the news is available in PDF format and the English side of the news is available in ASCII plain text format. A technique has been developed to convert contents from PDF documents to Unicode format. There are 15-20 common articles in each day in both the languages even though these articles are not the exact translations of each other . So, identification of the comparable articles is done from the publication of each day . From this collection, 23375 English and 22743 Manipuri sentences respectively are available in the comparable news corpus. The length of Manipuri sentences range from 10-30 words and the average length is 22.5 words per sentence. The individual articles with multiple paragraphs are reduced to single paragraphs. Use of abbreviation is very common and presence of such a list of abbreviations is necessary to improve the alignement score. The corpus cleaning process removes undesirable parts from texts such as headlines, place of news, date etc.

IV. Preprocessing of Manipuri sentences and lexicon preparation

A. Conversion from PDF to Unicode Format

The Manipuri side of the news is available in PDF format. A tool has been developed to convert Manipuri news PDF articles to Bengali Unicode[2]. The Bengali Unicode characters are used to represent Manipuri as well. The conversion of PDF format into Unicode involves the conversion to ASCII and then into Unicode using mapping tables between the ASCII characters and corresponding Bengali Unicode. The mapping tables have been prepared at different levels with separate tables for single characters and conjuncts with two or more than two characters. The single character mapping table contains 72 entries and the conjunct characters mapping table consists of 795 entries. There are conjuncts of 2, 3 and 4 characters. Sub-tables for each of the conjuncts are prepared. The preparation of such mapping table for different combination of 2,3 and 4 characters is a repetitive and time consuming process. The corpus is searched to find conjuncts with maximum number of characters (i.e., four) from the ASCII version of Manipuri file and if not found the process is repeated for conjuncts with lesser number of characters and so on. Once match is found the corresponding unicode characters are copied from the mapping table and the process is repeated for the remaining characters. English words are sometimes present in the Manipuri side of the news and these are filtered out to avoid unknown character features during the similarity-based alignment using bilingual

[2] http://unicode.org/charts/PDF/U0980.pdf

lexicon. The unknown characters are filtered and spellings are corrected manually.

B. Preparation of bilingual lexicon and parallel named entities list

The Manipuri-English lexicon [14] is being digitized and currently contains 9618 Manipuri words and the corresponding English words. Use of transliterated English words is very prominent in Manipuri. A list of 2611 Manipuri words and their English transliterations has been developed from the news corpus to improve the alignment quality. Names of people, places, and other entities often do not appear in the bilingual lexicon. The named entities which include person name, name of place and name of organisation are identified from the text based on the work of Named Entity Recognition (NER) for Manipuri using Support Vector Machine (SVM) machine learning technique [15] and transliterated using the Modified Joint Source Channel Model for Transliteration [16] . A total number of 58291 named entites have been identified in the Manipuri news side of 22743 sentences which is accountable for 11.39 % of the total words. Thus, the identification of the named entities is important and is playing a vital role in sentence aligned parallel corpora extraction for news domain.

1) Manipuri Named Entity Recognition: A part of the Manipuri news corpus of 28,629 wordforms has been manually annotated as training data with the major named entity (NE) tags, namely person name, location name, organization name and miscellaneous name to apply Support Vector Machine (SVM) based machine learning technique. Miscellaneous name includes the festival name, name of objects, name of building, date, time, measurement expression and percentage expression etc. The SVM based system makes use of the different contextual information of the words along with the variety of word-level orthographic features that are helpful in predicting the NE classes.

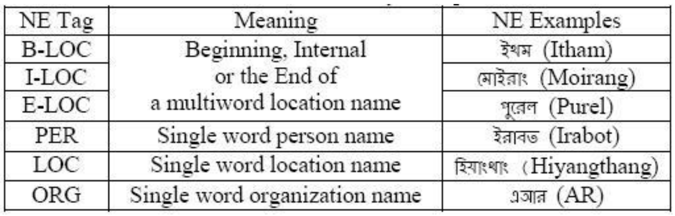
NE identification in Indian languages as well as in Manipuri is difficult and challenging as:

– Unlike English and most of the European languages, Manipuri lacks capitalization information, which plays a very important role in identifying NEs.
– A lot of NEs in Manipuri can appear in the dictionary with some other specific meanings.
– Manipuri is a highly inflectional language providing one of the richest and most challenging sets of linguistic and statistical features resulting in long and complex wordforms.
– Manipuri is a relatively free word order language. Thus NEs can appear in subject and object positions making the NER task more difficult compared to other languages.

The Manipuri NE tagging system includes two main phases: training and classification. The training process has been carried out by YamCha[3] toolkit, an SVM based tool for detecting classes in documents and formulating the NE tagging task as a sequence labeling problem. For classification, the TinySVM-0.07[4] classifier has been used that seems to be the best optimized among publicly available SVM toolkits.

In the present work, the NE tagset used have been further subdivided into the detailed categories in order to denote the boundaries of NEs properly. Table I shows the examples.

TABLE I
NAMED ENTITY TAGSET.



NE Tag	Meaning	NE Examples
B-LOC	Beginning, Internal or the End of a multiword location name	ইথম (Itham)
I-LOC		মোইরাং (Moirang)
E-LOC		পুরেল (Purel)
PER	Single word person name	ইরাবত (Irabot)
LOC	Single word location name	হিয়াংথাং (Hiyangthang)
ORG	Single word organization name	এআর (AR)

The best feature set (F) of Manipuri NER is identified as F=[prefixes and suffixes of length upto three characters of the current word, dynamic NE tags of the previous two words, POS tags of the previous two and next two words, digit information, length of the word].

2) Manipuri-English Transliteration: A transliteration system takes as input a character string in the source language and generates a character string in the target language as output. The process can be conceptualized as two levels of decoding: segmentation of the source string into transliteration units; and relating the source language transliteration units with units in the target language, by resolving different combinations of alignments and unit mappings. The problem of machine transliteration has been studied extensively in the paradigm of the noisy channel model. Translation of named entities is a tricky task: it involves both translation and transliteration. For example, the organization name Jadavpur viswavidyalaya is translated to Jadavpur University in which Jadavpur is transliterated to Jadavpur and viswavidyalaya is translated to University. Manipuri-English transliteration is based on Modified Joint Source Channel Model for transliteration [16].

A medium sized bilingual training corpus has been developed that contains entries mapping Manipuri names to their respective English transliterations. Transliteration units (TUs) are extracted from the Manipuri and the corresponding English names, and Manipuri TUs are associated with their English counterparts along with the TUs in context.

V. PARAGRAPH AND SENTENCE ALIGNMENT PROCESS

A. Paragraph Alignment

The Manipuri-English sentence alignment system is a two-step process. The schematic diagram of the sentence alignment process from the comparable news corpora is shown in Table I. As an initial step, the relevant articles of both sides are sorted out manually. The quantity and quality of

[3] http://chasen-org/ taku/software/yamcha/

[4] http://cl.aist-nara.ac.jp/ taku-ku/software/TinySVM

the output will decrease if less structured texts are used even if a large set of translation equivalents is used in the initial step. For highly structured texts like technical documentation, this method provides fast and precise results. An advantage is that any dictionary may be used by the algorithm as long as it suits the domain of the corpus. There were situations of many-to-one and one-to-many paragraphs between Manipuri and English articles and all the paragraphs in each articles are manually merged in one. The paragraphs are manually aligned since the boundaries are clearly marked.

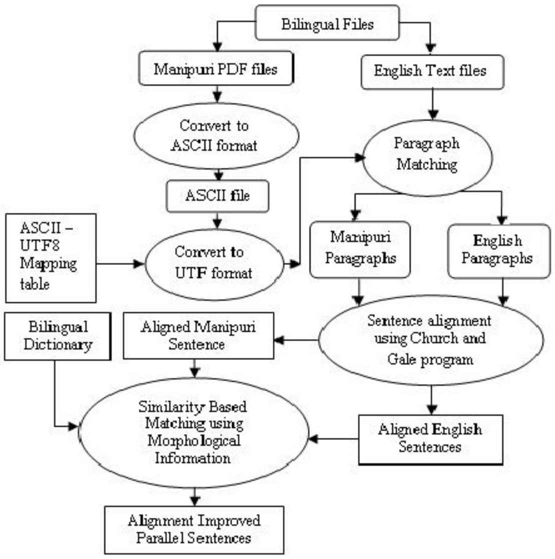

Bilingual Files
Manipuri PDF files
English Text files
Convert to ASCII format
Paragraph Matching
ASCII file
ASCII – UTF8 Mapping table
Convert to UTF format
Manipuri Paragraphs
English Paragraphs
Bilingual Dictionary
Aligned Manipuri Sentence
Sentence alignment using Church and Gale program
Similarity Based Matching using Morphological Information
Aligned English Sentences
Alignment Improved Parallel Sentences

Fig. 1. Schematic diagram of sentence aligned parallel corpora extraction

After the paragraphs are aligned, the sentences are aligned using the sentence alignment program of Gale and Church [1]. However the alignment achieved at this stage is not usable mainly because the sentence alignment program is based on a simple statistical model of character lengths. It is observed that the alignment quality using this approach is poor between a highly agglutinative Indian language like Manipuri and not so agglutinative language like English.

B. Gale and Church Sentence Alignment Method

The Gale and Church program [1] uses the fact that longer sentences in one language tend to be translated into longer sentences in the other language, and that shorter sentences tend to be translated into shorter sentences. A probabilistic score is assigned to each proposed correspondence of sentences, based on the scaled difference of lengths of the two sentences (in characters) and the variance of this difference. This probabilistic score is used in a dynamic programming framework to find the maximum likelihood.

In the following, the distance function $d(x_1, y_1; x_2, y_2)$, is defined in a general way to allow insertion, deletion, substitution, etc. x and y are sequences of objects, represented as non-zero integers to be aligned. Thus let
1. $d(x_1, y_1; 0, 0)$ be the cost of substituting x_1 with y_1,
2. $d(x_1, 0; 0, 0)$ be the cost of deleting x_1,
3. $d(0, y_1; 0, 0)$ be the cost of insertion of y_1,
4. $d(x_1, y_1; x_2, 0)$ be the cost of contracting x_1 and x_2 to y_1,
5. $d(x_1, y_1; 0, y_2)$ be the cost of expanding x_1 to y_1 and y_2, and
6. $d(x_1, y_1; x_2, y_2)$ be the cost of merging x_1 and x_2 and matching with y_1 and y_2.

The recursive equation used in dynamic programming algorithm is given by equation [1]. Let s_i, $i = 1...I$, be the sentences of one language, and t_j, $j = 1...J$, be the translations of those sentences in the other language. Let d be the distance function, and $D(i, j)$ be the minimum distance between sentences $s_1, ...s_i$ and their translations $t_1, ...t_j$, under the maximum likelihood alignment. $D(i, j)$ is computed by minimizing over six cases (substitution, deletion, insertion, contraction, expansion, and merger) which, in effect, impose a set of slope constraints. That is, $D(i, j)$ is defined by the following recurrence with the initial condition $D(i, j)$ = 0.

$$D(i,j) = \min \begin{cases} D(i, j-1) & +d(0, t_j; 0, 0) \\ D(i-1, j) & +d(s_i, 0; 0, 0) \\ D(i-1, j-1) & +d(s_i, t_j; 0, 0) \\ D(i-1, j-2) & +d(s_i, t_j; 0, t_{j-1}) \\ D(i-2, j-1) & +d(s_i, t_j; s_{i-1}, 0) \\ D(i-2, j-2) & +d(s_i, t_j; s_{i-1}, t_{j-1}) \end{cases} \tag{1}$$

C. Similarity-based approach to sentence alignment

The sentences in the aligned Manipuri and English paragraphs are aligned by a method based on Dynamic Programming (DP) matching. The Manipuri English sentences aligned using the Gale and Church program are realigned using the $align$ tool of Utiyama[5], 1-to-n or n-to-1 (1¡=n¡=6) alignments are taken care while aligning the sentences. In this section, the similarity measure [11] for aligning Manipuri and English sentences are discussed. Let M_i and E_i be the words in the corresponding Manipuri and English sentences for i-th alignment. The similarity between M_i and E_i is defined as in equation [2]:

$$SIM(M_i, E_i) = \frac{co(M_i \times E_i) + 1}{l(M_i) + l(E_i) - 2co(M_i \times E_i) + 1} \tag{2}$$

where

$$l(X) = \sum_{x \in X} f(x)$$

$f(x)$ is the frequency of word x in the sentences.

$$co(M_i \times E_i) = \sum_{(m,e) \in M_i \times E_i} \min(f(m), f(e))$$

$$M_i \times E_i = \{(m, e) | m \in M_i, e \in E_i\}$$

[5] http://mastarpj.nict.go.jp/ mutiyama/software.html#align

and $M_i \times E_i$ is a one-to-one correspondence between Manipuri and English words.

A measure that uses the similarity measures obtained during sentence alignments for paragraph alignment is defined in [11]. Thus $AVSIM(M, E)$ is defined as the similarity between a Manipuri article, M, and corresponding English article, E as given by equation [3].

$$AVSIM(M, E) = \frac{\sum_{k=1}^{m} (SIM(M_k, E_k))}{m} \quad (3)$$

where $(M_1, E_1), (M_2, E_2), \ldots (M_m, E_m)$ are the sentence alignments obtained by the method described in equation [2]. The sentence alignment measures in a correctly aligned article pair should have more similarity than the ones in an incorrectly aligned article pair. Consequently, article alignments with high $AVSIM$ are likely to be correct. The sentence alignment program aligns sentences accurately if the English sentences are literal translations of the Manipuri. However, the relation between English and Manipuri news sentences are not literal translations. Thus, the results for sentence alignments include many incorrect alignments. The sentence level similarity measure is defined in [11] as given by equation [4]

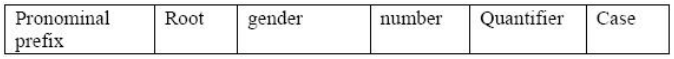
$$SntScore(Mi, Ei) = \frac{AVSIM(M, E)}{SIM(Mi, Ei)} \quad (4)$$

where $SntScore(Mi, Ei)$ is the similarity in the i-th alignment, (Mi, Ei), in the aligned articles M and E. When the correctness of two sentence alignments in the same article alignment is compared, the rank order of sentence alignments obtained by applying $SntScore$ is the same as that of SIM because they share a common $AVSIM$. However, when the correctness of two sentence alignments in different article alignments is compared, $SntScore$ prefers the sentence alignment with the more similar (high $AVSIM$) article alignment even if their SIM has the same value, while SIM cannot discriminate between the correctness of two sentence alignments if their SIM has the same value. Therefore, $SntScore$ is more appropriate than SIM if we want to compare sentence alignments in different article alignments, because, in general, an aligned sentence in a good article alignment is more reliable.

D. *Incorporate morphological information*

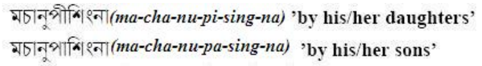
In order to improve the alignment quality between Manipuri and English, an affix adaptation module has been developed which uses the bilingual dictionary. There is no direct equivalence of the Manipuri case markers in English. So, establishing a word level similarity between Manipuri and English is more tedious if not impossible. Essentially, all morphological forms of a word and its translations have to exist in the bilingual lexicon, and every word has to appear with every possible case marker, which will require an impossibly huge amount of lexicon. In order to find the similarity between Manipuri and English based on the bilingual lexicon, the sentences of the Manipuri side are passed through the affix adaptation module and English side is searched for a corresponding match. By doing this, the number of matching words is increased thereby improving the similarity measures. The data sparseness problem can be reduced by applying similar techniques for other agglutinative and inflective languages. The affix adaptation module is developed based on the works on Manipuri Morphological analyzer [17] , Manipuri word classes and sentence type identification [18] , Morphology driven Manipuri POS tagger [19] and Manipuri-English MT system [20] . It is often observed that the number of mapping from a single Manipuri word to multiple English word is more. Whenever a dictionary is being compiled, spelling variants hamper the search for agreement between words, limiting the number of possible examples. Thus, making the right choice of English word for a Manipuri word is cumbersome.

1) Manipuri Morphology: There are free and bound roots in Manipuri. All the verb roots are bound roots. There are also a few bound noun roots, the interrogative and demonstrative pronoun roots. They cannot occur without some particle prefixed or suffixed to it. The bound root may form a compound by the addition of another root. The free roots are pure nouns, pronouns, time adverbials and some numerals. The bound roots are mostly verb roots although there are a few noun and other roots. The suffixes, which are attached to the nouns, derived nouns, to the adjectives in noun phrases including numerals, the case markers and the bound coordinators are the nominal suffixes. In Manipuri, the nominal suffixes are always attached to the numeral in a noun phrase and the noun cannot take the suffixes. Since numerals are considered as adjectives, the position occupied by the numerals in Manipuri may be regarded as adjective positions. There are a few prefixes in Manipuri. These prefixes are mostly attached to the verb roots. They can also be attached to the derived nouns and bound noun roots. There are also a few prefixes derived from the personal pronouns.



Pronominal prefix	Root	gender	number	Quantifier	Case

Fig. 2. Noun morphology



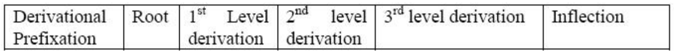
মচানুপীশিংনা(*ma-cha-nu-pi-sing-na*) **’by his/her daughters’**
মচানুপাশিংনা(*ma-cha-nu-pa-sing-na*) **’by his/her sons’**

Fig. 3. Noun morphology example

The $-ma$ ”his/her” is the pronominal suffix and $-cha$ ”child” is the noun root. The $-nu$ ”human” is suffixed by $-pi$ to indicate a female human and $-pa$ to indicate a male human. The $-sing$ or $-khoy$ or $yaam$ can be used to indicate plurality. $-sing$ cannot be used with pronouns or proper nouns and $-khoy$ cannot be used with nonhuman nouns. $-na$ meaning ”by the” is the instrumental case marker.

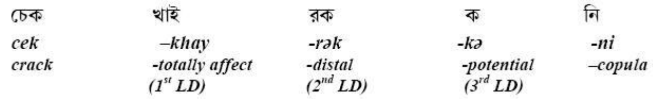
In Manipuri language, the number of verbal suffixes is more than that of the nominal suffixes. New words are easily formed in Manipuri using morphological rules. Inflectional morphology is more productive than derivative morphology. There are 8 inflectional (INFL) suffixes and 23 enclitics (ENC). There are 5 derivational prefixes out of which 2 are category changing and 3 are non-category changing. There are 31 non-category changing derivational suffixes and 2 category changing suffixes. The non-category changing derivational suffixes may be divided into first level derivatives (1st LD) of 8 suffixes, second level derivatives (2nd LD) of 16 suffixes and third level derivatives (3rd LD) of 7 suffixes. Enclitics in Manipuri fall in six categories: determiners, case markers, the copula, mood markers, inclusive/exclusive and pragmatic peak markers and attitude markers. The categories are determined on the basis of position in the word (category 1 occurs before category 2, category 2 occurs before category 3 and so on). The verb morphology is more complex than the noun. Figure 2 gives the noun morphology and its example is given by Figure 3.

Figure 4 gives the verb morphology and the example is given by Figure 5.



Derivational Prefixation	Root	1st Level derivation	2nd level derivation	3rd level derivation	Inflection

Fig. 4. Verb morphology



চেক	খাই	রক	ক	নি
cek	*–khay*	*-rək*	*-kə*	*-ni*
crack	*-totally affect (1st LD)*	*-distal (2nd LD)*	*-potential (3rd LD)*	*–copula*

Fig. 5. Verb morphology example

VI. Evaluation

Methods and practical issues in evaluating alignment techniques are discussed in Langlais [21] . In the experiments in the present work, different cases considering different sizes of corpus, effect of noise of the source and target language other than AVSIM score are considered as mentioned below:

- Same size without noise
- Same size with noise
- Different size with noise

Test data of 5000 Manipuri-English parallel sentences have been manually prepared on news domain. The noise is the unrelated data from other corpus and 10 percent of the corpus size is added as the noise. The number of pairs in a one-to-n alignment is n. The evaluation parameters Recall (R), Precision (P) and F-score (F) are defined by equations [5], [6] and [7] respectively.

$$R = \frac{\#\, of\, correctly\, aligned\, sentence\, pairs}{total\, \#\, of\, sentence\, pairs\, aligned\, in\, corpus} \quad (5)$$

$$P = \frac{\#\, of\, correctly\, aligned\, sentence\, pairs}{total\, \#\, of\, aligned\, sentence\, pairs\, proposed\, by\, program} \quad (6)$$

$$F = \frac{2 \times (Precision \times Recall)}{(Precision + Recall)} \quad (7)$$

TABLE II
Same Corpus Size using [Bilingual Dictionary].

	500 sentences	1000 sentences	2000 sentences	5000 sentences
Precision	86.0	85.9	84.6	83.3
Recall	86.8	86.1	85.8	85.5
F-Score	86.3	85.9	85.1	84.3

The Table II gives the baseline result of the system in terms of precision, recall and F-score using equal number of source and target sentences (i.e., same corpus size). The system uses only Manipuri English bilingual dictionary.

TABLE III
Same Corpus Size using [Bilingual Dictionary + Transliterated Words].

	500 sentences	1000 sentences	2000 sentences	5000 sentences
Precision	97.0	96.9	95.6	93.3
Recall	96.8	97.1	95.8	93.5
F-Score	97.0	96.8	95.6	93.3

The Table III gives the result of the system using the transliterated entities in addition to the Manipuri English bilingual dictionary in terms of precision, recall and F-Score with equal number of source and target sentences (i.e., same corpus size). It is observed that there is a slight decline in the performance of the system as the corpus size increase.

TABLE IV
Same Corpus Size using [Bilingual Dictionary + Transliterated Words + Morphological Information].

	500 sentences	1000 sentences	2000 sentences	5000 sentences
Precision	98.9	98.8	98.3	95.3
Recall	97.4	96.6	96.3	94.2
F-Score	98.1	97.6	97.2	94.7

The Table IV gives the result of the system by integrating the morphological information along with the Manipuri-English bilingual dictionary and the list of transliterated Manipuri-English entities. It is observed that the system outperforms the baseline system even with increase in the corpus size. There is equal number of source and target sentences (i.e., same corpus size). The system is evaluated by putting 10 percent unrelated English sentences from other source as noise. The result of this experiment is given in

Table V. It is observed that when noise is introduced, the system performance decreases slightly.

TABLE V
Noisy Corpus using [Bilingual Dictionary + Transliterated Words + Morphological Information].

	500 sentences	1000 sentences	2000 sentences	5000 sentences
Precision	95.9	94.5	93.5	92.7
Recall	94.2	93.9	93.2	92.1
F-Score	95.0	94.1	93.3	92.3

VII. Conclusion

The most important category for sentence alignment is one-to-one. The other alignments such as 1-to-n, n-to-1, n-to-m for $2 \leq n < 6$ and $2 \leq m < 6$ are discarded. The introduction of morphological information has further improved the alignment both for the same and different size corpus. The proposed system is evaluated considering the size, noise, transliterated entities and morphological information. The important alignment is the one-to-one with higher AVSIM score. They are shortlisted and checked for better alignment quality setting a threshold. The improvement over the baseline system after the introduction of the morphological information is observed overcoming the data sparseness on both the cases of clean as well as noisy test data. 10,350 parallel sentences have been collected in the first phase and it is planned that more parallel sentences will be collected using the technique in future. The performance of the system can be further improved by increasing the size of the bilingual dictionary including the transliterated list of named entities. The system gives a better output for highly agglutinative languages with constrained resources and can be extended to other Indian languages. This is the first attempt to extract parallel corpus of Manipuri and English from the web. The sentence aligned parallel corpora developed using this technique has been used in a Manipuri-English Statistical Machine Translation System.

References

[1] W. A. Gale and K. W. Church, “A program for aligning sentences in bilingual corpora,” in *Proceedings of 29th Annual Meeting of the Association for Computational Linguistics*, Berkeley, California, 1991, pp. 177–184.

[2] P. Koehn, “A parallel corpus for statistical machine translation,” in *In MT Summit X*, 2005.

[3] M. Kay and M. Roscheisen, “Text translation alignment,” in *Computational Linguistics*, 1993, pp. 121–142.

[4] P. F. Brown, J. C. Lai, and R. L. Mercer, “Aligning sentences in parallel corpora,” in *Proceedings of 29th Annual Meeting of the Association for Computational Linguistics*, Berkeley, California, 1991, pp. 169–176.

[5] P. F. Brown, S. A. D. Pietra, V. J. D. Pietra, and R. L. Mercer, “Mathematics of statistical machine translation: Parameter estimation,” in *Computational Linguistics*, 1993, pp. 163–311.

[6] M. Simard and P. Plamondon, “Bilingual sentence alignment: Balancing robustness and accuracy,” in *Machine Translation, 13(1)*, 1998, pp. 59–80.

[7] K. W. Church, “Char align: A program for aligning parallel texts at the character level,” in *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, Columbus, Ohio, 1993, pp. 1–8.

[8] S. F. Chen, “Aligning sentences in bilingual corpora using lexical information,” in *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, Columbus, Ohio, 1993, pp. 9–16.

[9] N. Collier, H. Hirakawa, and A. Kumano, “Machine translation vs. dictionary term translation - a comparison for english-japanese news article alignment,” in *In COLING-ACL 98*, 1998, pp. 263–267.

[10] K. Matsumoto and H. Tanaka, “Automatic alignment of japanese and english newspaper articles using an mt system and a bilingual company name dictionary,” in *In LREC-2002*, 2002, pp. 480–484.

[11] M. Utiyama and H. Isahara, “Reliable measures for aligning japanese-english news articles and sentences,” in *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, Sapporo, Japan, 2003, pp. 72–79.

[12] A. K. Singh and S. Husain, “Comparison, selection and use of sentence alignment algorithms for new language pairs,” in *Proceedings of the ACL-05: Association for Computational Linguistics Workshop*, Ann Arbor, USA, 2005, pp. 177–184.

[13] T. Utsuro, H. Ikeda, M. Yamane, Y. Matsumoto, and M. Nagao, “Bilingual text matching using bilingual dictionary and statistics,” in *In COLING’ 94*, 1994, pp. 1076–1082.

[14] S. I. Singh, “Manipuri to english dictionary.” Imphal, India: S. Ibetombi Devi, 2004.

[15] T. D. Singh, N. Kishorjit, A. Ekbal, and S. Bandyopadhyay, “Named entity recognition for manipuri using support vector machine,” in *In Proceedings of PACLIC 23*, Hong Kong, 2009, pp. 811–818.

[16] A. Ekbal, S. K. Naskar, and S. Bandyopadhyay, “A modified joint source-channel model for transliteration,” in *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*. Sydney: Association for Computational Linguistics, 2006, pp. 191–198.

[17] T. D. Singh and S. Bandyopadhyay, “Manipuri morphological analyzer,” in *In the Proceedings of the Platinum Jubilee International Conference of LSI*, Hyderabad, India, 2005.

[18] ——, “Word class and sentence type identification in manipuri morphological analyzer,” in *In Proceedings of MSPIL*, Mumbai, India, 2006, pp. 11–17.

[19] ——, “Morphology driven manipuri pos tagger,” in *In Proceedings of IJCNLP-08 Workshop on NLP for Less Privileged Languages*, Hyderabad, India, 2008, pp. 91–98.

[20] ——, “Manipuri-english example based machine translation system,” in *International Journal of Computational Linguistics and Applications (IJCLA), ISSN 0976-0962*. Delhi, India: Bahri Publication, 2010, pp. 147–158.

[21] P. Langlais, M. Simard, and J. Veronis, “Methods and practical issues in evaluating alignment techniques,” in *Proceedings of 16th International Conference on Computational Linguistics (COLING-96)*, 1996.

A Natural Language Dialogue System for Impression-based Music Retrieval

Tadahiko Kumamoto

Abstract—**Impression-based music retrieval is the best way to find pieces of music that suit the preferences, senses, or mental states of users. A natural language interface (NLI) is more useful and effective than a graphical user interface for impression-based music retrieval since an NLI interprets users' spontaneous input sentences to represent musical impressions and generates query vectors for music retrieval. Existing impression-based music retrieval systems, however, have no dialogue capabilities for modifying the most recently used query vector. We evaluated a natural language dialogue system we developed that deals not only with 164 impression words but also with 14 comparative expressions, such as "a little more" and "more and more," and, if necessary, modifies the most recently used query vector through a dialogue. We also evaluated performance using 35 participants to determine the effectiveness of our dialogue system.**

Index Terms—**Music retrieval, impression-based, natural langauge dialogue.**

I. Introduction

WHEN users want to locate specific music data from the huge volumes contained in music databases, they usually input bibliographic keywords, such as title and artist. When they do not have any bibliographic keywords, they can use content-based music-retrieval systems that enable them to find data by singing the song, typing parts of the lyrics, or humming the tune [1], [2], [3], [4]. However, these systems are ineffective if they do not specify the exact music data they want to find. In such situations, impression-based music-retrieval systems are best because they enable users to find pieces of music that suit their preferences, senses, or mental states [5], [6], [7], [8], [9].

Information for impression-based music-retrieval systems is generally entered using one of the following three methods: (i) users select one or more impression words from the multiple words presented [5], [8], (ii) users select one or more impression words from the multiple words presented and evaluate each of the selected words using a Likert scale [6], and (iii) users select one or more pairs of impression words from the multiple pairs presented and evaluate each of the selected pairs using a Likert scale [7], [9]. With these approaches, increasing the numbers of words presented increases the cost to the user in terms of time and the labor required to input impressions. A set of words that is too limited, on the other hand, will often not allow users to accurately represent their target impressions. A natural language interface (NLI), therefore, is needed so that users can input impressions without consciously limiting their vocabulary and wording.

Few NLIs have so far been developed as a user interface for an impression-based multimedia content retrieval system. For example, we have developed an NLI that interprets 164 impression words, such as "happy" and "sad," and 119 degree modifiers, such as "a little" and "comparatively," and generates a query vector for music retrieval [10], [11]. However, our NLI does not have dialogue capabilities for modifying the most recently used query vector through a dialogue. That is, when a user's impression from a listening to a retrieval result is not similar to or does not match the user's inputted impressions, the user is asked to enter a new sentence to obtain more accurate retrieval results rather than modify the most recent query vector. Harada et al. have developed an impression-based image-retrieval system with an NLI that deals with 40 impression words and a few comparative expressions [12]. However, their NLI only understands a few stereotypical expressions, such as "I would like to have a simpler one," and does not have sufficient dialogue capabilities for interactive impression-based retrieval.

In this paper, we, therefore, describe a natural language dialogue system we developed that helps users to generate or modify query vectors for impression-based music-retrieval in Japanese. We modified our impression-based music-retrieval system to include an NLI [9], [10], [11] as a base module for our dialogue system since our NLI deals with many impression words but has no dialogue capabilities for interactive retrieval. That is, we developed a method for dealing with 14 comparative expressions, and then incorporated it into our NLI. This enables users to obtain more accurate retrieval results for impression-based music retrieval since users can easily modify the most recently used query vectors through a dialogue with our dialogue system. Note that such comparatives as "brighter" and "happier" are not used in the target language of Japanese, while adjectives modified by such comparative expressions as "more," "a little more," and "still more" are used.

The remainder of this paper is organized as follows. Sect. II describes the specifications of the query vectors that are valid in our impression-based music-retrieval system. Sect. III proposes a method for interpreting users' spontaneous input sentences and, if necessary, modifying the most recently

Manuscript received November 4, 2009. Manuscript accepted for publication May 31, 2010.

The autor is with National Institute of Information and Communications Technology, Hikari-dai 3-5, Kansai Science City, Kyoto 619-0289, Japan (email: kuma@nict.go.jp).

TABLE I
TEN IMPRESSION SCALES FOR REPRESENTING MUSICAL IMPRESSIONS.

N	Impression word pair	N	Impression word pair
1	Quiet — Noisy	6	Leisurely — Restricted
2	Calm — Agitated	7	Pretty — Unattractive
3	Refreshing — Depressing	8	Happy — Sad
4	Bright — Dark	9	Relax — Arouse
5	Solemn — Flippant	10	The mind is restored — The mind is vulnerable

used query vectors. Sect. IV shows the process flow of our dialogue system into which the proposed method was incorporated. Sect. V contains the details and the results of a performance-evaluation experiment with 35 participants. Finally, Sect. VI concludes this paper and describes the future works to be undertaken.

II. REQUIRED QUERY VECTOR SPECIFICATIONS

This section contains a description of the specifications for query vectors that are valid in the impression-based music-retrieval system into which our proposed method is incorporated.

A query vector has ten components. Each component corresponds sequentially to each of the ten impression scales listed in Table I. A component's value is a real number between 0 and 8 corresponding to a seven-step Likert scale, and symbol "nil" denotes a "don't care" term in an impression scale defined using paired impression words. For instance, impression scale No. 8 "Happy — Sad" is characterized by the seven categories including "very happy," "happy," "a little happy," "medium," "a little sad," "sad," and "very sad," that correspond to 7.0, 6.0, 5.0, 4.0, 3.0, 2.0, and 1.0, respectively. That is, to find musical pieces that will create a happy impression, the following 10-dimensional vector would be generated as a query vector.

$$(nil\ nil\ nil\ nil\ nil\ nil\ nil\ 6.0\ nil\ nil)$$

Similarly, to find dark and sad musical pieces, the following query vector would be generated.

$$(nil\ nil\ nil\ 2.0\ nil\ nil\ nil\ 2.0\ nil\ nil)$$

The NLI into which our proposed method of enabling interactive impression-based retrieval is incorporated generates query vectors from 164 impression words using the interpretation rules between impression words and query vectors as listed in Table II. For example, when impression word "gentle" is only extracted from the sentence a user entered, the following query vector will be generated by applying an interpretation rule to the impression word.

$$(5.49\quad 5.79\quad 5.62\quad 5.27\quad nil\quad 5.62\quad 6.01\quad 5.10\quad 5.85\quad 6.16)$$

TABLE II
PORTION OF INTERPRETATION RULES BETWEEN IMPRESSION WORDS AND QUERY VECTORS.

Scale No.	1	2	3	4	5	6	7	8	9	10
pitiful	*nil*	*nil*	2.48	2.13	*nil*	*nil*	*nil*	1.75	*nil*	*nil*
not pitiful	*nil*	*nil*	*nil*	5.41	*nil*	*nil*	*nil*	5.37	*nil*	*nil*
classic	5.42	5.56	*nil*	3.47	5.57	*nil*	5.51	*nil*	5.06	5.09
gentle	5.49	5.79	5.62	5.27	*nil*	5.62	6.01	5.10	5.85	6.16
powerful	2.13	*nil*	*nil*	*nil*	*nil*	*nil*	*nil*	*nil*	2.38	*nil*

TABLE III
PARAMETERS USED TO INTERPRET 14 COMPARATIVE EXPRESSIONS. SOME COMPARATIVE EXPRESSIONS WERE TRANSLATED INTO THE SAME ENGLISH EXPRESSIONS. HOWEVER, THERE IS A SMALL DIFFERENCE BETWEEN THEIR NUANCES IN JAPANESE. A STANDS FOR THE ADJUSTED COEFFICIENT OF DETERMINATION.

Comparative expression	Coef- ficient b	Cons- tant c	A
a little more (mousukoshi)	0.92	0.92	0.998
more; longer; farther; —er (motto)	0.89	1.36	0.998
more; longer; farther; —er (yori)	0.91	1.15	0.996
a little more (mosukoshi)	0.95	0.60	0.995
more and more; increasingly (masumasu)	0.88	1.53	0.994
a little more (mosotto)	0.92	0.87	0.994
a little bit (honno-sukoshi)	0.96	0.44	0.989
much more; all the more (issou)	0.78	2.22	0.959
still more; further (sarani)	0.88	1.54	0.953
still more; further (ichidanto)	0.80	2.28	0.948
far; much (zutto)	0.85	2.26	0.916
far; much (zuutto)	0.40	4.86	0.906
with a jerk; much better (gutto)	0.48	4.26	0.619
with a jerk; much better (gunto)	0.39	4.93	0.535

III. DIALOGUE CAPABILITIES FOR IMPRESSION-BASED RETRIEVAL

This section proposes a method to interpret sentences for modifying the most recently used query vector, which enables interactive impression-based retrieval.

First, we defined comparative expressions that our dialogue system should interpret for interactive impression-based retrieval. We extracted 14 adverbs and adverbial expressions representing the complete degree of change from a Japanese thesaurus called "Ruigo-Kokugo-Jiten" [13] and set them as target comparative expressions in this paper. All target comparative expressions are listed in Table III together with the parameters used to interpret the comparative expressions, where the parameters will appear in the equation (2).

Next, we conducted a questionnaire-based experiment using 50 women and 50 men to investigate how a query vector, $v_i(i = 1, 2, \cdots, 10)$, should be modified on the basis of comparative expressions extracted from a sentence entered to modify the query vector. In this experiment, we assumed that a specified query vector had already been used. The query vectors to be generated in finding "totemo-shizukana (very quiet)," "shizukana (quiet)," "sukoshi-shizukana (a little quiet)," "dochiratomo-ienai (medium)," "sukoshi-shizukadenai (not a little quiet)," "shizukadenai (not quiet)," or "totemo-shizukadenai (not quiet at all)" tunes were assumed and used. Under each

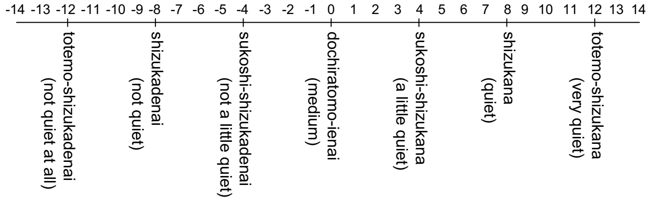

-14 -13 -12 -11 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
totemo-shizukadenai (not quiet at all)
shizukadenai (not quiet)
sukoshi-shizukadenai (not a little quiet)
dochiratomo-ienai (medium)
sukoshi-shizukana (a little quiet)
shizukana (quiet)
totemo-shizukana (very quiet)

Fig. 1. A base scale for scoring. Participants marked their evaluation on the base scale.

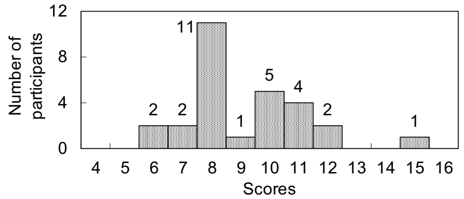


Fig. 2. How much "ichidanto-shizukana (still quieter)" tunes should be to create a quiet impression were marked on the base scale by twenty eight participants under the assumption that "sukoshi-shizukana (a little quiet)" tunes were the most recently retrieved. This histogram shows the results.

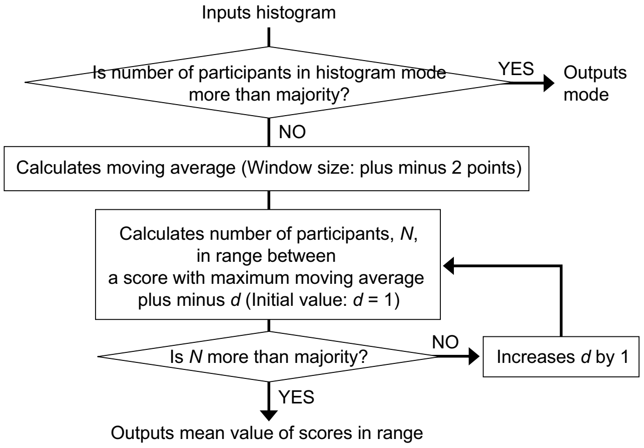

Inputs histogram
Is number of participants in histogram mode more than majority?
YES
Outputs mode
NO
Calculates moving average (Window size: plus minus 2 points)
Calculates number of participants, N, in range between a score with maximum moving average plus minus d (Initial value: d = 1)
Is N more than majority?
NO
Increases d by 1
YES
Outputs mean value of scores in range

Fig. 3. Process flow for determining a representative value from the input histogram.

assumption, about 30 participants evaluated impressions of the phrase containing an impression word and a comparative expression using a base scale for scoring, as shown in Fig. 1. For example, if a query vector for finding a "shizukana (quiet)" tune was just used, the participants were then asked to mark how much "motto-shizukana (quieter)" tunes should be to create a quiet impression on the base scale printed on a sheet of paper.

We found that the estimation results in this experiment varied widely, as shown in Fig. 2. We, therefore, introduced a method to determine the representative value of a histogram. The process flow is outlined in Fig. 3. First, if the number of participants in the mode of an input histogram is more than the majority, the method determines the value of the mode as a representative value of the histogram and terminates the process. Otherwise, the method performs the following processes. (1) Simple moving averages (*MA*) are first computed with the range of five for an input histogram. Then, *MA* is computed using the following equation in this paper.

$$MA_{score} = \frac{1}{5}\sum_{i=-2}^{2} N_{score+i} \tag{1}$$

where N_x denotes the number of participants in the score, x. (2) The number, N, of participants in the range between a score with the maximum moving average plus minus d is computed when the initial value of d is 1. (3) If N is greater than the majority, the mean value computed only from the scores of the participants in the range is determined as a representative value of the histogram, and the process is terminated. Otherwise, d is increased by 1, and the process then proceeds to Step (2).

Next, we used regression analysis to obtain a regression equation or linear equation between the most recently used query vector, w_i, and a query vector modified by a sentence that included a comparative expression, w'_i. Note that w_i equals the value of the i^{th} component of the query vector assumed to have been used most recently, and w'_i equals the representative value subsequently obtained from the corresponding histogram using the method mentioned above. Since this equation was computed based on the 29-step base scale, the equation was converted in terms of the seven-step impression scale. Coefficient b and constant c of equation ($v'_i = bv_i + c$) that were subsequently obtained were then registered in a parameter table for interpreting comparative expressions, which is listed in Table III along with the corresponding adjusted coefficients of determination [14]. Where v_i and v'_i denote comparative expressions converted from w_i and w'_i, respectively. We find that all the comparative expressions excluding "gutto (much better)" and "gunto (much better)" have very high adjusted coefficients of determination, and the adjusted coefficients of determination for "gutto (much better)" and "gunto (much better)" are not so low. Hence, we can say that satisfactory results were obtained in this regression analysis.

We used "shizukana (quiet)" as the impression word modified by a comparative expression. Since this impression word corresponds to 6 points on the seven-step scale, the equation ($v'_i = bv_i + c$) cannot be applied to impression words corresponding to 4 or lower points, such as "sad" and "dark." To solve this problem, we assumed that the difference, $v'_i - v_i$, between the most recently used query vector and a newly generated query vector was in proportion to query vector x_i generated from the impression word modified by a comparative expression. We formulated the following simultaneous equation.

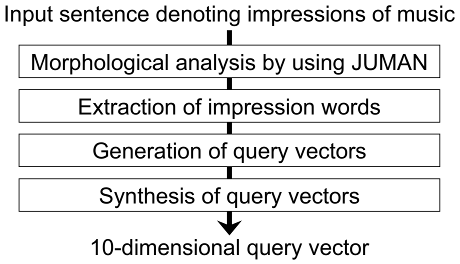

Input sentence denoting impressions of music
Morphological analysis by using JUMAN
Extraction of impression words
Generation of query vectors
Synthesis of query vectors
10-dimensional query vector

Fig. 4. Steps for generating query vectors from input sentences.

$$\begin{aligned} &\text{When } x_i = 6, \quad v'_i = bv_i + c \\ &\text{When } x_i = 4, \quad v'_i = v_i \\ &\text{When } x_i = 2, \quad v'_i = v_i - (v'_i - v_i) \quad = (2-b)v_i - c \end{aligned}$$

By solving this simultaneous equation, we obtained the following equation.

$$v'_i = \frac{(bx_i - x_i - 4b + 6)v_i + c(x_i - 4)}{2}, \tag{2}$$

where x_i denotes the value of the i^{th} component of a query vector generated from the corresponding impression word, and the values for b and c are obtained from the parameter table listed in Table III. In sessions where $x_i = nil$, $v'_i = v_i$ and the value of v'_i is kept. In sessions where $x_i \neq nil$ and $v_i = nil$, the value of v_i is replaced with the value of the i^{th} component from an impression vector [1] representing the first candidate of the most recently retrieved musical pieces.

IV. Dialogue Capabilities for Interactive Impression-based Retrieval

Sentences that users input into the NLI are processed as outlined in the steps shown in Fig. 4, and subsequent query vectors will be generated and used in inputs to retrieve music. Each step is outlined in the following.

A. Morphological Analysis

An input sentence is decomposed into words by using JUMAN [15], which is one of the most famous Japanese morphological analysis systems. The basic form, the part-of-speech name, and the conjugation name for each word are simultaneously annotated for the word as tags.

B. Extraction of Impression Words

Information for query vector generation, such as impression words, comparative expressions, and negative words, is extracted from annotated words obtained in the preceding step. If negative words were extracted, they would become a pair with the depending impression words for each pair, and each pair would be dealt with as one impression word in the following steps. For instance, the impression word “not-pretty” is extracted from the sentence “I don’t want a pretty one.”

C. Generation of Query Vectors

First, when the interpretation rules are applied to the impression words extracted in the preceding step, a query vector is generated from each of the words, as described in Sect. II. When there are results from most recently performed retrieval and a comparative expression is extracted with an impression word from an input sentence, the value of v'_i is obtained by substituting the values of v_i, b, c, and x_i into the equation (2), where v'_i denotes the value of the i^{th} component of the query vector input to the music-retrieval system, v_i denotes the value of the i^{th} component of the most recently used query vector, the values of b and c for the comparative expression are obtained from the parameter table listed in Table III, and x_i denotes the value of the i^{th} component of the query vector generated from the impression word.

D. Synthesis of Query Vectors

If two or more query vectors are generated, the query vectors are synthesized by computing the mean value in each component, where “nil” is excluded from the calculation. For example, when the impression words “happy” and “gentle” are extracted from the input sentence, the following two query vectors are generated from the words.

$$(nil\ nil\ nil\ nil\ nil\ nil\ nil\ 6\ nil\ nil)$$
$$(5.49\ 5.79\ 5.62\ 5.27\ nil\ 5.62\ 6.01\ 5.10\ 5.85\ 6.16)$$

Then, these query vectors are synthesized, and the following query vector is obtained.

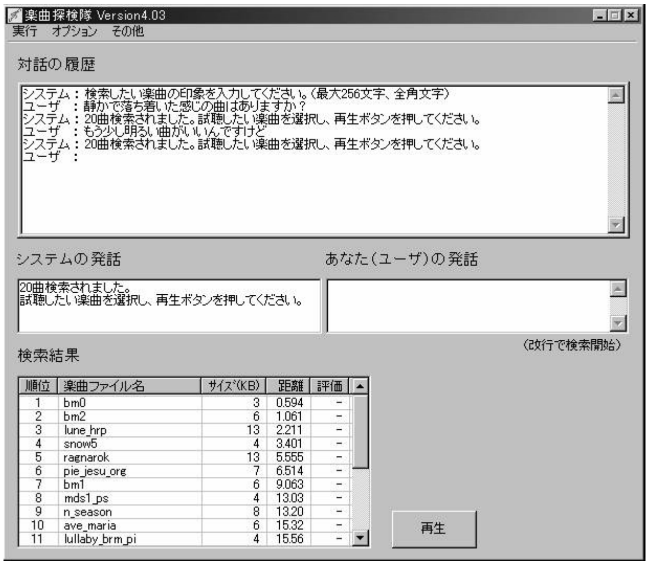
$$(5.49\ 5.79\ 5.62\ 5.27\ nil\ 5.62\ 6.01\ 5.55\ 5.85\ 6.16)$$

V. Performance-Evaluation of Dialogue Capabilities

Nineteen women and sixteen men participated in a performance-evaluation experiment of the proposed dialogue capabilities. We first asked all 35 participants the question, “What impression should the musical pieces you want to obtain create?”. Each of the participants replied to this question with four answers denoting her/his impressions of the musical pieces. We then asked the participants to individually enter their four answers into the impression-based music-retrieval system, which was proposed in the previous sections, and retrieve music [2] using the following procedures. When musical pieces were presented as retrieval results, the participants listened to the first retrieved musical piece with the shortest distance, and then evaluated whether the impressions

[1] A 10-dimensional vector consisting of real values between 0.0 and 8.0 was automatically assigned to every musical piece using a function of the original impression-based music-retrieval system [9] we used in this paper.

[2] The music database had 160 short classical pieces. The playback time ranged from about 30 seconds to several minutes. All the musical pieces have been released to the public through http://nocturne.vis.ne.jp/ and http://k2works.com/nerve/ and have been approved for secondary purposes. A 10-dimensional vector, consisting of real values between 0.0 and 8.0, was automatically assigned to each musical piece using a function of the original music-retrieval system [9] we used in this paper. The original music-retrieval system calculated the distances between a query vector and the vectors of musical pieces, and then presented, at most, the three musical pieces with the shortest distances.



システム：検索したい楽曲の印象を入力してください。（最大256文字、全角文字）
ユーザ　：静かで落ち着いた感じの曲はありますか？
システム：20曲検索されました。試聴したい楽曲を選択し、再生ボタンを押してください。
ユーザ　：もう少し明るい曲がいいんですけど
システム：20曲検索されました。試聴したい楽曲を選択し、再生ボタンを押してください。
ユーザ　：

20曲検索されました。
試聴したい楽曲を選択し、再生ボタンを押してください。

順位	楽曲ファイル名	ｻｲｽﾞ(KB)	距離	評価
1	bm0	3	0.594	–
2	bm2	6	1.061	–
3	lune_hrp	13	2.211	–
4	snow5	4	3.401	–
5	ragnarok	13	5.555	–
6	pie_jesu_org	7	6.514	–
7	bm1	6	9.063	–
8	mds1_ps	4	13.03	–
9	n_season	8	13.20	–
10	ave_maria	6	15.32	–
11	lullaby_brm_pi	4	15.56	–

Fig. 5. A captured snapshot of the user interface for retrieval. The text box in the upper part of the screen displays dialogue history, the text box in the left-middle part displays reports and instructions from the system, the text box in the right-middle part is for the user input, and the text box in the lower displays the results of the most recent retrieval. Rank, names of musical pieces, data size, distance, and so on are also displayed in this text box.

TABLE IV
PERFORMANCE-EVALUATION EXPERIMENT RESULTS.

Query vector	Score or change in score	Number of sessions
New	5 points	36
	4 points	1
	Number of zero hits	2
	Failed (out of target)	31
	Failed (use of unregistered word)	17
Modified	Increased	16
	No change	8
	Decreased	2
	Number of zero hits	13
	Failed (out of target)	9
	Failed (use of unregistered word)	5
Total		140

from this piece were similar to the input impressions using a five-point scale. For example, if the impressions from the first retrieved musical piece were very similar to those input, the participants would award five points. Conversely, if the impressions from the first retrieved musical piece were not at all similar to those input, the participants would award one point. For scores other than five points, the participants were asked to enter a sentence to modify the most recently used query vector and retrieve music again. The 14 comparative expressions, which the participants could enter, were printed on a sheet and presented to the participants, and the participants were asked to use the comparative expressions to form a sentence to modify a query vector. The participants repeated retrieving music until they awarded five points to the first retrieved musical piece or the results of the retrieval became empty. A sample of dialogues provided by our system is shown in Fig. 5, and the results of the experiment mentioned above are listed in Table IV.

Although the participants conducted a total of 140 music retrievals, they awarded five points in 51 (36.4%) of the 140 sessions. A score of five points was awarded for the first retrieval using a new query vector in 36 of the 51 sessions, and the scores were increased to five points by modifying the most recently used query vectors in 15 of the 51 sessions. Note that the remaining session of "Increased" was the case in which, although the score was increased to three points, the session was terminated for some reason.

Next, we analyzed the sessions in which a score of five points was not awarded at all. In Table IV, the "four points" for the new query vectors resulted from the fact that the sentence to modify the most recently used query vector was interpreted as one for generating a new query vector due to a participant's mistype and the session was terminated. The reason why the situation in which the numbers of hits for new query vectors were zero occurred is because the negative form of suffix "sugiru (too)" [3] was not interpreted adequately. In total, 40 (28.6%) of the 140 sessions were classified "Failed (out of target)." This phenomenon occurred when the impression words, which the dialogue system could deal with, were not extracted from the sentences the participants entered. The impression words the dialogue system can deal with are limited to the 164 words that represent the affective characteristics of musical pieces or the change in the listeners' affective states. The input sentence the dialogue system failed to interpret contained critical comments about musical pieces (11 sessions), statements about scenic images (seven sessions), comments about a register of musical instrument (five sessions), and comments about musical structures (four sessions). We did not present the impression words the dialogue system could deal with to the participants to enable them to spontaneously enter sentences. In total, 22 (15.7%) of the 140 sessions were classified "Failed (use of unregistered word)." That is, any query vectors were not generated due to the use of words unregistered in the interpretation rules and sessions were terminated. This indicates the need to develop a method to deal with unregistered words and to incorporate it into the dialogue system. This is one of the areas for our future works. In 2 of the 140 sessions, the scores were decreased through dialogues. One case was when a participant scored low even though the same musical piece was presented as the first retrieval result, and the other was a case when a wrong retrieval result was presented due to wrong interpretation of the sentence indicating changed impressions during the performance of a musical piece. In 8 of the 140 sessions, the scores were not changed through dialogues and were less than five points. These cases occurred since the sessions were terminated for some reasons even though the conditions for

[3] The Japanese word "sugiru" is a suffix and forms a verb by being concatenated with an adjective and can be translated into the English word "too." An example usage is "it is too simple to do this."

finishing a session had not been satisfied. We would like to reconsider the instructions provided to participants.

As mentioned above, when the dialogue system successfully interpreted the user input sentences, the scores for musical pieces presented as the first retrieval results improved. This proves that the dialogue system enables users to obtain better retrieval results through dialogues for impression-based retrieval.

VI. Conclusion

We evaluated a method we proposed to deal with 14 comparative expressions and to enable interactive impression-based retrieval. We incorporated the proposed method into an existing impression-based music-retrieval system that lacked dialogue capabilities but accepted natural language input and conducted a performance-evaluation experiment with 35 participants. Results revealed that the dialogue system was effective in obtaining better retrieval results using dialogues than language input. Note that, although we did not mention in this paper the details of the 119 degree modifiers, such as “a little” and “comparatively” that an original natural language interface can deal with [11] due to page limits, our dialogue system also can deal with the 119 degree modifiers.

Performance evaluation of our dialogue system is the basis for our future work. Other future work includes developing a method to deal with unregistered words and a method to manage changed impressions of a musical piece during performance.

References

[1] Ghias, A., Logan, J., Chamberlin, D., and Smith, B.: Query By Humming – Musical Information Retrieval in an Audio Database. In Proc. of ACM Int. Multimedia Conf., San Francisco, USA (1995)

[2] Blackburn, S. G., and De Roure, D. C.: A Tool for Content based Navigation of Music. In Proc. 6th ACM Int. Multimedia Conf., Bristol, UK (1998) 361–368

[3] Sonoda, T., Goto, M., and Muraoka, Y.: A WWW-based Melody Retrieval System. In Proc. Int. Computer Music Conf., Michigan, USA (1998) 349–352

[4] Kosugi, N., Nagata, H., and Nakanishi, T.: Query-by-Humming on Internet. In Proc. Int. Conf. on Database and Expert Systems Applications (2003) 589–600

[5] Tsuji, Y., Hoshi, M., and Ohmori, T.: Local Pattern of a Melody and Its Applications to Retrieval by Sensitivity Words. Technical Report of IEICE of Japan, Vol. SP96-124 (1997) 17–24

[6] Sato, A., Ogawa, J., and Kitakami, H.: An Impression-based Retrieval System of Music Collection. In Proc. of 4th Int. Conf. on Knowledge-Based Intelligent System and Allied Technologies, Brighton, U.K. (2000) 856–859

[7] Ikezoe, T., Kajikawa, Y., and Nomura, Y.: Music Database Retrieval System with Sensitivity Words Using Music Sensitivity Space. Trans. IPS of Japan, Vol. 42, No. 12 (2001) 3201–3212

[8] Omae, H., Ishibashi, N., Kiyoki, Y., and Anzai, Y.: An Automatic Metadata Creation Method for Music with Multiple Musical Instruments. Technical Report of IPS of Japan, Vol. 2001-DBS-125, No. 84 (2001) 145–152

[9] Kumamoto, T., and Ohta, K.: A Query by Musical Impression System using N-gram Based Features. In Proc. IEEE Conference on Cybernetics and Intelligent Systems (2004) 992–997

[10] Kumamoto, T.: Design and Implementation of Natural Language Interface for Impression-based Music-retrieval Systems. Lecture Notes in Artificial Intelligence, Vol. 3214, Springer-Verlag, Berlin Heidelbelg New York (2004) 139–147

[11] Kumamoto, T., and Ohta, K.: Design and Development of Natural Language Interface for an Impression-based Music Retrieval System. Technical Report of IPS of Japan, Vol. 2003-NL-153, No. 13 (2003) 97–104

[12] Harada, S., Itoh, Y., and Nakatani, H.: Interactive Image Retrieval by Natural Language. Optical Engineering, Vol. 36, No. 12 (1997) 3281–3287

[13] Ohno S., and Hamanishi, M.: Ruigo-Kokugo-Jiten. Kadokawa Shoten Publishing Co.,Ltd., Tokyo, Japan (1985)

[14] Kan, T.: Multivariate Statistical Analysis. Gendai-Sugakusha, Kyoto, Japan (1996)

[15] Kurohashi, S., and Nagao, M.: Manual for Japanese Morphological Analysis System JUMAN Version 3.61. http://pine.kuee.kyoto-u.ac.jp/nl-resource/juman.html (1999)

[16] Taniguchi, T.: Music and Affection. Kitaooji Syobo Press, Kyoto, Japan (1998)

Retrieving Lexical Semantics from Multilingual Corpora

Ahmad R. Shahid and Dimitar Kazakov

***Abstract*—This paper presents a technique to build a lexical resource used for annotation of parallel corpora where the tags can be seen as multilingual ‘synsets’. The approach can be extended to add relationships between these synsets that are akin to WordNet relationships of synonymy and hypernymy. The paper also discusses how the success of this approach can be measured. The reported results are for English, German, French, and Greek using the Europarl parallel corpus.**

***Index Terms*—Multilingual coropora, lexical realtions.**

I. Introduction

THE aim of this work is to build a WordNet-like resource which can be used for Word Sense Disambiguation (WSD) and other such tasks where semantics of words and phrases is the main objective. The multilingual aspect of the approach helps in reducing the ambiguity inherent in any words/phrases in the pivotal language, which is English in the case shown here.

In order to create such a resource we used proceedings from the European Parliament (Europarl)[1]. Four languages were selected with English as the pivotal language in addition to German, French and Greek.

The paragraph-aligned bilingual corpora were fed into a word-alignment tool, GIZA++, to obtain the pair-wise alignments of each language with English. These pair-wise aligned words were later merged into phrases where one word in one language was aligned with more than one word in the other language. Using English as the pivotal language, there were combined into 4-tuples, effectively resulting in a database of multilingual synsets. The synsets were then used to sense disambiguate the individual words and phrases in the original corpora from which they originated. Each of the synsets were latter Part of Speech (POS)-Tagged using the Brill Tagger. The POS tags can help in further removing any ambiguity. Edit distance between any two synsets was also computed in order to use that information for merging any two synsets that are deemed sufficiently close.

II. Related Work

WSD has attracted the attention of the research community for long. It is a tricky issue and needs resources that define the semantic relationships between words. In the last twenty five years various research activities have been undertaken to build large repositories that combined the description of semantic concepts with their relationships. Two efforts worth mentioning here are the Cycorp Cyc project [1] and the lexical semantic database WordNet [2]. Both approaches use a number of predicates to define relationships between concepts, such as “concept A is an instance of concept B” or “concept A is a specific case of concept B.” WordNet also defined the notion of *synsets*, which defines a semantic concept through all relevant synonyms, *e.g.*, {mercury, quicksilver, Hg}.

The original version of the WordNet covered only the English language but the effort has been replicated for other languages as well [3]. Yet all these efforts have been handcrafted, rather than automatically generated and are monolingual in nature. Even though they are highly comprehensive, they require a major, sustained effort to maintain and update.

The work [4] used word alignment in an unsuperised manner to create pseudo-translations which were used for sense tagging of the parallel corpora. They used WordNet as the sense inventory of English. Firstly they aligned each French word with one or more words in English in each sentence. Then to create synsets they looked at the alignment of each French word with all corresponding translations in English in the whole corpus. In order to narrow down the number of combinations they used WordNet to identify nominal compounds, such as $honey\_bee$ and $queen\_bee$. WordNet was also used to manually assign sense tags to words in the subset of the corpus used for evaluation. They found the performance of their approach comparable with other unsupervised approaches.

Interest in the use of parallel corpora for unsupervised WSD has grown recently [5], [6]. In both cases, the use of multilingual synsets is discussed together with various ways of reducing their number.

III. Multilingual Synsets

Multilingual synsets are at the core of this project. Naturally emanating from word alignment in parallel corpora, they make a crucial link between semantics in the original bilingual corpora and the development of a WordNet like resource, rich in semantics and semantic relations between words and phrases.

The concept is simple. A synset, as the name suggests, is a set of synonyms. In the context of this paper, its the aligned

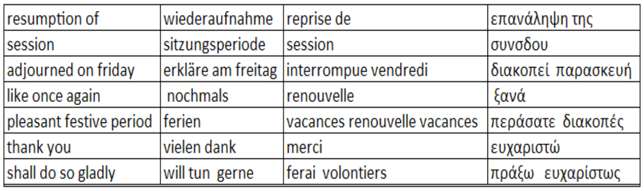
Manuscript received March 24, 2010. Manuscript accepted for publication June 14, 2010.

Authors are with Department of Computer Science, University of York, YO10 5DD, UK (ahmad@cs.york.ac.uk; kazakov@cs.york.ac.uk).

[1] http://www.statmt.org/europarl/

words-phrases in the parallel corpora, put together in the form of 4-tuples.

Figure 1 gives a few examples of the synsets. As can be seen many synsets are phrases rather than words. In the example one synset is comprised of four words “shall do so gladly”.



resumption of	wiederaufnahme	reprise de	επανάληψη της
session	sitzungsperiode	session	συνσδου
adjourned on friday	erkläre am freitag	interrompue vendredi	διακοπεί παρασκευή
like once again	nochmals	renouvelle	ξανά
pleasant festive period	ferien	vacances renouvelle vacances	περάσατε διακοπές
thank you	vielen dank	merci	ευχαριστώ
shall do so gladly	will tun gerne	ferai volontiers	πράξω ευχαρίστως

Fig. 1. Examples of Synsets.

Multilingual synsets help in disambiguating the senses of a word. Translating the English word ‘bank’ with the French ‘banque’ suggests two possible meanings: a financial institution or a collection of a particular kind (e.g., a blood bank), as these words share both meanings, but eliminating the English meaning of a ‘river bank’. Increasing the number of languages could gradually remove all ambiguity, as in the case of {EN: bank, FR: banque, NL: bank}. Insofar these lists of words specify a single semantic concept, they can be seen as WordNet-like synsets that makes use of words of several languages, rather than just one. The greater the number of translations in this multilingual WordNet, the clearer the meaning, yet, one might object, the fewer the number of such polyglots, who could benefit from such translations. However, these multilingual synsets can also be useful in a monolingual context, as unique indices that distinguish the individual meanings of a word.

When annotating parallel corpora with lexical semantics, the multilingual synsets become the sense tags and the parallel corpora are tagged with corresponding tags in a single unsupervised process. The idea is as simple as it is elegant: assuming we have a word-aligned parallel corpus with n languages, annotate each word with a lexical semantic tag consisting of the n-tuple of aligned words. As a result, all occurrences of a given word in the text for language $\mathcal{L}$ are considered as having the same sense, provided they correspond to (are tagged with) the same multilingual synset.

Two great advantages of this scheme are that it is completely unsupervised, and the fact that, unlike manually tagged corpora using WordNet, all words in the corpus are *guaranteed* to have a corresponding multilingual synset.

IV. Synset Generation and WSD

In order to generate the synsets we needed the word-aligned corpora. The Europarl corpus was taken. It was pre-processed, which included among other steps, tokenization of text, lowercasing, removal of empty lines and the removal of XML-tags. After pre-processing a paragraph aligned parallel corpus was obtained. English corpus was used as the pivotal one. All these were fed to GIZA++[2], a standard and freely available tool for word alignment. For alignment, pair-wise corpora were fed into GIZA++ (German with English, French with English, and Greek with English). Thus the output of GIZA++ were pair-wise aligned parallel corpora with markings indicating which words in the target language aligned with which words in English. It might be the case that one word in one language aligns with more than one words in another or it aligns with nothing. Only the aligned words were of any use while generating synsets from the aligned corpora.

For actual synset generation from the aligned corpora we designed our algorithm, which links two or more words in one language together if they align with the same word in another language. The process had to be carried out simultaneously for all the four languages, so as no useful information is lost.

The algorithm links the words of the pivotal language (PL) into phrases and maps all words of the non-pivotal languages to one of these phrases. The array a[1..N] serves to store in the field a[i] the number of the phrase to which word i in the pivotal language belongs. Initially, all PL words are assumed to belong to different phrases (i.e., they form a phrase on their own). Two or more PL words $a[j], ...a[j+k]$ are placed in the same group if there is a word in another language, which is aligned with all of them. This information is stored by assigning the same phrase number to $a[j], ..., a[j+k]$. The array t is used to store information about the word alignment between each non-PL and the PL. The assignment t[l,i]:= k represents the fact that the i-th word in non-PL l was aligned with the k-th word in the PL.

Subsequently, each synset is spelt out by producing a phrase in the pivotal language (consisting of one or more PL words with the same phrase number) and extracting for each non-PL language all the words that point to a PL word in that group: this final step is straightforward, and due to space limitations is not shown in Figure 2.

While performing the task of synset generation WSD of the original corpus in English was done automatically. That was achieved because the start of each separate phrase in English is numbered with the index number of the first word in that phrase in the whole original corpus. Thus the phrase “shall do so gladly” (reference Fig. 1) is assigned the number 41, which is the index of the word *pleasant* in the whole original English corpus. Thus the start of each phrase in the English corpus has been assigned a sense tag (the 4-tuple synset) and it constitutes the WSD part of the process.

Part of Speech (POS) is an extra bit of useful information that can be used for WSD [7], [8]. POS tags of the neighbors of the target word help in narrowing down the meanings of the word. We used Brill Tagger [9] to assign POS tags to individual words in the English phrases in the synsets.

The approach described here produces a large number of what we would call ’proto-synsets’—for a corpus of more than 1.8 million words, there are more than 1.5 million different such synsets. Their number can be reduced and their composition—brought closer to what one would expect to see in a hand-crafted dictionary in the following two ways:

[2] http://fjoch.com/GIZA++.html

```
Data Structures:

int N % number of words in the PL
int M % number of non-PLs
int array a[1..N] int array t[1..N,1..M]

Initialize:

for i=1 to N do a[i] := i

Form phrases:

for l=1 to M
|  L := number of words in lang.l
|  for i=1 to L
|  | if word i in lang. l is aligned
|  |     with word j in the PL
|  | then t[l,i] := j
|  |  elseif word i in lang.l is aligned
|  | with words j,j+1,j+k in the PL
|  |    then
|  |      t[l,i] :=j
|  | for z=1 to k do
|__|______|_ a[j+z] := a[j]
```

Fig. 2. Synset Generation Algorithm.

firstly, through the identification and merger of proto-synsets only varying in word forms corresponding to the same lexical entry (e.g., flight-X-Y-Z, flights-X-Y-Z); secondly, through the merger of proto-synsets in which the differences are limited to words that are synonyms in the given language (e.g., car-auto-*automobile* vs car-auto-*voiture*). These two approaches are addressed in the following two sections.

V. Edit Distances

We need to merge the redundant synsets, based on their syntax and semantics, since morphemes could be both inflectional and derivational. In inflectional morphemes the meaning is not changed. Hence both *dog* and *dogs* have the same meaning and *dogs* is an inflection of *dog*. In derivational morphemes, however, the meaning might change. Thus *unhappy* is derived from *happy*, yet they are antonyms of each other.

Both inflectional and derivational morphemes need to be taken care of and corresponding synsets merged in order to reduce the number of synsets and making the resource more concise and useful. For inflectional morphology we used the edit distance, for derivational we intend to use synonymy detection, which is discussed in the next section.

Edit distance measures the minimum number of edit steps required to convert one string into another [10], [11], [12]. The only three operations allowed are *insertion* of a character from the first string, *deletion* of a character from the first string, or *substitution/replacement* of a character in the first string with a character in the second string. Thus *dogs* has an edit distance of 1 with *dog*, since only a deletion of ‘s’ would suffice for conversion. There might be more then one ways to conversion, hence the minimum edit distance is a more useful measure.

We divided the synsets into two groups. The first group contained all the synsets with frequency one, based on the English phrase. The other group contained synsets which have frequency more than one, based on their English phrase. Pair-wise edit distances were measured between every two synsets that shared the English phrase. This information would be used in future to determine which two synsets should be merged.

VI. Synonymy Detection

Synonymy is a relationship between words which makes them inter-substitutable. Yet [13] says that “natural languages abhor absolute synonyms just as nature abhors a vacuum.” Absolute synonymy is rare and restricted mostly to technical terms [14]. Near-synonyms are of greater significance and are very similar but not completely inter-substitutable or identical.

According to [15] a common approach to synonymy detection is distributional similarity. Thus synonymous words share common contexts, and thus they could be inter-substituted without changing the context. They showed that use of multilingual resources for extraction of synonyms had higher precision and recall as compared to the monolingual resources.

Turney [16] used PMI-IR (Pointwise Mutual Information and Information Retrieval) to determine the synonymy between two words. The algorithm maximizes Pointwise Mutual Information [17], [18], which in turn is based on co-occurrence [19].

We can use the above ideas to detect synonymy between the words/phrases for a given language, then merge the multilingual proto-synsets that only vary in this respect. Similarly, we can apply similarity measures to 4-tuples, e.g., if the words/phrases in all but one langugage are the same, or a number of alternatives for some languages appear together in several permutations, e.g., car-auto-auto, car-auto-voiture, automobile-auto-auto, automobile-auto-voiture, we can consider them as synonyms.

VII. Conclusion

The value of this approach is in its use of unsupervised techniques that do not require an annotated corpus. In this way, *all* words are guaranteed to be tagged with a synset, which is not often the case with other approaches. This has been done on a large dataset with more than 1.8 million words. WSD of such a large corpus is valuable even if the additional benefits of the lexical resource produced are not considered.

References

[1] D. B. Lenat, “Cyc: A large-scale investment in knowledge infrastructure,” *Communications of the ACM*, vol. 38, no. 11, pp. 33–38, 1995.

[2] G. A. Miller, “Five papers on wordnet,” *Special Issue of International Journal of Lexicogrphy*, vol. 3, no. 4, 1990.

[3] P. Vossen, Ed., *Eurowordnet: A Multilingual Database with Lexical Semantic Networks*. Kluwer Academic Publishers, 1998.

[4] M. Diab and P. Resnik, “An unsupervised method for word sense tagging using parallel corpora,” in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002, pp. 255–262.

[5] D. Kazakov and A. R. Shahid, “Unsupervised construction of a multilingual wordnet from parallel corpora,” in *Workshop on Natural Language Processing methods and Corpora in Translation, Lexicography, and Language Learning, RANLP*, 2009.

[6] E. Lefever and V. Hoste, “Semeval-2010 task 3: Cross-lingual word sense disambiguation,” in *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, 2009.

[7] R. Bruce and J. Wiebe, “Word-sense disambiguation using decomposable models,” in *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 1994, pp. 139–146.

[8] Y. K. Lee and H. T. Ng, “An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation,” in *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, 2002, pp. 41–48.

[9] E. Brill, “A simple rule-based part of speech tagger,” in *Proceedings of the Third Conference on Applied Natural Language Processing*, 1992, pp. 152–155.

[10] D. Gusfield, *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK, 1997.

[11] J. B. Kruskal, “An overview of sequence comparison: Time warps, string edits, and macromolecules,” *SIAM Review*, vol. 25, no. 2, pp. 201–237, 1983.

[12] V. I. Levenstein, “Binary codes capable of correcting, insertions and reversals,” *Sov. Phys. Dokl.*, vol. 10, pp. 707–710, 1966.

[13] A. D. Cruse, *Lexical Semantics*. Cambridge University Press, Cambridge, UK, 1986.

[14] P. Edmonds and G. Hirst, “Near-synonymy and lexical choice,” *Computational Linguistics*, vol. 28, no. 2, pp. 105–145, 2002.

[15] L. van der Plas and J. Tiedemann, “Finding synonyms using automatic word alignment and measures of distributional similarity,” in *Proceedings of ACL/COLING 2006*, 2006.

[16] P. D. Turney, “Mining the web for synonyms: Pmi-ir versus lsa on toefl,” in *Proceedings of the Twelfth European Conference on Machine Learning*, 2001, pp. 491–502.

[17] K. W. Church and P. Hanks, “Word association norms, mutual information and lexicography,” in *Proceedings of the 27th Annual Meeting of the Association of Computational Linguistics (ACL)*, 1989, pp. 76–83.

[18] K. W. Church, W. Gale, P. Hanks, and D. Hindle, *Using Statistics in Lexical Analysis*. Lawrence Erlbaum, 1991, ch. In Lexical Acquisition: Using On-Line Resources to Build a Lexicon, edited by Uri Zernik, pp. 115–164.

[19] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999.

Análisis de Opiniones con Ontologías

Enrique Vallés Balaguer, Paolo Rosso, Angela Locoro y Viviana Mascardi

***Resumen*—En este artículo presentamos un trabajo sobre análisis de opiniones llevado a cabo gracias a un enfoque innovador basado en fusión de ontologías. El objetivo de este trabajo es permitir que dos empresas puedan intercambiar y compartir los resultados de los análisis de las opiniones de sus productos y servicios.**

***Palabras clave*—Minería de opiniones, mapeo y fusión de ontologías, Web 2.0, Empresa 2.0.**

Opinion Mining using Ontolgies

***Abstract*—In this paper we present a work dealing with opinion analysis carried out thanks to an innovative approach based on ontology matching. The aim of this work is to allow two enterprises to share and merge the results of opinion analyses on their own products and services.**

***Index Terms*—Opinion mining, ontology matching and merging, Web 2.0, Enterprise 2.0.**

I. Introducción

PARA una pequeña y mediana empresa (empresa) tanto la cantidad como la calidad de información no tiene precio. El principal objetivo de la información es la de apoyar a la toma de decisiones, puesto que con ella se tendrán más bases sustentables para poder decidir cuáles son los pasos a seguir y qué rumbo hay que tomar para lograr los objetivos que se planificaron. Es por ello que en una empresa se le debe de poner una atención sumamente especial a la información que se genera cada día, la adecuada interpretación de ésta establecerá los cimientos necesarios para consolidarse como una Empresa 2.0 de éxito en el mercado. La información le permitirá identificar cuáles son las fortalezas con las que cuenta y cuáles las debilidades y sectores vulnerables que presenta como organización. Teniendo en cuenta estos datos podrá tener una planificación más alcanzable y factible, ya que podrá identificar donde tiene que aumentar los esfuerzos y que parte de la empresa necesita mayor atención. Hoy en día, la principal fuente de información para conocer los puntos fuertes y débiles de una organización es a través de las opiniones que generan los propios consumidores.

En la actualidad, una gran cantidad de las compras y de los servicios contratados que se efectúan, no están condicionados por las sugerencias de las campañas de publicidad y los trucos del marketing, sino por los comentarios que otros consumidores han escrito en los múltiples foros virtuales (públicos y privados) que hoy ofrece la Web 2.0. Con la explosión de la Web 2.0, plataformas como blogs y redes sociales, los consumidores tienen a su disposición un lugar donde compartir sus experiencias con las diferentes marcas y donde poder dar sus opiniones, positivas o negativas sobre cualquier producto o servicio. Las principales empresas empiezan a darse cuenta que estos comentarios de los consumidores pueden manejar la enorme influencia en la formación de las opiniones de otros consumidores. Las empresas que deseen crecer deben de responder a las perspicacias de los consumidores, y es por todo esto que tienen la obligación de analizar los medios de comunicación sociales, para obtener la información adecuada para modificar sus mensajes de marketing, modificar el desarrollo de los productos, etc [1].

Gracias a la Web 2.0 gana peso la opinión del ciudadano frente a las marcas y sus técnicas comerciales más tradicionales. Según el Instituto Nacional de Estadística (INE[1]), el 80% de los internautas reconoce que acude a la red para informarse sobre productos, marcas y servicios. En otro estudio realizado en 2009 por la Asociación para la Investigación de Medios de Comunicación (AIMC[2]), el 75.5% de internautas españoles admite haberse documentado en internet durante el último año, como paso previo a formalizar una compra de productos o servicios.

Dadas estas circunstancias, los responsables del marketing de las empresas tienen la obligación de supervisar en las redes sociales la información relacionada con sus productos y servicios. Sin embargo, en lo últimos años se ha producido una explosión en la Web 2.0 sin precedentes, ocasionando que la supervisión manual de las opiniones de los consumidores se convierta en un trabajo irrealizable. La empresa Technorati[3] especializada en blogs estima que 75.000 nuevos blogs son creados diariamente, con 1,2 millones de nuevos comentarios cada día en las que el consumidor comenta sobre productos y servicios.Con estos datos las empresas se ven en la necesidad de aunar esfuerzos por encontrar un método automático que sea capaz de analizar las opiniones de los consumidores.

Efectivamente, encontrar un método capaz de clasificar automáticamente, como positivas o negativas, las opiniones en un texto, sería de enorme utilidad para el marketing de

Manuscrito recibido el 24 de febrero del 2010. Manuscrito aceptado para su publicación el 31 de mayo del 2010.

Enrique Vallés Balaguer y Paolo Rosso trabajan en NLE Lab - ELiRF, DSIC, Universidad Politécnica de Valencia, Valencia, España (e-mail: {evalles,prosso}@dsic.upv.es).

Angela Locoro y Viviana Mascardi trabajan en DISI, Università degli Studi di Genova, Genova, Italia (e-mail: {angela.locoro,viviana.mascardi}@unige.it).

[1] http://www.ine.es

[2] http://www.aimc.es/aimc.php

[3] http://www.technorati.com/

las empresas; del mismo modo que para aquellos que buscan información a partir de grandes cantidades de noticias y de datos Web [2]; incluso también se beneficiarían los sistemas de recomendación y colaboración [3].

En este artículo proponemos un método automático para clasificar la polaridad de las opiniones. Sin embargo, al igual que las empresas están interesadas en las opiniones de los consumidores, también están interesadas en conocer de qué producto o característica del producto están opinando. Este hecho se ve reforzado puesto que en una misma opinión se puede comentar dos conceptos diferentes del mismo dominio con polaridades distintas; es más, incluso en una misma frase. Por ejemplo:

El hotel era bonito aunque las habitaciones estaban sucias

En esta frase el *hotel* tiene una polaridad positiva; pero por otro lado, *las habitaciones* tiene una polaridad negativa; por otro lado, en general esta opinión es positiva. El interés de una empresa no es solamente conocer la polaridad de la opinión en conjunto, sino conocer que características de un producto, gustan o no a los consumidores. Esto ayudará a una empresa a mejorar el producto o servicio según los gustos de los clientes. Volviendo al ejemplo, no sólo es interesante conocer que el cliente esté de acuerdo, sino poder saber que cierto aspecto, en este caso concreto *las habitaciones*, podrían mejorarse.

Es por esto que, dado que las empresas disponen de una ontología propia en la cual se incluyen todos sus productos, hemos basado nuestro algoritmo en ontolgías para poder encontrar todas aquellas opiniones realizadas sobre los productos pertenecientes a ésta.

Sin embargo, dado el coste de conseguir la opinión de los consumidores, puede que varias empresas decidan compartir e intercambiar la información que poseen sobre las opiniones de los consumidores, o incluso, llegado el caso extremo en el que dos empresas se fusionen. En estos casos, se debe de encontrar algún método que sea capaz de poder analizar automáticamente las opiniones de los clientes y además que sea compatible con las diferentes ontologías. Por este motivo, proponemos un algoritmo que incluye dentro del análisis de opiniones, una fusión de ontologías.

El objetivo de este artículo es un estudio preliminar sobre el análisis de opiniones con ontologías dentro del dominio del turismo. Para ello nos ponemos en la piel de dos empresas dedicadas al turismo las cuales analizarán opiniones de consumidores sobre conceptos del dominio del turismo y después simularemos que dichas empresas deciden compartir la información.

El artículo está organizado de la siguiente manera: en la sección 2 se introduce el problema del análisis de opiniones, el análisis de opiniones basado en ontologías, y expone el algoritmo propuesto. En la sección 3, se introduce el problema de la fusión de ontologías. En la sección 4 exponemos los resultados obtenidos en los experimentos de fusión de ontologías, y posteriormente, los experimentos en el análisis de opiniones con ontologías. Finalmente en la sección 5 comentamos las conclusiones y la línea a seguir.

II. Análisis de Opiniones

En la actualidad la información se ha convertido en los cimientos que sostiene al mundo empresarial. La información facilita a la empresa la identificación de sus puntos fuertes, y mucho más importante, sus puntos débiles donde deberá aumentar sus recursos para crecer en el mercado. La mayor fuente de información existente es la Web 2.0, puesto que gracias a las redes sociales, los consumidores tienen a su merced un espacio donde compartir las experiencias vividas con las diferentes marcas. Para las empresas es de vital importancia poder disponer de dicha información.

No obstante, en la actualidad se generan al día una gran cantidad de blogs con sus respectivos comentarios, llegando incluso a generar millones de comentarios al día. Como consecuencia, una supervisión manual de las diferentes opiniones de los consumidores se convierte en una tarea irrealizable. Sin embargo, la información proporcionada por los comentarios de los consumidores es la materia prima más preciada con la que una Empresa 2.0 puede construir su futuro; y es por esto que las empresas realizan continuos esfuerzos para encontrar un método automático que sea capaz de analizar las opiniones de los consumidores.

A. *Subjetividad y Opiniones*

La adquisición de la polaridad de las palabras y frases es en sí misma una línea activa de investigación dentro del análisis de sentimientos, promovido por el trabajo de Hatzivassiloglous y McKeown en la predicción de la polaridad o la orientación semántica de los adjetivos [4]. Desde entonces se han propuesto varias técnicas para determinar la polaridad de palabras: desde medidas probabilísticas de asociación de palabras [5], así como técnicas que explotan la información sobre relaciones léxicas [6], [7], y utilizando glosas [8] por ejemplo de WordNet [9].

Hasta la actualidad se han realizado diversos trabajos dentro de la disciplina de minería de opiniones con diferentes objetivos: desde determinar si un texto contiene opiniones, es decir, clasificar un texto como subjetivo u objetivo tales como [10], [11], hasta trabajos que se centran en determinar la polaridad (orientación semántica) a nivel de palabras y frases, tales como [12], [13], [14], [15]; existen también estudios donde no sólo determinan la polaridad sino el nivel de ésta, es decir, si es alto/medio/bajo positivo/negativo [16], [17], [18], [19].

Otros estudios relacionados con el análisis de opiniones se centran en la extracción de emociones a partir del texto [20], [21], [22], [23]; otros en cambio se centran en la extracción del opinante para tareas de búsqueda de respuesta (QA, por sus siglas en inglés) de opiniones [24]; también, existen estudios destinados a encontrar las tendencias de los consumidores en los blogs [25], [26]. Otros trabajos relacionados miden la influencia que tienen las opiniones introducidas en los blogs sobre los consumidores [27], [28], [29].

B. Análisis deOopiniones basado en Ontologías

Sin embargo, la mayoría de los trabajos anteriormente mencionados, se centran en la obtención de la opinión sobre un tema específico. Muy pocos tienen en cuenta que un tema puede estar dividido en diferentes subtemas, los cuales describen dos aspectos distintos del tema. Es más, dos subtemas del mismo tema pueden tener polaridades opuestas. Uno de los primeros estudios que se centra en este aspecto fue el trabajo realizado por Hu y Liu [30], el cual se centra en la obtención de las opiniones sobre las propiedades de un producto. Dichas propiedades se obtenían utilizando minería de asociación entre palabras.

Recientemente se han realizado dos trabajos con el objetivo de calcular la polaridad por medio de las propiedades de un concepto, pero a diferencia del trabajo de Hu y Liu, en estos trabajos construyen primero las ontologías del dominio al que pertenecen los textos del corpus y, a partir de éstas, extraen los calificativos de las propiedades de los productos que aparecen en los textos.

El primer trabajo de ellos es el estudio de Zhou y Chaovalit en [3]. En dicho estudio el primer paso que se realiza es generar una ontología; la generación de esta ontología se realiza manualmente por analistas a partir del corpus que posteriormente se utilizará para los experimentos de minería de opiniones. Una vez construída la ontología, suponen que dado un concepto c de la ontología, el cual está descrito con n propiedades: $p_1, p_2, ..., p_n$, y que en un texto t se puede expresar opiniones sobre cualquiera de estas propiedades; por tanto, de acuerdo con el número de propiedades de c, t puede dividirse en m segmentos (siendo $m < n$). Es decir que, la predicción de la polaridad de t se define como:

$$polaridad(t) = \begin{cases} positivo, & \text{si } \left(\dfrac{1}{m}\displaystyle\sum_{i=1}^{n} w_i v_i \geq 0\right) \\ negativo, & \text{en caso contrario} \end{cases} \tag{1}$$

siendo $w_i \in [0, 1]$ el peso de la propiedad p_i y $v_i \in [-1, 1]$ el valor de la polaridad de la propiedad p_i calculadas por *estimación de máxima verosimilitud*.

El otro estudio que se ha realizado hasta ahora sobre análisis de opiniones basado en ontologías, es el trabajo de Zhao y Li en [31]. En este estudio, se genera una ontología automáticamente a partir del corpus que posteriormente se utilizará para el análisis de opiniones. Una vez generada la ontología, los autores proponen extraer los calificativos de las propiedades de los conceptos por medio de la ontología, para posteriormente identificar la polaridad de dichos calificativos utilizando la herramienta SentiWordNet[4]. Una vez extraídos los calificativos y calculado sus polaridades, obtienen la orientación semántica del texto a partir de la jerarquía de la ontología, para ello calculan la orientación negativa, positiva y neutra según las siguientes ecuaciones:

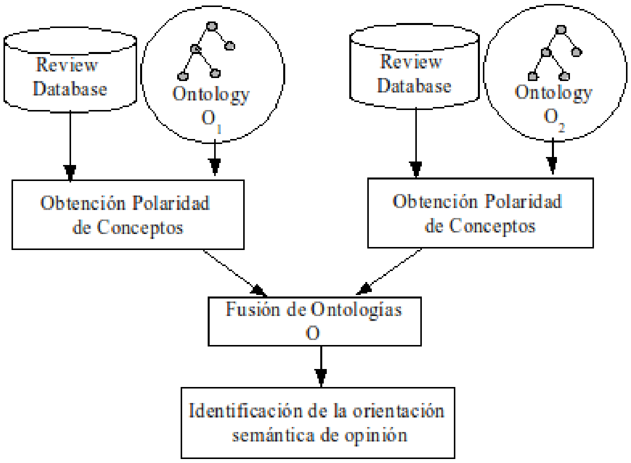

Review Database
Ontology O1
Review Database
Ontology O2
Obtención Polaridad de Conceptos
Obtención Polaridad de Conceptos
Fusión de Ontologías O
Identificación de la orientación semántica de opinión

Fig. 1. Algoritmo para el análisis de opiniones via fusión de ontologías.

$$op_{hlc}(neg) = \frac{\sum_{ch\_node_{ws_i \in neg}} score(ch\_node_{ws_i \in neg})}{|ch\_node_{ws_i \in neg}|} \tag{2}$$

$$op_{hlc}(pos) = \frac{\sum_{ch\_node_{ws_i \in pos}} score(ch\_node_{ws_i \in pos})}{|ch\_node_{ws_i \in pos}|} \tag{3}$$

$$op_{hlc}(neu) = \frac{\sum_{ch\_node_{ws_i \in ne}} score(ch\_node_{ws_i \in ne})}{|ch\_node_{ws_i \in ne}|} \tag{4}$$

donde $|ch\_node_{ws_i}|$ representa la cardinalidad de todos los hijos con la misma opinión. Por último, escogen como orientación del texto aquella de las tres que es mayor que el resto.

C. Análisis de Opiniones via Fusión de Ontologías

Sin embargo, cuando dos empresas tengan la necesidad de compartir información sobre las opiniones de los productos (o llegado el caso en que dos empresas se fusionen), hay que encontrar una manera para poder analizar las opiniones y posteriormente enviar la información a ambas empresas intentando perder lo menos posible de ésta. Para dicho problema proponemos un algoritmo de análisis de opiniones via fusión de ontologías. Se puede ver un esquema de dicho algoritmo en la figura 1. El algoritmo propone que la empresa e_1 obtenga la polaridad de los conceptos de su ontología (O_1), del mismo modo la empresa e_2 obtendrá la polaridad de los conceptos de su ontología O_2. Posteriormente se realizará una fusión de ontologías mediante una ontología general O (*upper ontology*, de nivel más alto) y a través de ésta, se realizará un cálculo de la orientación semántica de la opinión t mediante la ecuación:

$$os_{neg}(t) = \sum_{c \in O} v_{neg}(c) \tag{5}$$

[4] http://sentiwordnet.isti.cnr.it/

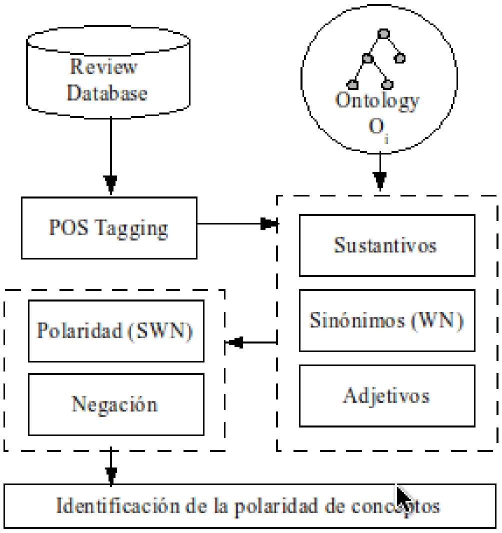


Fig. 2. Algoritmo para la identificación de la polaridad de conceptos.

$$os_{pos}(t) = \sum_{c \in O} v_{pos}(c) \tag{6}$$

donde c son los conceptos que pertenecen a la ontología O, $v_{neg}(p)$ es la polaridad negativa de la propiedad p y $v_{pos}(p)$ la polaridad positiva. Por tanto, la orientación semántica de la opinión (t) se define como:

$$polaridad(t) = \begin{cases} positivo, & \text{si } os_{pos}(t) > os_{neg}(t) \\ negativo, & \text{en caso contrario} \end{cases} \tag{7}$$

Para la obtención de la polaridad de los conceptos y propiedades de las ontologías se propone los siguientes pasos (se puede ver un esquema en la figura 2):

- En el primer paso etiquetamos cada una de las palabras de los textos (Part Of Speech tagger); para este paso hemos utilizado el toolkit GATE[5].
- Posteriormente, buscamos las frases que contienen algún concepto (c) de la ontología (O_i); para ello buscamos en cada texto los nombres (o grupos de nombres) que coinciden con un concepto de la ontología. Para aquellas frases que no contengan ningún concepto de la ontología, utilizamos WordNet[6] (WN) para encontrar sinónimos de los nombres que aparecen en la frase, que pueden ser sinónimos a su vez, de algún concepto de la ontología.
- Seguidamente, extraemos de las frases obtenidas en el paso anterior, los adjetivos adyacentes de cada concepto $(adj(c))$.
- En el siguiente paso obtenemos la polaridad de los adjetivos utilizando SentiWordNet (SWN).
- Comprobamos que la frase es afirmativa, en caso contrario, invertimos la polaridad que nos devuelve SentiWordNet.
- Guardamos las polaridades de cada concepto en un archivo XML con los datos del concepto y el valor de la polaridad.

Hay que destacar, como ya hemos comentado, tanto en el estudio de Zhou y Chaovalit como en el de Zhao y Li, la ontología utilizada se crea a partir del corpus que posteriormente se utilizará en los experimentos para el análisis de opiniones, por otro lado en nuestro trabajo utilizamos ontologías existentes con anterioridad. Nuestra opinión es que esta forma es la más cercana al problema del mundo real, puesto que las empresas tienen una ontología propia y no creemos que sea adecuado modificar su ontología para realizar la búsqueda de opiniones.

Sin embargo para la realización de nuestro algoritmo hemos de encontrar un método de fusión de ontologías que sea eficaz.

III. Fusión de Ontologías

Las empresas interactuan con la información almacenada mediante ontologías, las cuales proveen un vocabulario de un dominio específico y una especificación de los términos utilizados en dicho vocabulario [32]. Sin embargo, la capacidad de los humanos para tener diferentes puntos de vista de un mismo concepto no tiene límites. Por ello, no es de extrañar que diferentes analistas, desarrolladores e ingenieros del conocimiento tengan diferentes conceptos de la realidad sobre un mismo dominio, generando diferentes ontologías.

Sin embargo, por circunstancias del mercado las empresas pueden fusionarse, y en este caso es necesario no perder ninguna información almacenada anteriormente por cada una de ellas, sobre todo teniendo en cuenta lo esencial que es la información. Tampoco sería coherente utilizar dos ontologías distintas para un mismo dominio en una misma empresa. Por tanto, el primer paso técnico con relación a la información que se realiza es encontrar las relaciones semánticas entre los conceptos de las diferentes ontologías. Este proceso se conoce como *fusión de ontologías* [32].

Realizar la fusión manualmente es un trabajo laborioso, y que en la mayoría de los casos se convierte en imposible. Por consiguiente, es necesario encontrar un método (semi-)automático que facilite la fusión entre diferentes ontologías. El problema de la fusión entre ontologías puede ser abordado desde diversos puntos de vista y este hecho se refleja en la variedad de métodos de fusión que se han propuesto en la literatura. Muchos de ellos tienen sus raíces en el problema clásico de la fusión de esquemas en el área de las bases de datos, tales como Artemis [33], COMA [34], Cupid [35], Similarity Flooding [36]...; mientras que otros han sido específicamente diseñados para trabajar con ontologías, como son GLUE [37], QOM [38], OLA [39], S-Match [40], ASCO [41], PROMPT [42], HCONEE-merge [43] y SAMBO [44]. Algunos de estos métodos se basan en el razonamiento

[5] http://gate.ac.uk/

[6] http://wordnet.princeton.edu/

formal de la estructura de las descripciones de la entidad como S-Match y OLA; otros en cambio, utilizan una combinación de similitud y grafos de razonamiento como en Similarity Flooding; e incluso otros se basan en algoritmos de aprendizaje automático como GLUE.

En estas investigaciones se explota la información lingüística, la estructura de la información así como el dominio del conocimiento para encontrar un buen alineamiento entre los elementos de cada ontología [45]. Algunos resultados experimentales de algunas herramientas sobre un test estándar de parejas de ontologías aparecen en I3CON 2003[7], EON 2004[8] y OAEI 2005[9]. Dos buenas revisiones sobre los trabajos realizados para el estudio de la fusión de ontologías puede encontrarse en [46], [47].

A. *Ontologías de Nivel más Alto*

Una ontología de nivel más alto (*upper ontology*), como se define en [48], es una ontología independiente del dominio, que proporciona un marco por el cual distintos sistemas pueden utilizar una base común de conocimiento y desde el cual se pueden derivar ontologías de dominio específico. Los conceptos expresados son destinados a ser fundamentales y universales para garantizar la generalidad y la expresividad de una amplia gama de dominios [49]. Una ontología de nivel más alto se caracteriza a menudo como la representación de conceptos que son básicos para la comprensión humana del mundo [50].

Existen varias ontologías de nivel más alto implementadas, como BFO [51], Cyc [52], DOLCE [53], GFO [54], PROTON [55], Sowa's ontology [56] y SUMO [57]. Se puede encontrar una comparación de las distintas ontologías de nivel más alto mencionadas anteriormente en [58].

En nuestros experimentos hemos utilizado SUMO y OpenCyc. En los siguientes sub-apartados describimos cada una de estas ontologías generales.

1) SUMO: SUMO[10] (*Suggested Upper Merged Ontology*) es una ontología creada por Teknowledge Corporation[11] con una amplia contribución de la lista de correo SUO[12] (*Standard Upper Ontology*), y fue propuesta como documento de iniciación para el Grupo de Trabajo SUO [57], un grupo de trabajo con colaboradores de los campos de la ingeniería, la filosofía y ciencias de la información. SUMO es una de las más grandes ontologías formales públicas existentes hoy día. Cuenta con 20.000 términos y 70.000 axiomas cuando todos los dominios son combinados. SUMO está compuesto por una ontología de nivel medio (*MId-Level Ontology* (MILO)), ontologías de comunicaciones, países y regiones, computación distribuida, economía, finanzas, componentes de ingeniería, geografía, gobierno, militar, sistema de clasificación industrial de Norte América, gente, los elementos físicos, cuestiones internacionales, transporte, aeropuertos mundiales y armas de destrucción masiva.

TABLA I
ONTOLOGÍAS USADAS EN LOS EXPERIMENTOS.

Ontologías	Conceptos
ETP-tourism	194
qallme-tourism	125
Tourism-ProtegeExportOWL	86
TravelOntology	35
e-tourism	20

2) OpenCyc: El *Cyc Knowledge Base*[13] (KB) es una base de conocimiento multicontextual desarrollada por Cycorp. Cyc es una representación formalizada de una cantidad enorme de conocimiento fundamental humano: hechos, reglas básicas, y heurísticas para razonar sobre los objetos y los acontecimientos de la vida cotidiana. Cyc KB consiste en términos y las aserciones que relacionan los términos.

KB Cyc se divide en varias *microteorías*, cada una es esencialmente un conjunto de las aserciones que comparten un juego común de suposiciones; algunas *microteorías* son enfocadas en un dominio particular de conocimiento, en un nivel particular de detalle, etc.

Actualmente, el KB Cyc contiene casi doscientos mil términos y varias docenas de aserciones introducidas manualmente sobre cada término. Cyc es un producto comercial, sin embargo Cycorp dispone de una versión de código abierto OpenCyc[14].

IV. EXPERIMENTOS EN EL DOMINIO DEL TURISMO

La intención de los experimentos es la de validar el algoritmo propuesto de análisis de opiniones via fusión de ontologías en el dominio del turismo. Para ello hemos realizado dos fases de experimentos: en la primera fase, se ha realizado un estudio para encontrar el método más idóneo para la etapa de fusión de ontologías en el dominio del turismo; y en la segunda fase, se han realizado los experimentos dedicados a la validación del algoritmo propuesto de análisis de opiniones via fusión de ontologías en el dominio del turismo.

A. *Fusión de Ontologías con SUMO y OpenCyc*

Para realizar la búsqueda de ontologías existentes de en el dominio del turismo, hemos utilizado Swoogle[15]. Tras estudiar las ontologías existentes en el dominio del turismo, hemos seleccionado para nuestros experimentos las ontologías que se muestran en la tabla I.

Hemos empezado creando un alineamiento manual entre todas las ontologías, para utilizarlo como base para medir la calidad de los alineamientos generados por las técnicas utilizadas en los experimentos. Posteriormente, hemos

[7] http://www.atl.external.lmco.com/projects/ontology/i3con.html

[8] http://km.aifb.uni-karlsruhe.de/ws/eon2004/

[9] http://oaei.inrialpes.fr/2005/

[10] http://dream.inf.ed.ac.uk/projects/dor/sumo/

[11] http://www.teknowledge.com/

[12] http://suo.ieee.org/

[13] http://www.cyc.com

[14] http://opencyc.org/

[15] http://swoogle.umbc.edu/

realizado los tests sobre los métodos directos. Para estas pruebas hemos utilizado la API de Euzenat[16], la cual es una API para la ejecución de alineamientos entre ontologías. El Alignment API permite la ejecución de diferentes medidas de distancia entre textos. En los tests hemos utilizado las medidas *equal* que compara si el texto es exacto, *SMOA* [59] y por último, la distancia de edición de *Levenshtein* [60].

Seguidamente, hemos realizado las pruebas con las técnicas vía ontologías de nivel más alto. Para ello hemos utilizado la API desarrollada en [61]. La API está desarrollada en Java y permite la ejecución de dos métodos diferentes[17]:

- El método no estructurado: los conceptos $c \in o$ y $c' \in o'$ están relacionados si corresponden al mismo concepto $c_u \in u$, donde u es la ontología de nivel más alto; o y o' son las ontologías a fusionar.
- El método estructurado: los conceptos $c \in o$ y $c' \in o'$ están relacionados si:
  - Los conceptos c y c' corresponden al mismo concepto $c_u \in u$, por tanto están relacionados con una medida de confianza de $conf_1 * conf_2$.
  - El concepto c corresponde al concepto c_u, c' corresponde con el concepto c'_u, y c'_u es un super-concepto de c_u en u (o viceversa), por tanto están relacionados con una medida de confianza de $conf_1 * conf_2 * df$.
  - El concepto c corresponde al concepto c_u, c' corresponde con el concepto c'_u, y c'_u tiene algún super-concepto en común con c_u, por tanto están relacionados con una medida de confianza de $conf_1 * conf_2 * df^2$.

En la ejecución primero hemos utilizado la API de Euzenat entre la ontología de nivel más alto y cada una de las ontologías a alinear. Al utilizar la API de Euzenat nos permitía ejecutar cada una de las funciones de distancia anteriormente citadas. Una vez realizados los alineamientos de las ontologías seleccionadas con las ontologías de nivel más alto utilizamos la API de [62] para realizar los alineamientos estructurados y no estructurados. Este proceso lo realizábamos para cada una de las dos ontologías de nivel más alto.

En la tabla II[18] aparecen los resultados obtenidos en los experimentos para el alineamiento entre las ontologías *ETP-tourism*[19] y *qallme-tourism*[20]. Los campos que aparecen en la tabla corresponden a: función de distancia (Dis), ontología de nivel más alto utilizada (UO), método estructurado o no estructurado (Met), número de alineamientos encontrados (Enc), número de alineamientos correctos (Cor), y las tres medidas: precisión (Pre), recall (Rec) y F-measure (F-M). Por último, hemos señalado en la tabla los mejores resultados para cada test en las diferentes medidas.

TABLA II
RESULTADOS DE MAPEO DE ONTOLOGÍAS.

Dis	**UO**	**Met**	**Enc**	**Cor**	**Pre**	**Rec**	**F-M**
Man.	None	None	98	98	1.00	1.00	1.00
equal	None	None	195	80	0.41	0.82	0.55
smoa	None	None	221	84	0.38	**0.86**	0.52
Lev.	None	None	205	82	0.40	0.84	0.54
equal	Sumo	NoSt	18	14	**0.78**	0.14	0.24
smoa	Sumo	NoSt	415	83	0.20	0.85	0.32
Lev.	Sumo	NoSt	264	74	0.28	0.76	0.41
equal	Sumo	Str	20	14	0.70	0.14	0.24
smoa	Sumo	Str	461	83	0.18	0.85	0.30
Lev.	Sumo	Str	264	74	0.28	0.76	0.41
equal	Cyc	NoSt	38	16	0.42	0.16	0.24
smoa	Cyc	NoSt	143	80	0.56	0.82	0.67
Lev.	Cyc	NoSt	122	78	0.64	0.80	**0.71**
equal	Cyc	Str	53	16	0.30	0.16	0.21
smoa	Cyc	Str	200	80	0.40	0.82	0.54
Lev.	Cyc	Str	144	78	0.54	0.80	0.64

Observando los datos podemos destacar que al aplicar la técnica de *matching* vía ontologías de nivel más alto con el método no estructurado [61] obtenemos una ganancia media de precisión frente a los métodos directos, de alrededor del 4,3 % para SUMO y de un 36,6 % para OpenCyc. En recall se produce una pérdida media de 31,2 % para SUMO y de 30 % para OpenCyc. Y finalmente, en F-measure se produce una pérdida media de 39,6 % para SUMO y una ganancia media de 1,3 % para OpenCyc. En cambio, al aplicar la técnica de *matching* vía ontología de nivel más alto con el método estructurado obtenemos una pérdida media de precisión frente a los métodos directos del 3,97 % para SUMO y una ganancia media de un 4,5 % para OpenCyc. En recall se produce una pérdida media de 31,2 % para SUMO y de 30 % para OpenCyc. Y finalmente, en F-measure se produce una pérdida media del 41 % para SUMO y de 13,2 % para OpenCyc.

Estos datos nos podrían indicar que se obtienen mejores resultados con las técnicas directas que realizando el mapeo vía ontologías de nivel más alto. Sin embargo, en los diferentes experimentos se ha comprobado que conforme aumenta el número de términos de cada ontología, es notable una mejora en los métodos utilizando ontologías de nivel más alto respecto a los métodos directos en cuanto a las medidas de precisión y F-measure, y una mínima pérdida en recall. Esta mejoría se nota sobre todo al utilizar OpenCyc como ontología de nivel más alto. Este hecho se puede comprobar en la tabla II. Esto nos puede dar un indicio de que cuanto mayor son las ontologías es preferible utilizar las técnicas vía ontología de nivel más alto.

Por tanto, hemos decidido utilizar para el paso de fusión de ontologías, los métodos no estructurados vía ontología de nivel más alto, seleccionando Cyc como ontología de nivel más alto, y utilizando como ontologías de origen *ETP-tourism* y *qallme-tourism*.

[16] http://alignapi.gforge.inria.fr/

[17] Información detallada puede verse en [62]

[18] Las limitaciones de espacio nos impiden ilustrar el resto de alineamientos; para una descripción más detallada véase http://users.dsic.upv.es/grupos/nle/?file=kop4.php

[19] http://www.info.uqam.ca/Members/valtchev_p/ mbox/ETP-tourism.owl

[20] http://qallme.fbk.eu/

TABLA III
RESULTADOS OBTENIDOS DIVIDIENDO EL CORPUS.

Ontología	Num.	Acc.
ETP Tourism qallme-tourism	1.500 1.500	72,41 % 70,92 %
Mapeo de ontologías	3.000	71,13 %

TABLA IV
RESULTADOS OBTENIDOS CON EL CORPUS COMPLETO.

Ontología	Num.	Acc.
ETP Tourism qallme-tourism	3.000 3.000	72,2 % 71,2 %
Mapeo de ontologías	3.000	71,33 %

B. *Análisis de Opiniones con OpenCyc*

Una vez validado el método para la fusión de ontologías, hemos realizado los experimentos para el análisis de opiniones vía fusión de ontologías. Para la realización de dichos experimentos hemos creado un corpus formado por 3.000 textos de opiniones (1.500 positivos y 1.500 negativos) extraídas desde TripAdvisor[21]. Los textos corresponden a conceptos como hoteles, restaurantes y ciudades. TripAdvisor nos ofrecía una ventaja, ya que en este blog los usuarios no sólo escriben sus opiniones sino que además puntúan el producto entre Excelente, Muy bueno, Regular, Malo y Pobre. De esta forma no teníamos que analizar las opiniones manualmente para clasificar el valor de la orientación semántica de cada una de ellas.

En los experimentos hemos intentado simular como actuarían dos empresas dedicadas al dominio del turismo para obtener información sobre algunos productos ofertados. Para ello primero realizarán cada una de ellas un proceso de análisis de opiniones sobre el corpus creado con dos diferentes ontologías del dominio del turismo. Posteriormente, suponemos que las empresas desean intercambiar información, o compartirla, o en un caso extremo que dos empresas se fusionen, por tanto realizarán un proceso de fusión de ontologías. Con estos experimentos podremos tener una base para decidir la validez de nuestro algoritmo basado en la fusión de ontologías para el análisis de opiniones.

Para poder medir la eficacia del algoritmo propuesto, hemos realizado dos diferentes experimentos: en el primer experimento hemos separado el corpus para cada una de las dos empresas, con la intención de simular que ocurriría si dos empresas analizan diferentes textos antes de compartir la información sobre el análisis de opiniones; y en el segundo, hemos utilizado el corpus completo para las dos ontologías, simulando que dos empresas analizan anteriormente los mismos textos.

En la tabla III se muestran los resultados obtenidos en el primer experimento. Un dato destacable es que tras realizar el proceso de fusión de ontologías se obtiene una tasa de aciertos (Acc) de 71,33 % que es cercano al obtenido por separado en cada ontología, incluso es superior al obtenido con la ontología *qallme-tourism*. Estos resultados nos dan a entender que al realizar el proceso de fusión de ontologías no se pierden datos referentes al proceso de análisis de opiniones realizado con antelación.

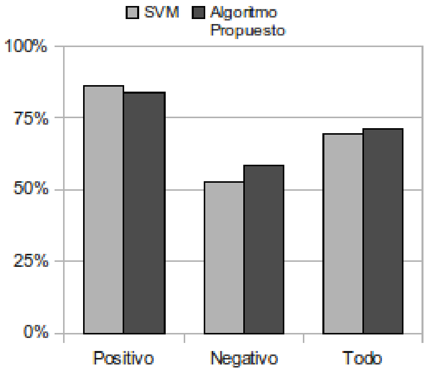


Fig. 3. Resultados de minería de polaridad.

Los resultados del segundo experimento realizado el proceso de calcular la polaridad para cada ontología con el corpus completo se muestra en la tabla IV. Como se observa en los resultados tras realizar el proceso de fusión de ontologías es muy similar al obtenido en el experimento anterior. Estos resultados nos dan a entender que al realizar el proceso de fusión de ontologías no se pierden datos referentes al proceso de análisis de opiniones realizado con antelación.

En la figura 3 se muestra una comparación entre los resultados obtenidos mediante nuestro algoritmo propuesto y utilizando un método de aprendizaje automático como *support vector machine* (SVM). La figura muestra que nuestro algoritmo incrementa el porcentaje de aciertos casi en dos puntos, de un 69,5 % a un 71,33 %. Es más también se observa que nuestro algoritmo tiene menos diferencia entre los resultados obtenidos con las opiniones positivas y las negativas, que utilizando SVM donde los resultados con las opiniones positivas son mucho mayores que en las negativas.

Una interesante observación que se desprende de los resultados de los experimentos y que se ve reflejado en la figura 3 es la diferencia de porcentajes de aciertos que existe cuando examinamos únicamente las opiniones positivas frente a cuando examinamos las opiniones negativas. Esto no ocurre únicamente con nuestro algoritmo propuesto, sino que también cuando utilizamos SVM. Analizando los textos que componen nuestro corpus hemos observado que las personas cuando estamos en desacuerdo con algún producto o servicio tenemos cierta tendencia a utilizar la ironía y el sarcasmo [63], [64]. Éste provoca que al extraer los adjetivos para obtener la orientación semántica, éstos tendrán la polaridad cambiada, llevando a clasificar los textos con la polaridad incorrecta. Pero este hecho, aunque es mucho más frecuente en

[21] http://www.tripadvisor.com

opiniones negativas, también se utiliza pero con menor medida en opiniones positivas.

V. Conclusiones y Trabajo Futuro

En este trabajo se ha presentado un método para el análisis de opiniones vía fusión de ontologías. Esta fusión permitirá a empresas poder compartir y/o intercambiar información sobre las opiniones de los consumidores con respecto a productos. Sin embargo, se ha demostrado la dificultad que entraña la tarea de analizar las opiniones en la Web 2.0 de los usuarios de blogs a través de una ontología, principalmente cuando esta ontología esta preestablecida. En estudios anteriores como [31] y [3], las ontologías en cambio se generaban a partir del corpus con lo que facilitaba la búsqueda de conceptos. Sin embargo, creemos que nuestros experimentos están más próximos al problema del mundo real, puesto que las empresas ya poseen de antemano una ontología del dominio. No obstante, hemos comprobado cómo al realizar el proceso de fusión de ontologías no se pierde prácticamente ningún dato de los anteriormente calculados por el análisis de opiniones.

Un hecho resaltable en el proceso de fusión de ontologías es que conforme aumenta el número de términos de las ontologías orígenes, se obtiene mejores resultados utilizando los métodos vía ontologías de nivel más alto respecto a los métodos directos en relación con las medidas de precisión y F-measure, y una leve pérdida en recall. En cuanto a las ontología de nivel más alto, se observa que con OpenCyc hay una leve mejora frente a SUMO, en precisión y F-measure, y una mayor mejoría de más del 10 % en recall. Por tanto, es lógico pensar que es preferible utilizar en el dominio del turismo OpenCyc.

Para un futuro estudio sería interesante utilizar no sólo los adjetivos para hallar la polaridad de los textos sino también verbos o adverbios. Otro aspecto interesante a tener en cuenta en futuros trabajos sería introducir algún método automático para detectar la ironía y el sarcasmo. Esto nos ayudaría a poder clasificar correctamente la polaridad, ya que si utilizamos el sarcasmo podemos invertir la polaridad de sus palabras. Detectar automáticamente el sarcasmo sería efectivo sobre todo para mejorar el porcentaje de aciertos en las opiniones negativas.

Agradecimientos

El trabajo de los dos primeros autores se engloba dentro del proyecto del MICINN: *TEXT-ENTERPRISE 2.0: Técnicas de Comprensión de textos aplicadas a las necesidades de la Empresa 2.0* (TIN2009-13391-C04-03). El trabajo ha sido el resultado de una estancia de Erasmus-Máster de 4 meses en la Universidad de Génova.

Referencias

[1] J. Zabin and A. Jefferies, "Social media monitoring and analysis: Generating consumer insights from online conversation," Aberdeen Group Benchmark Report, January 2008.

[2] T. Wilson, J. Wiebe, and R. Hwa, "Just how mad are you? finding strong and weak opinion clauses," *AAAI'04: Proceedings of the 19th national conference on Artifical intelligence*, pp. 761–769, 2004.

[3] L. Zhou and P. Chaovalit, "Ontology-supported polarity mining," *Journal of the American Society for Information Science and Technology*, vol. 59, no. 1, pp. 98–110, 2008.

[4] V. Hatzivassiloglous and K. R. McKeown, "Predicting the semantic orientation of adjectives," *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, pp. 174–181, 1997.

[5] P. D. Turney and M. L. Littman, "Measuring praise and criticism: Inference of semantic orientation from association," *ACM Trans. Inf. Syst.*, vol. 21, no. 4, pp. 315–346, 2003.

[6] J. Kamps and M. Marx, "Words with attitude," *1st International WordNet Conference*, pp. 332–341, 2002.

[7] S. Kim and E. Hovy, "Determining the sentiment of opinions," *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, pp. 1267–1373, 2004.

[8] A. Esuli and F. Sebastiani, "Determining the semantic orientation of terms through gloss classification," *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pp. 617–624, 2005.

[9] A. Andreevskaia and S. Bergler, "Mining wordnet for a fuzzy sentiment: Sentiment tag extraction from wordnet glosses," *Proceedings of the 11rd Conference of the European Chapter of the Association for Computational Linguistics (EACL-2006)*, pp. 209–216, 2006.

[10] E. Riloff, J. Wiebe, and W. Phillips, "Exploiting subjectivity classification to improve information extraction," in *Proc. of the NCAI*, vol. 20, 2005, p. 1106.

[11] J. Wiebe, T. Wilson, and C. Cardie, "Annotating expressions of opinions and emotions in language," in *Language Resources and Evaluation*, 2005, pp. 165–210.

[12] A. Esuli and F. Sebastiani, "Determining term subjectivity and term orientation for opinion mining," *EACL 2006, 11st Conference of the European Chapter of the Association for Computational Linguistics*, pp. 193–200, 2006.

[13] H. Kanayama and T. Nasukawa, "Fully automatic lexicon expansion for domain-oriented sentiment analysis," *EMNLP '06: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pp. 355–363, 2006.

[14] P. D. Turney, "Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews," *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pp. 417–424, 2004.

[15] B. Pang and L. J. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pp. 271–278, 2004.

[16] K. Dave, S. Lawrence, and D. M. Pennock, "Mining the peanut gallery: opinion extraction and semantic classification of product reviews," *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pp. 519–528, 2003.

[17] A. B. Goldberg and X. Zhu, "Seeing stars when there aren't many stars: graph-based semi-supervised learning for sentiment categorization," *TextGraphs '06: Proceedings of TextGraphs: the First Workshop on Graph Based Methods for Natural Language Processing on the First Workshop on Graph Based Methods for Natural Language Processing*, pp. 45–52, 2006.

[18] K. Shimada and T. Endo, "Seeing several stars: A rating inference task for a document containing several evaluation criteria," *PAKDD 2008: Proceedings of Advances in Knowledge Discovery and Data Mining, 12th Pacific-Asia Conference*, pp. 1006–1014, 2008.

[19] S. Baccianella, A. Esuli, and F. Sebastiani, "Multi-facet rating of product reviews," *ECIR '09: Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, pp. 461–472, 2009.

[20] C. Strapparava and A. Valitutti, "WordNet-Affect: an affective extension of WordNet," *LREC '04: Proceedings of the 4th International Conference on Language Resources and Evaluation*, vol. 4, pp. 1083–1086, 2004.

[21] C. Strapparava and R. F. Mihalcea, "Semeval-2007 task 14: affective text," *SemEval '07: Proceedings of the 4th International Workshop on Semantic Evaluations*, pp. 70–74, 2007.

[22] A. Balahur and A. Montoyo, "Applying a culture dependent emotion triggers database for text valence and emotion classification," *Procesamiento del lenguaje natural*, vol. 40, pp. 107–114, 2008.

[23] C. Strapparava and R. Mihalcea, "Learning to identify emotions in text," *SAC '08: Proceedings of the 2008 Association for Computational Linguistics Symposium on Applied Computing*, pp. 1556–1560, 2008.

[24] S. Kim and E. Hovy, "Identifying opinion holders for question answering in opinion texts," *Proc. of AAAI Workshop on Question Answering in Restricted Domains*, 2005.

[25] N. S. Glance, M. Hurst, and T. Tomokiyo, "Blogpulse: Automated trend discovery for weblogs," *WWW 2004 Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics*, 2004.

[26] M. Platakis, D. Kotsakos, and D. Gunopulos, "Discovering hot topics in the blogosphere," in *Proc. of the Panhellenic Scientific Student Conference on Informatics, Related Technologies and Applications EUREKA*, 2008, pp. 122–132.

[27] K. E. Gill, "How can we measure the influence of the blogosphere?" *WWW 2004 Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics*, 2004.

[28] A. Java, P. Kolari, T. Finin, and T. Oates, "Modeling the spread of influence on the blogosphere," *Proceedings of the 15th International World Wide Web Conference*, 2006.

[29] A. Kale, "Modeling trust and influence in the blogosphere using link polarity," *ICWSM '07: Proceedings of the International Conference on Weblogs and Social Media*, 2007.

[30] M. Hu and B. Liu, "Mining and summarizing customer reviews," *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 168–177, 2004.

[31] L. Zhao and C. Li, "Ontology based opinion mining for movie reviews," in *Proc. of KSEM*, 2009, pp. 204–214.

[32] P. Shvaiko, "Iterative schema based semantic matching," in *PhD-Thesis, International Doctorate School in Information and Communication Technology*, 2006, univ. Trento, Italia.

[33] S. Castano, V. De Antonellis, and S. De Capitani di Vimercati, "Global viewing of heterogeneous data sources," *IEEE Trans. on Knowl. and Data Eng.*, vol. 13, no. 2, pp. 277–297, 2001.

[34] H.-H. Do and E. Rahm, "COMA: A system for flexible combination of schema matching approaches," *VLDB '02: Proceedings of the 28th international conference on Very Large Data Bases*, pp. 610–621, 2002.

[35] J. Madhavan, P. A. Bernstein, and E. Rahm, "Generic Schema Matching with Cupid," *VLDB '01: Proceedings of the 27th International Conference on Very Large Data Bases*, pp. 49–58, 2001.

[36] S. Melnik, H. Garcia-Molina, and E. Rahm, "Similarity Flooding: A Versatile Graph Matching Algorithm and Its Application to Schema Matching," *ICDE '02: Proceedings of the 18th International Conference on Data Engineering*, p. 117, 2002.

[37] A. Doan, J. Madhavan, R. Dhamankar, P. Domingos, and A. Halevy, "Learning to match ontologies on the semantic web," *The VLDB Journal*, vol. 12, no. 4, pp. 303–319, 2003.

[38] M. Ehrig and S. Staab, "Qom - quick ontology matching," pp. 683–697, 2004.

[39] J. Euzenat, P. Gugan, and P. Valtchev, "OLA in the OAEI 2005 alignment contest," *Proceedings of the K-CAP Workshop on Integrating Ontologies*, pp. 61–71, 2005.

[40] F. Giunchiglia, P. Shvaiko, and M. Yatskevich, "S-match: an algorithm and an implementation of semantic matching," in *Proc. of ESWS 2004*, Y. Kalfoglou and et al., Eds. Springer, 2004, pp. 61–75.

[41] B. Le, R. Dieng-Kuntz, and F. Gandom, "On ontology matching problems - for building a corporate semantic web in a multi-communities organization," in *Proc. of the Sixth International Conference on Enterprise Information Systems*, no. 4, Abril 2004, pp. 236–243.

[42] N. Noy and M. Musen, "The PROMPT suite: interactive tools for ontology merging and mapping," *International Journal of Human-Computer Studies*, vol. 59, no. 6, pp. 983–1024, 2003.

[43] K. Kotis, G. Vouros, and K. Stergiou, "Capturing semantics towards automatic coordination of domain ontologies," in *the 11th International conference of Artificial Intelligence: Methodology, Systems, Architectures - Semantic Web Challenges - AIMSA 2004*. Springer-Verlag, 2004, pp. 22–32.

[44] P. Lambrix and H. Tan, "SAMBO-A system for aligning and merging biomedical ontologies," *Web Semant.*, vol. 4, no. 3, pp. 196–206, 2006.

[45] Y. Qu, W. Hu, and G. Cheng, "Constructing virtual documents for ontology matching," in *WWW '06: Proceedings of the 15th international conference on World Wide Web*, 2006, pp. 23–31.

[46] N. F. Noy, "Semantic integration: a survey of ontology-based approaches," *SIGMOD Rec.*, vol. 33, no. 4, pp. 65–70, 2004.

[47] E. Rahm and P. A. Bernstein, "A survey of approaches to automatic schema matching," *The VLDB Journal*, vol. 10, no. 4, pp. 334–350, 2001.

[48] C. Phytila, "An Analysis of the SUMO and Description in Unified Modeling Language," 2002, no publicado.

[49] S. Semy, M. Pulvermacher, and L. Obrst, "Toward the use of an upper ontology for U.S. government and U.S. military domains: An evaluation," in *Submission to Workshop on IIWeb*, 2004.

[50] A. Kiryakov, K. Simov, and M. M. Dimitrov, "Ontomap: portal for upper-level ontologies," in *Proc. of the FOIS*. ACM, 2001, pp. 47–58.

[51] P. Grenon, B. Smith, and L. Goldberg, "Biodynamic ontology: applying BFO in the biomedical domain," in *Ontologies in Medicine*, D. M. Pisanelli, Ed. IOS Press, 2004, pp. 20–38.

[52] D. Lenat and R. Guha, *Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[53] A. Gangemi, N. Guarino, C. Masolo, A. Oltramari, and L. Schneider, "Sweetening ontologies with DOLCE," in *Proc. of EKAW*. Springer, 2002, pp. 166–181.

[54] H. Herre, B. Heller, P. Burek, R. Hoehndorf, F. Loebe, and H. Michalek, "General formal ontology (GFO): A foundational ontology integrating objects and processes. Part I: Basic principles," Research Group Ontologies in Medicine (Onto-Med), Univ. Leipzig, Tech. Rep. Nr. 8, 2006.

[55] N. Casellas, M. Blzquez, A. Kiryakov, P. Casanovas, M. Poblet, and V. Benjamins, "OPJK into PROTON: Legal domain ontology integration into an upper-level ontology," in *Proc. of WORM 2005*. Springer, 2005, pp. 846–855.

[56] J. Sowa, *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks Cole Publishing, 2000.

[57] I. Niles and A. Pease, "Towards a standard upper ontology," in *FOIS '01: Proceedings of the international conference on Formal Ontology in Information Systems*. New York, NY, USA: ACM, 2001, pp. 2–9.

[58] V. Mascardi, V. Cord, and P. Rosso, "A comparison of upper ontologies," in *Atti del Workshop Dagli Oggentti agli Agenti, WOA*, M. Baldoni and et al., Eds. Seneca Editore, 2007, pp. 55–64.

[59] G. Stoilos, G. Stamou, and S. Kollias, "A string metric for ontology alignment," in *Proc. of the ISWC*, 2005, pp. 624–637.

[60] V. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707–710, 1966.

[61] V. Mascardi, A. Locoro, and P. Rosso, "Automatic ontology matching via upper ontologies: A systematic evaluation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 99, no. 1, 2009, doi: 10.1109/TKDE.2009.154.

[62] A. Locoro, "Ontology Matching using Upper Ontologies and Natural Language Processing," in *PhD-Thesis Course in Electronic and Computer Engineering, Robotics and Telecommunications*, 2010, univ. Genova, Italia.

[63] A. Utsumi, "A unified theory of irony and its computational formalization," *Proceedings of the 16th conference on Computational linguistics*, pp. 962–967, 1996.

[64] A. Reyes, P. Rosso, and D. Buscaldi, "Humor in the blogosphere: First clues for a verbal humor taxonomy," *Journal of Intelligent Systems*, vol. 18, no. 4, 2009.

Aprendizaje de Reglas Encadenas para la Creación de Grafos Conceptuales

Sonia Ordoñez Salinas

***Resumen*—El documento presenta una forma de aprendizaje sobre reglas encadenadas para la generación de nuevas reglas que al aplicarlas deberán permitir la construcción de Grafos Conceptuales. La propuesta se basa en la inclusión de reglas encadenadas y de de un método supervisado. Las reglas son definidas sobre la base de tres elementos: a)La marcación o rol que ocupa la palabra dentro de la oración, b)El estándar de Grafos Conceptuales y c) La definición de un Objeto que funciona como una caja de negra de Grafos. Las pruebas se realizaron sobre algunos de los textos correspondientes a los títulos y comentarios que hacen parte de la colección de imágenes médicas del ImageClefmed del 2008. Para la realización de las marcas se utilizó el metatesauro UMLS y la herramienta MMTx y para los procesos de clasificación se uso el Weka. Como resultado se estiman nuevas reglas.**

***Palabras clave*—Aprendizaje de reglas encadenas, UMLS, grafos conceptuales, anotación de imagen médica.**

Learning of Chained Rules for Construction of Conceptual Graphs

***Abstract*—El documento presenta una forma de aprendizaje sobre reglas encadenadas para la generación de nuevas reglas que al aplicarlas deberán permitir la construcción de Grafos Conceptuales. La propuesta se basa en la inclusión de reglas encadenadas y de de un método supervisado. Las reglas son definidas sobre la base de tres elementos: a)La marcación o rol que ocupa la palabra dentro de la oración, b)El estándar de Grafos Conceptuales y c) La definición de un Objeto que funciona como una caja de negra de Grafos. Las pruebas se realizaron sobre algunos de los textos correspondientes a los títulos y comentarios que hacen parte de la colección de imágenes médicas del ImageClefmed del 2008. Para la realización de las marcas se utilizó el metatesauro UMLS y la herramienta MMTx y para los procesos de clasificación se uso el Weka. Como resultado se estiman nuevas reglas.**

***Index terms*—Aprendizaje de reglas encadenas, UMLS, grafos conceptuales, anotación de imagen médica.**

I. Introduction

La necesidad de una comunicación efectiva entre el hombre y la máquina, de tal forma que una persona pueda expresarse en su Lenguaje Natural y la máquina pueda responderle en el mismo lenguaje es una necesidad cada vez más requerida. Ningún dominio se escapa a esta necesidad y en particular el dominio médico. Las personas involucradas con la asistencia médica generalmente realizan sus anotaciones en lenguaje natural y por más que exista software especializado y muy estandarizado, a la hora de expresar los síntomas, el diagnostico de un paciente o la anotación sobre una imagen médica, el personal médico simplemente lo digita en su lenguaje natural.

A pesar de que existen herramientas especializadas como las ontologías y los tesauros que ayudan a comprender algunos de los términos digitados como nombres de enfermedades y algunas relaciones entre conceptos, es claro que poder procesar el lenguaje natural por el computador es una tarea que no es fácil. Si bien para el ser humano, ante una frase, inmediatamente no solo la relaciona con todo su conocimiento previamente adquirido sino que es capaz de filtrar todo ese conocimiento, para ubicar la frase en el entorno o dominio adecuado y dejar exactamente lo que necesita para saber de qué se trata. Los procesos utilizados con la máquina o el computador no pueden hacer esto tan inmediato, ni con tan buen resultado, ante una frase solo pueden ubicar una cadena de símbolos que no significan nada, por lo que se debe recurrir a ontologías o tesauros y consultar que pueden significar estos símbolos. Por otro lado, para poder individualizar un símbolo o un conjunto de estos y sus relaciones se requiere que sean representados dentro de estructuras más estándares y adecuadas para el computador. Adecuadas en el sentido que se pueda gestionar uno o varios de estos símbolos, es decir, almacenarlos, consultarlos y eliminarlos.

Existe una gran variedad de estructuras que se utilizan para realizar la tarea de procesar el texto a través de un computador, unas con mayores ventajas que otras. Sin embargo todas cuentan con la dificultad de requerir pasar los símbolos y/o conocimiento (cuando el símbolo ya tiene algún significado) a dicha estructura. Generalmente se utiliza la ayuda de un experto para representar el texto expresado en lenguaje natural en una de estas estructuras.

Dentro de estas estructuras se encuentran Los Grafos Conceptuales que a diferencia de las demás estructuras, permiten incluir detalles de la semántica propia del texto escrito en Lenguaje Natural y cuentan con algunas características del lenguaje matemático.

Este trabajo se ocupa en forma particular, de cómo automáticamente generar nuevas reglas a partir de un conjunto suministrado de reglas encadenadas. La idea de contar con un conjunto de reglas que permitan la consecución de manera

Manuscrito recibido el 15 de abril del 2010. Manuscrito aceptado para su publicación el 14 de junio del 2010.

La autora trabaja en Universidad Distrital F.J.C, Colombia y estudia en Universidad Nacional de Colombia (e-mail: sordonez@udistrital.edu.co).

automática de Grafos Conceptuales a partir de un texto expresado en lenguaje Natural.

La temática de cómo crear grafos a partir de un texto, se ha tratado de diferentes enfoques desde aquellos que implican la consecución de cada uno de los elementos de manera secuencial hasta aquellas que involucran procesos estadísticos. A pesar de las diferentes propuestas en torno al fin planteado y de acuerdo al estado del arte, se puede observar que cualquier esfuerzo encaminado a dar nuevas alternativas aún es válido.

Este documento, presenta la propuesta de encontrar reglas a partir de reglas encadenas por medio un método supervisado. Las reglas encadenadas al ser aplicadas sobre un texto expresado en lenguaje natural, dan como resultado la representación de dicho texto en una estructura de Grafo Conceptual. Las reglas encadenadas son encontradas con base en tres elementos: a) el rol que ocupa una palabra dentro la frase b) las estructuras básicas de los Grafos Conceptuales y c) la definición de un *Objeto* que funciona como una caja negra de grafos pero que da la posibilidad de inferir las reglas inmersas dentro de esta caja negra. Los textos son extractados de las anotaciones que hacen parte de la colección de imágenes médicas del ImageClefMed del 2008.

Este documento se organiza de la siguiente forma: en la sección II se definen los conceptos básicos incluyendo los lineamientos generales sobre el tema tratado, permitiendo ubicar al lector. En la sección III se incluye una revisión del estado del arte. En la sección IV se presenta el método utilizado; en V se muestra el desarrollo experimental. En VI se presentan las conclusiones y trabajo futuro.

II. Conceptos Básicos

El dominio de la medicina cuenta con herramientas muy sofisticadas para ayudar a los sistemas computacionales con la terminológica propia del dominio. Dentro de estas ayudas especializadas se encuentra el Metatesauro "*Unified Medical Language System (UMLS) of the National Mibrary of Medicine of the National Institute of Health of United States*" [20], desarrollado en el 2003. Dicha herramienta consta de una base de datos multilingüe que relaciona más de un millón de conceptos biomédicos, de la salud y léxicos, incluidos en más de 100 fuentes de información. Cuenta además, con una serie de programas especializados para manejar dicha información. Dentro de la herramientas que hace parte de la colección está el MetaMap Transfer (MMtx) [13] que permite entre otras funciones, relacionar términos del metatesauro UMLS y hacer marcación del rol que ocupa una palabra dentro de la Oración (*Part-of-speech tag*). Esta herramienta a diferencia de otras de su categoría es que está hecha específicamente para que funcione con el metatesauro UMLS y que por ende, reconoce las palabras propias de las ciencias biomédicas y de la salud. Es así, que no solo reconoce una palabra o conjunto de palabras sino que permite etiquetar estas palabras.

A pesar de la existencia de herramientas como el UMLS aún se hace necesario la utilización de otras técnicas que permitan suplir las necesidades relacionadas con el análisis del lenguaje y que son propias de la actividad del personal médico y hospitalario.

El Procesamiento del Lenguaje Natural (PLN) es una disciplina encargada de la gestión del Lenguaje Natural (LN). Entendida como gestión la extracción de información a partir de un texto expresado en lenguaje natural y su procesamiento [8]. El proceso incluye desde almacenar e indexar la información hasta la búsqueda y la consulta de un requerimiento en particular. Dicho procesamiento es posible gracias a una serie de artificios o estructuras más acordes para el ámbito computacional que un texto escrito en lenguaje natural. Dentro de estas estructuras se mencionan algunas a continuación.

A. *Estructuras Matemáticas*

Las estructuras matemáticas son las más utilizadas para el procesamiento del lenguaje natural. Dentro de estas se pueden referenciar las estructuras vectoriales que pueden ser utilizadas con números o palabras. Trabajar con estas estructuras es muy fácil, sin embargo, tienen la desventaja que al representar un texto escrito en LN en dichas estructuras permiten que se pierdan muchos de los detalles propios de la semántica inmersa dentro del lenguaje natural. Se puede, por ejemplo relacionar el trabajo presentado en [6], que utiliza los vectores y el presentado en [18], que utiliza una estructura más compleja como son los grafos. En esta categoría se pueden incluir la clásica estructura para modelamiento de bases de datos y conocida como Modelo Entidad Relación y el modelo de clases utilizado para el diseño de aplicaciones de software.

B. *Lenguajes Estructurados*

Utilizados especialmente para describir los pasos y procedimientos que debe incluir un programa de software o los conceptos y funcionalidades propias de una ontología.

Los lenguajes clasificados bajo esta categoría se definen como metalenguajes orientados al reconocimiento y descripción de objetos y clases. Dentro de esta categoría se encuentran la mayoría de lenguajes de programación orientados a objetos, donde la unidad primaria de organización es un marco que tiene características como el nombre y los atributos. Un atributo igualmente tiene su nombre y puede contener valores.

C. *Estructuras Gramaticales*

Las estructuras gramaticales permiten incluir la información gramatical producto de los analizadores gramaticales y generalmente es expresada en forma de árboles y reglas gramaticales. Dentro de estas estructuras estas las reglas utilizadas para expresar las gramáticas libres de contexto.

D. *Estructuras Conceptuales*

Las estructuras propias de esta categoría, a diferencia de las anteriores permiten incluir una mayor riqueza semántica.

Dentro de estas estructuras se pueden catalogar, las redes semánticas [13], los lenguajes para descripción e intercambio de conocimiento como el *Knowledge Interchange Format* (KIF), el *Resource Description Framework* (RDF), el *web ontology language* (OWL) y los Grafos Conceptuales [12]. Estos últimos fueron diseñados para que su representación incluyera a demás detalles propios de la semántica, algunas de las bondades propias de las matemáticas.

III. Estado del Arte

En esta sección se mencionan las características principales de los Grafos Conceptuales y se relacionan algunos de los trabajos sobre cómo de manera automática representar un texto escrito en el Lenguaje Natural en un Grafo Conceptual. Dentro de los trabajos se presentan algunos que se enfocan a la parte procedimental o metodológica, a procesos puramente secuenciales y aquellos que de alguna forma utilizan aprendizaje maquinal.

A. Grafos Conceptuales

Los grafos conceptuales son definidos por Sowa [19] con base en la lógica de primer orden y los grafos existenciales de Pierce [6]. El estándar incluye dos tipos de nodos (conceptos y relaciones), aristas y expresiones de tipo lógico. Por sus características los Grafos Conceptuales no solo, permiten incluir detalles propios del lenguaje natural, sino que soportan operaciones básicas como la unión, la intersección y la inferencia. El grafo correspondiente a *"La radiografía enseña un corazón crecido"* se puede visualizar en la Figura 1.

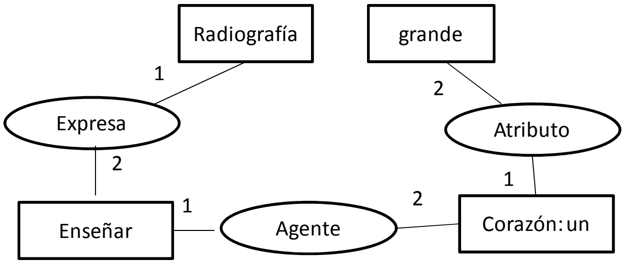


Fig. 1. Ejemplo de un Grafo Conceptual.

B. Representación del Lenguaje Natural en Grafos Conceptuales

En cuanto a la forma cómo pasar un texto expresado en lenguaje natural a una estructura de Grafo Conceptual se han propuesto diferentes métodos, entre los que siguen pasos secuenciales y procedimentales para la consecución de cada uno de los elementos propios de los Grafos Conceptuales y aquellos que incluyen procesos de aprendizaje automático.

1) Métodos Secuenciales y Procedimentales. Dentro de los trabajos que se enfocan a la parte procedimental o metodológica se pueden citar el presentado [17] que es puramente procedimental y el [15] que presenta la creación de los Grafos Conceptuales desde el punto de vista de modelamiento remedando los lineamientos que se siguen al crear un modelo de clases.

Como trabajo que sigue una serie de pasos de manera secuencial se encuentra el [5] que presenta un sistema que convierte un texto en español a Grafos Conceptuales, utilizando un analizador morfológico (véase, por ejemplo, [8]), un analizador sintáctico e identifica cada uno de los elementos de los grafos.

2) Métodos que Involucran Tareas de Aprendizaje Automático. En el trabajo presentado en [1] se obtienen los Grafos Conceptuales a partir de textos en francés basado en algunas estructuras y el análisis sobre los verbos.

Sobre el dominio específico de la medicina se encuentran trabajos como el presentado en [16], donde se busca construir una estructura de reportes basada en Grafos Conceptuales. Con tal fin, se analizan las frases a partir de reglas de proximidad que tienen en cuenta la información semántica y sintáctica obteniendo un texto anotado. Con base a dicho análisis se consiguen los componentes que permiten construir los Grafos Conceptuales.

En el propuesto en [19], se construye un sistema que se enmarca en las radiologías y se representa el contenido de las anotaciones de las radiologías en Grafos Conceptuales.

El trabajo presentado en [4], con el objeto de lograr un mejor desempeño a la hora de realizar consultas y recuperación, presenta algunos aspectos de la transcripción de registros de asistencia médica a una estructura de Grafos Conceptuales. El trabajo se centra en el modelamiento de una base de conocimiento utilizando Grafos Conceptuales, muy similar a como se modela bajo el paradigma *entidad-relación*.

Utilizando reglas sintácticas y semánticas se puede citar el trabajo presentando en [10] y el planteado en [2] que utilizan un analizador gramatical para construir árboles gramaticales y a partir de una serie de reglas transforman dichos árboles en grafos conceptuales. Una vez transformados se optimizan los grafos al desambiguarlos estructural y semánticamente.

En la investigación presentada en [23], construyen los Grafos Conceptuales con base en gramáticas de enlace y máquinas de aprendizaje como un problema de clasificación que puede ser entrenado para diferentes dominios.

IV. Método Utilizado

Con el fin de conseguir de manera automática los Grafos Conceptuales, se propone el aprendizaje maquinal a través de un conjunto de reglas encadenadas. Cada clase de regla aplicada sobre un texto fue etiquetada obteniendo al final una cadena de reglas encadenas. La idea es que, al aplicar un conjunto de reglas de forma encadenada y sobre un texto particular se obtiene el Grafo Conceptual correspondiente a dicho texto

A. Colección Experimental

La colección de entrenamiento y prueba se conforman a partir de un conjunto de texto en lenguaje natural y relacionado con anotaciones de imágenes médicas de la colección de ImageClefMed del 2008. Los textos fueron extractados tanto del título como de la anotación de la imagen.

De algunos de los textos de esta colección y utilizando el software MetaMap Transfer (MMTx) se etiquetaron los textos con las marcas correspondientes a la función que juega cada palabra dentro de la oración (ver Tabla I).

TABLA I
Ejemplo de las Etiquetas Utilizadas por MMTx.

Etiqueta de MMTx	Significado
noun	Sustantivo
adj	Adjetivo
prep	Preposicion
det	Determinante
aux	auxiliar
ver	Verbo

A esta colección se le realizó un preprocesamiento previo con el fin de facilitar las tareas siguientes. Dentro de este preprocesamiento se resolvió la correferenciación y se asumió:

Sustantivo: un sustantivo equivalente a un conjunto de sustantivos consecutivos. Esto es, que en el caso de encontrar más de dos sustantivos consecutivos se asumen como si fuera un único sustantivo. Lo anterior permitió definir las entidades nombras, como "*Seymour H. Levit, MD Radiology Society of North America*" (en este caso la herramienta MMTx detectó "*Society of North America*")*".*

Concepto Básico (CB): Un sustantivo con características definidas por el estándar de Grafo Conceptual (ver Tabla II).

En el Lenguaje Natural, una idea se puede expresar de diferente forma y para el raciocinio del ser humano dicha idea significa lo mismo. Para el ámbito computacional, esto no es tan simple. Existe la técnica computacional llamada Parafraseo que consiste en cambiar una frase por otra sin alterar su significado. Este cambio se puede dar en dos sentidos, el primero se cambia algunos términos por sinónimos y en el segundo se cambia totalmente la forma de la oración. En esta última forma, se pueden cambiar todas o algunas palabras y el orden de las mismas, por ejemplo pasar una oración de forma activa a su forma pasiva. Por lo que encontrar si dos oraciones son similares se debe aplicar la técnica de parafraseo a una de ellas con el fin de ver si de alguna forma se llega a la otra.

Los grafos conceptuales como una estructura estándar busca representar algunas ideas de manera estándar o que una idea siempre se represente de igual forma. Es así, con el caso de la presentación de un sustantivo, incluido la categoría del sustantivo y su individualización. Por ejemplo la frase "Paul is a child", puede escribirse de varias formas entre ellas: "Paul who is a child", "the child is Paul", "the child name is Paul". Para los Grafos Conceptuales, la idea se representa a través del nodo concepto [child:Paul]. Para esta estandarización un nodo concepto debe estar formado por mínimo la categoría. Esto es, que si no se conoce el nombre del niño el grafo sería [child]. Pero si solamente se conoce que es una persona, animal o cosa se llama Paul, el concepto no se puede representar como [Paul], ya que es completamente ambiguo, por lo que se debe incluir la categoría y esto requiere de un proceso de búsqueda el cual no es objeto del presente trabajo.

Para resolver la categoría se definió la categoría "nn" permitiendo representar un concepto de la forma [nn:Paul].

TABLA II
Ejemplo de GC de tipo Concepto Básico.

Paul is a child	Notacion GC
That child	[niño:Paul]
Some child	[niño:#That]
Every child	[niño:{*}]
Ther exist a child	[niño:□]
One cat	[cat:@'one]

Objeto (Obj): Un elemento que puede ser concepto o el resultado de aplicar una regla y que no es otra cosa que un uno o varios Grafos Conceptuales compuestos por nodos concepto, nodos relaciones y aristas. Este Objeto se define para que funcione como una caja negra que incluye un grafo o n-grafos y sobre la cual se puede aplicar una función que da como resultado otra caja negra.

Regla: Las Reglas se definen bajo los estándares de los Grafos Conceptuales. Al aplicarlas equivalen a las relaciones entre los nodos. De acuerdo al estándar de los grafos conceptuales y siguiendo con la idea del parafraseo, se busca que estandariza hasta donde sea posible la representación de varias ideas. Es el caso de los verbos, donde algunos emiten la idea de una percepción, entonces, todos las ideas que incluyan una percepción siempre se deberá representar de igual forma. Y entonces, se puede definir una regla específica para cuando esté incluido uno de estos verbos. Estas reglas se definieron de acuerdo a las posibles estructuras definidas por las etiquetas y cada uno de los roles de las palabras dentro de la frase de acuerdo al estándar de los GC. (ver Tabla III).

A partir de las reglas así definidas, la consecución de un grafo, corresponde a la aplicación de reglas de manera encadenada sobre el texto. Esto se puede visualizar mejor en el siguiente ejemplo:

Frase:
"*Illustration of a neonate at autopsy whose demise was attributed to thymic death.*"

Reglas:
1)Attr([Illustration], neonate)
2)Atrr([death],thymic)
3)In (Atrr([caption], neonate),autopsy)
4)Efect([demise], was attributed)
5)Rslt([Efect([demise], was ttributed)],[Atrr([death],thymic)])
6)Rslt(In (Atrr([caption], neonate),autopsy),Rslt([Efect([demise], was attributed)],[Atrr([death],thymic)]))

Por último se etiquetaron las reglas y se encontró la frecuencia de aplicación de cada regla sobre una frase. Así, en el ejemplo anterior, la regla 5 se aplicó 2 veces.

Teniendo en cuenta las reglas aplicadas en un texto, esta estará representada por un vector de frecuencias donde un valor *fx* significa el número de veces que se aplicó la regla *f*.

TABLA III
EJEMPLO DE REGLAS UTILIZADAS.

Regla	Condición	Acción o Función
1	Más de un nodo del mismo concepto	Ref(Obj,obj)
5	(adj, obj) **OR** (adj prep obj)	Attr(obj,adj)
5a	(obj prep det obj) **OR** (obj prep obj) **OR** (obj prep det obj) **OR** (obj with obj)	Atrr(obj,obj)
6	(obj a_prep obj) **OR** (obj in_prep an_det obj) **OR** (obj in_prep obj)	In(obj,obj)

B. Bigramas

Con el fin de relacionar las reglas con las etiquetas del lenguaje dentro de la oración se encontraron los bigramas de las marcas correspondientes al rol del lenguaje dentro de la oración. Se utilizaron bigramas dado que estos proveen una mayor cantidad de información al combinar las marcas consecutivas. Y la representación de cierto bigrama dentro de la frase esta dado por la presencia o ausencia del mismo.

Finalmente un texto expresado en lenguaje natural se representa por la unión de la representación de bigramas con la representación de reglas. Esto es:

RepresentacionTexto =
{ RepresentaciónBigramas, RepresentaciónReglas}

C. Función de Clasificación

Con el fin de encontrar una forma de aprendizaje sobre dichas reglas, se creó una función de clasificación basada en la sumatoria de las frecuencias de las reglas aplicadas Σf_x.

V. DESARROLLO EXPERIMENTAL

Con el objeto de obtener un mayor aprendizaje sobre las reglas, se realizaron pruebas a nivel de función o regla y a nivel de todo el conjunto de reglas.

Con base en la matriz adecuada y el software *Weka [22]* se utilizó un método de árbol de clasificación con el fin de aprender sobre las reglas ya previamente definidas. Los algoritmos para árboles de clasificación se basan en la cantidad de información mutua que puede darse entre una variable predictiva y su clase. Dentro de los algoritmos para árboles de clasificación más populares se encuentra el ID3 y el C4.5 [3]. Ambos métodos se basan en la cantidad de información mutua, el ID3 favorece las variables con mayor número de valores, mientas que el C4.5, corrige dicha ponderación del ID3 y realiza una poda que consiste en la aplicación de una prueba estadística para saber si se debe expandir o no una rama. El algoritmo "*RandonTree*" utilizado en el Software de Weka, construye un árbol partiendo de un número aleatorios de atributos para cada nodo. No realiza poda, pero utiliza un método de valoración de clase de acuerdo a las probabilidades.

En todos los casos de clasificación a la colección se le dio un tratamiento de validación por cruce con un particionamiento de 10, es decir "*10 fold cross validation*"

Se aplicaron los clasificadores tanto a cada regla de manera independiente como en conjunto a todas las reglas. Es decir, que se hicieron dos clases de experimentaciones, una donde la función de clase corresponde a la aplicación de una sola regla *f* y otra en donde la función de clase corresponde a la aplicación de todas las reglas.

A. Resultados Experimentales por Regla

La experimentación se aplicó a cada una de las reglas deducidas. Sin embargo en el documento se incluye únicamente el análisis de dos de ellas.

1) *Regla Atributo (Attr):* Para la regla 5 (5 y 5a) que corresponde a la aplicación de una regla *Attr* y que al aplicarla conduce según el estándar de Grafos Conceptuales al establecimiento de una relación (nodo relación de tipo Atributo) entre dos conceptos (nodos concepto). De acuerdo al árbol (ver Fig. 2) generado para esta regla se puedo observar que efectivamente tiene en cuenta la aparición de un adjetivo por lo que en el nodo inicial aparece el adjetivo.

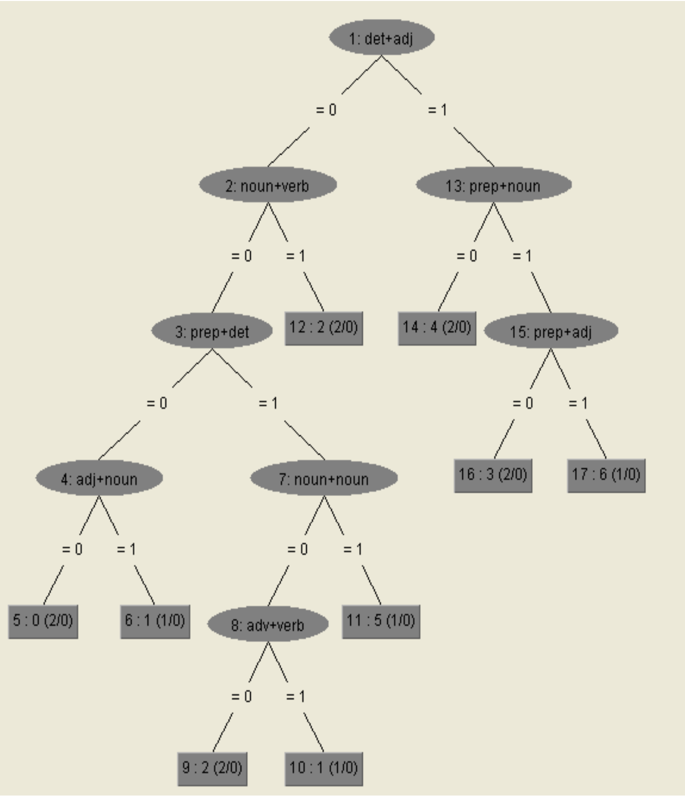


Fig. 2. Árbol generado por la regla Attr.

Con base en la (Fig 2) se pueden extractar las siguientes reglas:

a) Regla 1
determinante+adjetivo+preposición+sustantivo+preposición+ adjetivo.

b) Regla 2
preposición+determinante+sustantivo+sustantivo

c) Regla 3
preposición+adjetivo
d) Regla 4
sustantivo+preposición+adjetivo
e) Regla 5
preposición+sustantivo+preposición+adjetivo
f) Regla 6
adjetivo+reposición+sustantivo+preposición+adjetivo
g) Regla 7
preposición+determinante+sustantivo

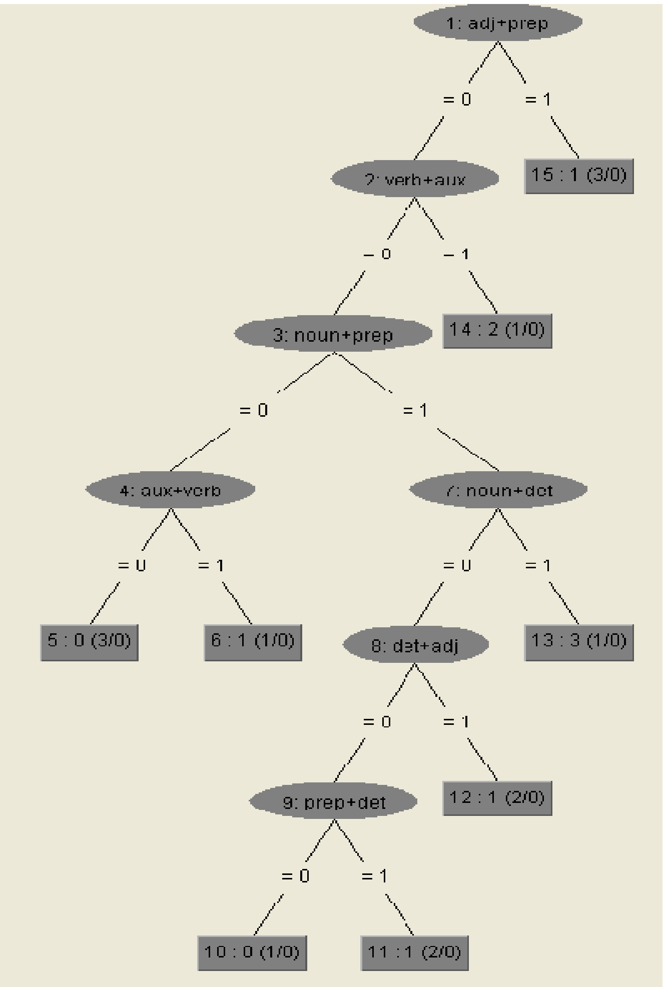


Fig. 3.Árbol Generado por la regla Rslt.

Dentro de los supuestos que se han hecho para resolver el problema está la definición del “Objeto” que se comporta como una caja negra que contiene grafos. Con el método aplicado de lo que se trata es de extractar las reglas de cada una de estas cajas. Y es así que el método ha generado nuevas reglas o ha extractado las reglas de esto objetos.

Por otro lado, al revisar las estadísticas correspondientes a la clasificación se puede observar que un 50% de los casos quedan bien clasificados.

2) *Regla Resultado (Rslt):* Esta regla define la existencia de una relación de *resultado (Rslt)*. Un concepto es el resultado de otro, aún cuando dicho resultado no necesariamente se presenta por la aplicación de un verbo. De acuerdo al árbol generado (ver Fig. 3) esta regla se podría aplicar en presencia de las combinaciones de:

a) Regla 1.
sustantivo+preposición+sustantivo+determinante.
b) Regla 2
sustantivo+determinante
c) Regla 3
preposición+sustantivo+determinante

De acuerdo a las estadísticas de clasificación, se obtuvo un 46% de casos bien clasificados.

De igual forma se realizó la clasificación para otras reglas obteniendo nuevas posibles combinaciones de marcaciones del idioma.

B. Resultados Experimentales aplicando todas las Reglas

Por último se aplicó el clasificador donde la función de clasificación corresponde a la aplicación de todas las reglas. Como resultado se obtuvieron las combinaciones que aparecen en el árbol de la Fig. 4.

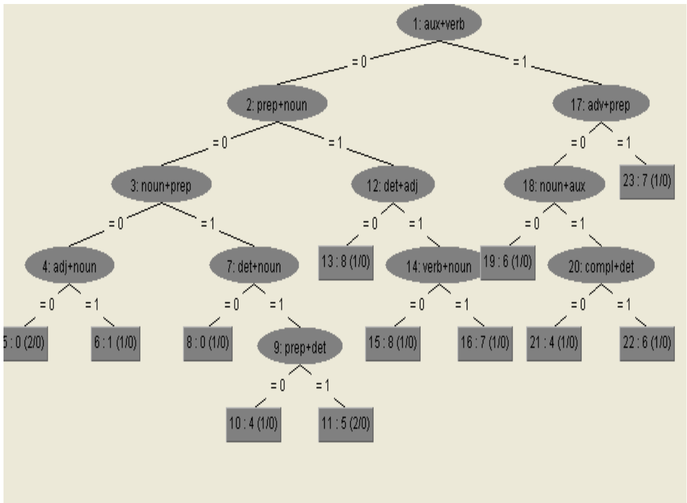


Fig. 4. Árbol Generado por la aplicación de todas las reglas.

a) Regla 1
proposición+sustantivo determinanate+adjetivo+verbo+sustantivo.
a) Regla 2
sustantivo+determinanate+adjetivo+verbo+sustantivo.
b) Regla 3
determinanate+adjetivo+verbo+sustantivo.
c) Regla 4
adjetivo+verbo+sustantivo.
d) Regla 5
verbo+sustantivo.
e) Regla 6
sustantivo+preposición+determinante+sustantivo+preposición+determinante
f) Regla 7
preposición+determinante+sustantivo+preposición+determinante
g) Regla 8

determinante+sustantivo+preposición+determinante

h) Regla 9

sustantivo+preposición+determinante

i) Regla 10

preposición+determinante

j) Regla 11

preposición+determinante+sustantivo+preposición

k) Regla 12

auxiliar+verbo+adverbio+preposición

l) Regla 13

verbo+adverbio+preposición

m) Regla 14

adverbio+preposición

De acuerdo a los anteriores resultados se pude observar que efectivamente con el método propuesto si se pueden aprender nuevas reglas.

Por último se puede verificar que de acuerdo a las estadísticas del clasificador se obtuvieron resultados aceptables, con un 45% de instancias bien clasificadas.

VI. Conclusiones y Trabajo Futuro

De acuerdo a los resultados obtenidos se puede concluir que se pueden deducir reglas a partir de las reglas encadenadas previamente definidas y con la definición del Objeto como artificio estructural. En un posterior trabajo se deben buscar mecanismos que permitan evaluar si la regla generada es válida para el estándar de los Grafos Conceptuales. En este sentido sería deseable hacer las pruebas necesarias para el comportamiento con trigramas e incluir técnicas que permitan tener en cuenta el orden de aplicación de las funciones.

Agradecimientos

Este documento se desarrolla como resultado de la estancia realizada en el Laboratorio de Lenguaje Natural del Centro Investigación en Computación del Instituto Politécnico Nacional de México. La estancia ha sido parcialmente financiada por la Universidad Nacional de Colombia y la Universidad Distrital F.J de C.

Referencias

[1] T. Amghar, D. Battistelli, and T. Charnois, "Reasoning on aspectual temporal information in French within conceptual graphs," in *Proc. 14th IEEE International Conf. Tools with Artificial Intelligence (ICTAI 02)*, Washington DC, 2002, pp. 315-322.

[2] C.-Barrière and N. C. Barrière, *From a Children's First Dictionary to a Lexical Knowledge Base of Conceptual Graphs*. St. Leonard's (NSW): Macquarie Library, 1997.

[3] Building Classification Models: ID3 and C4.5. Available: http://www.cis.temple.edu/~ingargio/cis587/readings/id3-c45.html.

[4] Chih-Shyang Chang and Arbee L. P. Chen, "Supporting Conceptual and Neighborhood Queries on WWW," *IEEE Trans. on Systems, Man, and Cybernetics, Part C: application and reviews,* Vol. 28, no. 2, pp. 300-308, 1988.

[5] M. Hernandez Cruz, "Generador de los grafos conceptuales a partir del texto en español," Tesis de Maestría. Instituto Politécnico Nacional. Centro de Investigación en computación, 2007.

[6] J. Farkas, "Improving the classification accuracy of automatic text processing systems using context vectors and back-propagation algorithms," in *Proc. Canadian Conference on Electrical and Computer Engineering,* University of Calgary, Alberta, 1996, pp. 696-699.

[7] G. Allwein and J. Barwise, *Logical Reasoning with Diagrams.* New York, Oxford University Press, 1996.

[8] A. Gelbukh and G. Sidorov, *Procesamiento automático del español con enfoque en recursos léxicos grandes*, Mexico, IPN, 2006.

[9] A. Gelbukh and G. Sidorov, "Approach to construction of automatic morphological analysis systems for inflective languages with little effort", *Lecture Notes in Computer Science, N 2588,* Springer-Verlag, pp. 215–220, 2003.

[10] S. Hensman, "Construction of Conceptual Graph representation of texts," in *Proc. of Student Research Workshop at HLT-NAACL,* Department of Computer Science, University College Dublin, Belfield, Dublin, 2004.

[11] Medical Image Retrieval Challenge Evaluation. Available: http://ir.ohsu.edu/image/2008protocol.html.

[12] S. Kamaruddin, A. Bakar, A. Hamdan, and F. Nor, "Conceptual graph formalism for financial text representation" in *International Symposium Information Technology,* Kuala Lumpur, Aug. 26-28, 2008, pp 1-6.

[13] M. Last and O. Maimon, "A Compact and Accurate Model for Classification," *IEEE Trans. on Knowledge and Engineering,* Vol.16, pp. 203-215, Feb 2004.

[14] MetaMap Transfer (MMTx). Available: http://ii-public.nlm.nih.gov/MMTx/.

[15] G. W. Mineau, G. Stumme and R. Wille, "Conceptual Structures Represented by Conceptual Graphs and Formal Concept Analysis," in *Proc. of 7th Int. Conf. on Conceptual Structures (ICCS-99),* W. Tepfenhart and W. Cyre (Eds), Springer-Verlag, 1999, pp. 423-441.

[16] A. M. Rassinoux, R. H. Baud, C. Lovis, J. C. Wagner, and J. R. Scherrer, "Tuning Up Conceptual Graph Representation for Multilingual Natural Language Processing in Medicine Conceptual Structures: Theory, Tools, and Applications," in *Proc. 6th International Conference on Conceptual Structures, ICCS'98*, Montpellier, France, 1998, pp. 334-350.

[17] G. Salton and M. Lesk. "The SMART automatic document retrieval systems: an illustration," *Comm. ACM,* 8 (6), pp. 391-398, Jun. 1965.

[18] A. Schenker and H. Bunke, *Graph-Theoretic Techniques for Web Content Mining,* World Scientific Publishing, 2005.

[19] M. Schröder, "Knowledge based analysis of radiology reports using conceptual graphs," *Lecture Notes in Computer Science,* Vol. 754 (Conceptual Structures: Theory and Implementation), Springer Berlin, pp. 293-302, 1993.

[20] J. F. Sowa, "Semantics of conceptual graphs," in *Proceedings of the 17th Annual Meeting on Association For Computational Linguistics,* La Jolla, California, July 1979, pp. 39-44.

[21] Unified Medical Language System (UMLS). Available: http://www.nlm.nih.gov/research/umls/about_umls.html.

[22] Weka software of The University of Waikato, Available: http://www.cs.waikato.ac.nz/~ml/weka/.

[23] R. A. Williams, "Computational Effective Document Semantic Representation," in *Proc. of IEEE-IES Digital Eco Systems and Technologies Conf. DEST'07,* 2007.

On a Framework for Complex and ad hoc Event Management over Distributed Systems

Genoveva Vargas-Solar, Paolo Bucciol, and Christine Collet

***Abstract*—Internet-based communications have amazingly evolved in recent years. As a consequence, the number – and complexity – of distributed systems which provide access to services and applications has dramatically increased. As long as these services have been extended to support an increasing number of communication media (voice, audio, video, ...) and systems, ad hoc communication protocols and methodologies have been designed and developed. Given the autonomy of available services and applications, distributed systems generally rely on event-based communications for integrating these resources. However, a general model for the management of event-based communications, suitable for complex and ad hoc event processing as well as for the generic publish/subscribe messaging paradigm, is still missing. This paper presents[1] a general and flexible event detection and processing framework which can be adapted based on specific requirements and situations. Within the framework, the main aspects of event management over distributed systems are treated, such as event definition, detection, production, notification and history management. Other aspects such as event composition, are also discussed. The goal of the paper is to provide a common paradigm for event-based communications, providing at the same time new advantages with respect to the existing standards such as composition, interoperability and dynamic adaptability.**

***Index Terms*—Event management, modeling, distributed systems, interoperability, adaptability.**

I. Introduction

THE "fully connected world" is perhaps the most remarkable step in the evolution of the communication system in the last century. Through the Internet, each user can virtually communicate *events* almost instantaneously with any other user. The concept of *event* is of major importance in the communications field, since it provides an enough general abstraction layer through which dynamic aspects of applications can be modeled. Events produced in the real world are converted into a sequence of bits, sent through one or more networks to reach their destination and then elaborated and presented following the consumer's indications. Events are well fitted to represent the dynamic aspects of applications and distributed characteristics of systems: the sequence of their execution, the state of a given process or of a data structure, the communication between entities. Examples of events range from the evolution of data (*the price of Euro has varied in the United States*), to the change in the execution state of a given process (*a web page has been modified*), to the communication and interaction between its components (*a print request has been performed*).

A vast number of event management models and systems has been, and continues to be, proposed [1]–[3]. Several standardization efforts are being made to specify how entities can export the structure and contents of the events [4]. Models proposed in the literature range from simple models which describe the notification of signals to evolved models which take into account various policies to manage the events. Existing models have been defined in an *ad hoc* way, notably linked to the utilization context (active DBMS event models), or in a very general way in middleware (Java event service, MOMs). Of course, customizing solutions prevents systems to be affected with the heavy weight of an event model way too sophisticated for their needs. However, they are not adapted when the systems evolve, cooperate and scale, leading to a lack of adaptability and flexibility.

The event management framework proposed in this paper, based on an in-depth event characterization, is targeted to the definition of a common event management model.

Through the definition of the conceptual mechanisms and the general semantics for the integration of distributed systems and heterogeneous networks, the proposed model aims at providing modular, adaptable and extensible mechanisms, which are well adapted when building applications and systems.

A list of *dimensions* is proposed [2], that characterize the *detection*, the *production*, the *notification* of events and the *history management* respectively. The proposed dimensions allow to: (i) highlight the aspects inherent to the management of events; (ii) propose a meta-model of event management to give a general characterization of management models independently of their applicative context [1].

The remainder of this paper is organized as follows. The generic structure of event-based systems is presented in Section II, while more specific aspects concerning events are described in Section III (event definition), Section IV

Manuscript received March 2, 2010. Manuscript accepted for publication June 14, 2010.

Genoveva Vargas-Solar and Christine Collet are with CNRS UMI 3175, French-Mexican Laboratory of Informatics and Automatic Control, Grenoble Institute of Tecnology and with Laboratory of Informatics of Grenoble, France (Genoveva.Vargas@imag.fr, Christine.Collet@imag.fr).

Paolo Bucciol is with CNRS UMI 3175, French-Mexican Laboratory of Informatics and Automatic Control, Grenoble Institute of Tecnology, France (paolo.bucciol@gmail.com).

[1] This work was partially supported by an external research project financed by Orange Labs, France.

[2] In this context, the meaning of dimension is "significant aspect of a thing" rather than its geometric meaning.

(event detection), Section V (event production), Section VI (event notification), and Section VII (event history). A generic event service which supports event management in distributed systems is introduced in Section VIII. Finally, conclusions and future work are presented in Section IX.

II. Event-based Systems

Event-based communication, based on a cause-effect principle, provides the basis for anonymous and asynchronous communication. The *event*, the cause, represents the change in the state of a system which leads to the production of a message. On the other hand, the *reaction*, the effect, corresponds to a set of reactions within the system, which react to the cause and can lead to the production of more events. This principle allows to model the evolution of a distributed system based on events asynchronously detected.

An event is modeled as an object of a specific type. The type is specified by the programmer, as, for instance, an object of type "event". The production environment is represented by the object attributes and methods, however finer categorization may be defined by the programmer.

An event channel is an object which allows multiple producers to communicate with multiple consumers in an asynchronous way. The event channel, at the same time producer and consumer of events, can notify changes related to a certain object. In this sense, it acts as an intermediary between objects which have been modified and an object interested (or involved) in such changes. When a change has taken place, an event can be notified to all the objects interested. In this vision, the underlying communication channel and network (e.g. multi-hop network) is transparent to the event channel.

Producers and consumers can communicate through the event channel by means of the *push and pull* model. The producer notifies some events to the channel; then, the channel notifies the events to the consumers. The consumer consumes the events through the event channel, which, in turn, detects them from the producer. The decoupled communication between producers and consumers is highly relevant in contexts where consumers would not be able to receive and interpret the messages, which is, for instance, the case of sensor networks. When a sensor needs to be awake to receive the events in real time (*on* status), it stays continuously connected to the network and thus wastes significant amounts of energy. This communicating strategy is not feasible in the common situation where sensors have strong computational and power constraints [5]. Event channels are therefore of deep importance to sensor networks, in which the consumers are given the possibility to receive information (events) asynchronously. Moreover, event channels determine how to propagate the changes between producers and consumers. For instance, an event channel determines the persistence of an event: it is the channel which decides for which period to hold an event, to which to send it and when. The producers generate events without having to know the identity of consumers and vice versa.

Common event models are considered an actual challenge also in the field of multimedia stream management. especially when the considered system is distributed over heterogeneous networks ("There is the need of querying and event processing architectures, simple schemes for sensor networks are not designed for multimedia sensors" [6]). Among the most interesting new approaches to multimedia event management, it is worth noting the six-dimensional "5W+1H" approach, derived from the journalism [7], [8] and particularly related to the emerging field of multimedia event management.

Other recent challenges in the field of multimedia and stream event management and modeling include multimedia over sensor networks [6], which introduce interesting dimensions related to multimedia requirements such as class type, data type, and bandwidth. In [9], event management for RFID sensors is discussed. Events are characterized as {event time, ID of RFID tags, ID of RFID readers} tuples, and XML-based communication is foreseen. A similar approach is discussed in [10]. However, this is not applicable to other types of nodes (e.g. sensors), where the low communication bandwidth foresees more efficient communication modes.
In [11] several aspects of multimedia applications are considered (structural, temporal, informational, experiential, spatial and causal). A common event model is foreseen to provide interoperability to multimedia applications of different areas, such as multimedia presentations and programming frameworks. Chang *et al.* [12] consider multimedia elements as a primary data type (*micon*), and define apposite operators like Ψ (conversion between media formats).

In addition, in practical situations, events produced by sensors such as wireless motes and RFID readers, are not significant enough for consumers. They must be combined or aggregated to produce meaningful information. By combining and aggregating events either from multiple producers, or from a single one during a given period of time, a limited set of events describing meaningful situations may be notified to consumers. Therefore, academic research and industrial systems have tackled the problem of event composition. Techniques such as complex patterns detection [13]–[16], event correlation [17], [18], event aggregation [19], event mining [20], [21] and stream processing [22]–[24], have been used for composing events. In some cases event composition is done on event logs (e.g. data mining) and in other cases it is done dynamically as events are produced (e.g. event aggregation and stream processing). Nevertheless, to the best of our knowledge, there is no approach that integrates different composition techniques. Yet, pervasive and ubiquitous computing, network and environment observation, require to observe behavior patterns that can be obtained by aggregating and mining statically and dynamically huge event logs or histories.

III. Event Definition

The definition of an event type is described by a dimension and a domain. We consider the word “dimension” as *a parameter or coordinate variable assigned to such a property*[3]. The identified dimensions are not independent, but instead they are organized in layers and are cross-referenced to characterize the events. Table 6 summarizes the main event dimensions identified in the work for characterizing events’ definition (Dimensions 1-7). Dimensions 1-3 are related to the event representation: if the event is represented by a type or not, the structure of the type, if it has a production environment or not. Dimensions 4-6 characterize the types of operators which can be associated to the event types to represent composite event types. Finally, dimension 7 (net effect) considers the net effect by means of: (i) inverse types, and (ii) an algebra of production environments.

The *net effect of a sequence of operations* represents the balance of those operations after the end of the sequence. If, for instance, a sequence of operations creates an entity, and destroys it afterwards, the net effect will be considered as zero. If two entities are created, a and b, and afterwards a is destroyed, the net effect will be the creation of b.

The definition of event types is strongly related to time, taking into account the temporal nature of the notion of event. Indeed, the definition of an event type often integrates concepts like interval and occurrence instant. Certain types can also represent other temporal dimensions such as duration, dates, periods, etc. In general, event models are based on the time models inherited from the context in which they have been designed: programming languages, data models, etc. The following characterizes the dimensions to be considered for defining events.

Dimensions 8-12 (ref. Table 2) characterize a time model. The occurrence instant of an event is represented as a point on a timeline which can be discrete or continuous (dimension 8). Granularity and temporal types, defined in dimensions 9 and 11 respectively, can be used to define event types. Granularity also characterizes the operations which may be executed on the temporal types, and which can be also used to describe event types – like, for example, *5 minutes after an user connection*: *BeginConnection + 5 minutes with delta(login:string, instantconnection:date)*.
The possibility to have conversion functions is foreseen in dimension 10, while dimension 12 describes the types of temporal operators available.

The most basic concept regarding time is the *timeline*. Abstractly speaking, a timeline is a pair $(D, <_T)$ composed by a finit set of *chronons* [4] D [25] and a binary relation $<_T$, which defines a complete and linear order over D. From the event management point of view, a timeline serves to model a discretized position in the production of a succession of events linearly ordered. This notion is particularly important, since the types conceptually represent occurrences produced within a timeline. The characteristics of the timeline allow then to choose, for instance, specific ordering algorithms, but also to specify the temporal relations between events, such as: an event e_1 was produced after e_2, or an event e_1 was produced between $9:00$ and $17:00$.

We refer to *granularity* as one partition of the set of chronons of a given timeline and its convex subsets, named *particles*, and to *minimal granularity* as the granularity obtained by dividing a timeline in singletons. Weeks, months and years correspond to *granularities*. The partial order relation *finer than* $\prec$ defines a hierarchical structure on a same timeline, which for instance allows to define that the granularity *seconds* is finer than *hours*.
The $\prec$ relation also allows to convert particles belonging to different granularities by means of conversion functions like *approximation*, which allows to rough guess a particle of a granularity G_1 by means of a particle G_2 which contains it (*zoom in*), and *expansion*, which allows to associate a set of granularities G_1 to each particle of granularity G_2 (*zoom out*). Interested readers can refer to [26] for further details.

Starting from the concept of granularity, a set of types which get involved directly in the definition of event types has been identified:

– An *instant* is a point within a timeline which can be represented by an integer, when a discrete time representation is adopted;
– A *duration* is a number of particles used as a distance measure between two instants, to allow the expression of movements in time with respect to a given instant. In general, it is characterized by a positive integer (its measure) and by a granularity, like *4 seconds*;
– An *interval* is represented by the bias between two instants, or by an instant (the lower bound of the interval) and its duration. Given that the lower and upper bounds of an interval are both of the same granularity, the interval can be represented by means of a granularity and two positive integers (the positions of the bounds).

The temporal types of the programming languages and the query languages are generally provided with operators. In our opinion, the four basic following operators on time models which should be included in any software implementation are:

– *selectors* of the maximum/minimum instant and of the duration of a set of instants with the same granularity,
– *order relations* on the instants ($<, >, =$),
– *arithmetic operators* between instants and durations like addition and subtraction of a duration to an instant, or between two durations, and
– *conversion operators* between two temporal values observed with different granularity levels. Interested readers can refer to [27] for more details.

[3] Merriam-Webster English dictionary. The geometric meaning of dimension, for space definition (e.g. *a three-dimensional space*), will not be considered here.

[4] A chronon is a proposed quantum of time in the Caldirola’s discret time theory.

TABLE I
DIMENSIONS OF EVENT TYPES.

	Dimension	Domain
1	Event	{*with type, without type*}
2	Event type	{*string, expression, object*}
3	Production environment	{*yes, no*}
4	Operator types	{*selection, algebraic, temporal*}
5	Validity interval	{*interval, period, none*}
6	Filtering	{*regular expressions, predicates*}
7	Net effect	{*inverse events, algebra of production environments, none*}

TABLE II
DIMENSIONS CHARACTERIZING THE TIME MODEL.

	Dimension	Domain
8	Time	{*discrete, continuous*}
9	Granularity	{*day, month, year, hour, minute, second*}
10	Conversion function	{*yes, no*}
11	Temporal types	{*instant, interval, duration*}
12	Temporal operators	{*comparison, selection, join*}

IV. DETECTION

DETECTION is the process through which an event is recognized and associated to a point in time named *instant of occurrence*. The events may be observed by a process external and independent from the producer, or signaled by the producer itself.

The detection is characterized by the conditions in which the events are observed, and can be modeled by dimensions from 13 to 16 described in Table 3 and explained in the remainder of this section. The production unit is the interval during which producer can observe events (dimension 13). The observation point (dimension 14) can be located at the operation start or end points. The interaction protocols used to retrieve them, finally, are described by type (dimension 15) and detection mode (dimension 16).

The PRODUCTION UNIT identifies the interval during which the events can be detected. More precisely, it specifies the interval within the execution of a producer in which events are produced. The interval can be defined by the duration of the execution of a program, a transaction, an application, an user connection.

For example, the transaction is the production unit in most of centralized active DBMS [28]. Few active DBMS allow event detection mechanisms without transactions. The so-called external events, that is, the event which do not represent operations on the base and aren't inevitably produced inside transactions, are also detected in the context of a transaction. Distributed active systems [29]–[31] allow detection of events coming from different DBMS and applications. In this case, the events are observed by the detectors within the transactions. The detectors are synchronized by a global detection mechanism, which builds a global view of events produced within different transactions – and without transactions.

It is possible to associate an implicit production unit to a set of producers. In this case, event detection is active as long as there is any producer subscribed, even if there are no consumers. The Microsoft event service [32] defines the production unit as implicit and bound to the duration of the execution of the producer objects, while the Java event service [33] bounds the production unit to the duration of the execution of the producer objects, "reducing" the management to a *multicast* notify mechanism: the producer objects notify events to the subscribed consumer objects. The production unit in streams in sensor networks and also managed by Data Stream Management Systems (DSMS) is determined by explicit time intervals.

Event detection mechanisms face a granularity issue when the processes to be observed have a duration and the events which represent them are instantaneous. To this concern, certain systems distinguish between physical and logical events. A PHYSICAL EVENT is the instance which has been detected, while a LOGICAL EVENT is its conceptual representation (expressed by an event type).

We refer to the physical events which have been detected as an *occurrence of an event type*. A logical event which represents an observed operation can be split in two physical events detected repectively *before* and *after* the operation, according to the observation point of the detection process. For instance, the update operation on the *balance* variable of an entity with type ACCOUNT may be represented by an event type associated to an observation point named "point_obs": < *point_obs* > *UpdateAccount with delta(accountnumber:integer, newbalance:real, oldbalance:real).*

TABLE III
DIMENSIONS OF EVENT DETECTION.

	Dimension	**Domain**
13	Production unit	{*duration of execution of the producer, transaction, connection, application*}
14	Observation point	{*before, after*}
15	Detection protocol	{*pull, push* }
16	Detection mode	{*synchronous, asynchronous*}

The operation can be handled by associating it to a time interval $[t_0, t_1]$ where t_0 correspond to the operation start and t_1 to its end. An event can be observed at t_0, corresponding to the transition:

update operation inactive $\rightarrow$ *update operation running*[5], and at t_1, corresponding to the transition:

update operation running $\rightarrow$ *update operation finished*[6].

Both cases refer to the same event, but detected in two different instants. The modifier *before* and *after* allow to state the events observation point with respect to the operation executed within a production unit.

The DETECTION PROTOCOL identifies the way of interaction with the producer in order to retrieve the events. In general, events are detected with a protocol of type *push* if the producer explicitly reports the events. In case of type *pull*, the detection mechanism queries or observes the producer to retrieve the events. The choice of one of the two protocols depends on the characteristics of the producers.

The DETECTION MODE relates the event detection mechanism to the execution of the producer. In the *synchronous* detection mode, the producer stops its execution to signal the event. On the contrary, the *asynchronous* detection mode assumes that the producers report the events to the detection mechanism without having to interrupt their execution.

In general, asynchronous detection is achieved by means of a shared memory space. The *asynchronous pull* detection mode assumes that there is a monitoring mechanism implemented in the producer which observes its execution and stores the events in a shared memory space accessible by the detection mechanism. In case of *synchronous pull* detection, the execution of the producer can be interrupted instead.

V. PRODUCTION

The PRODUCTION process corresponds to the time stamping process of a detected event – taking into account the instant at which the event occurs – and to its insertion within an event history. Production is based on read and write access to an history of produced events, as well as on the computation of the *production instant* of the events. The dimensions of the production specify policies for ordering and composing detected events (see Dimensions 17 - 20 in Table 4). Such policies determine how to time stamp events, and which events of the history should be used to produce composite events.

The TIME STAMPING process is the process through which events are labeled with information regarding their instant of occurrence. This process is based on the notion of *clock*, that is, a function $C(e)$ which associates an event e to its instant of occurrence I_{occ}. A *time stamp* specifies the position of an event on a timeline. The structure of time stamps varies depending on the observation of events with respect to a local or global reference.

In a centralized system, time can be described as a completely ordered sequence of points [7], where instants correspond to readings of the system local clock. Let e_1 and e_2 be two events, detected respectively at the instants $I_{occ}(e_1)$ and $I_{occ}(e_2)$. It is then possible to establish a total or partial order between the two events by ordering their instants of occurrence. Consequently, e_1 is produced before, after or at the same time than e_2.

In a distributed system, events are generally produced at points in time identified by different clocks. According to the observation point, the relative order between two events can vary depending on the observer's position. As a consequence, when events are produced by multiple producers and observed by multiple consumers, it is necessary to choose a reference clock in order to have a global perception of the events. The time point which an event is associated is then "associated" with a point of the reference clock, taking into account the drift of producer and reference clock with respect to an universal global reference point.

The GLOBAL TIME g_{t_k} of the instant I_{local_k}, read in a local clock k, is described by a point of the Gregorian calendar (*Universal Time Coordinated*) truncated to a global granularity g_g:

$g_{t_k}(I_{local_k}) = TRUNC_{g_g}(clock_k(I_{local_k}))$,

where $TRUNC()$ is a rounding function like $round()$, $ceil()$, $floor()$ depending on the application context.

The "global" time stamp of a event allows to determine its production instant in a global timeline, knowing its position on another temporal reference called *local* with respect to a

[5] The transition is represented by the following event: $<$ *Before* $>$ *UpdateAccount with delta(accountnumber:integer, newbalance:real, oldbalance:real).*

[6] The transition is represented by the following event: $<$ *After* $>$ *UpdateAccount with delta(accountnumber:integer, newbalance:real, oldbalance:real).*

[7] Declaring that the time points are completely ordered implies that, for any pair of points t_1 and t_2, the temporal relation between them is either $t_1 < t_2$, $t_1 = t_2$ or $t_1 > t_2$.

TABLE IV
DIMENSIONS OF EVENT PRODUCTION.

	Dimension	**Domain**
17	Time stamping	$\{ < I_{occ}global >, < site, I_{occ}local >, < I_{occ}local, site, I_{occ}global > \}$
18	Granularity	{*instance, set of the same type, set of different type*}
19	Consumer range	{*local,global*}
20	Production mode	{*continuous, recent, chronological, cumulative*}

given *site*. The time stamp of an event $T(e)$ is thus a tuple of the form $<$ $I_{occ}(e)$, *site*, $I_{occ_{global}}(e)$ $>$, where $I_{occ}(e)$ is the instant of occurrence with respect to the local clock of the producer *site*, and $I_{occ_{global}}(e)$ is the *global instant of occurrence* with respect to a reference clock. For example, the time stamp of an event e produced in the site k may be of the following form:
$T(e)$=*<23991548127, k, ('19/10/95',* $2:32:27.32$*)>*

The time stamping of a composite event is determined by the most recent component event. In a distributed context, the notion of "most recent" is not unique, that is, multiple component events exist which are virtually produced at the same time and which may contribute to triggering a composite event. In this case, all events contribute to determine the time stamp of the composite event. The time stamping process is summarized in Equation 1.

Given a set of time stamps ES and a time stamp $st \in ES$, the maximum can be defined as:
$st = Max(ES) \iff (\nexists\ st_1 \in ES, st < st_1)$.
A set of maximum time stamps is therefore defined as:
$Max(ES) = \{st\ \in ES : st\ is\ a\ maximum\ of\ ES\}$.

The time stamp $T(e)$ of a composite event is the set $Max(ES)$, where ES is the set of time stamps of the component events. The production instant, as a function of the semantics of composition operators, can then be computed by using the set $Max(ES)$. For example, given the following join and sequence semantics:

$(E_1 \wedge E_2)\,(st) =\ \exists st_1, st2\,[E_1(st1) \wedge E_2(st2)] \wedge$
$[st = Max(st_1, st_2)]$ *(intersection)*
$(E_1; E_2)\,(st) =\ \exists st_1\,[E_1(st1) \wedge E_2(st2)] \wedge$
$(st_1 < st)$ *(strict sequence)*

To establish an order between events, it is necessary to compare their time stamps. The ordering procedure concerns the management of the event history, which will be discussed in Section VII.

It is possible to distinguish between events of different granularities according to the number of occurrences they are composed of. In general, the following two PRODUCTION GRANULARITIES can be identified:

– *instance*: an event is produced every time that an occurrence of event type is detected; and
– *set*: an event is produced at the moment of the detection and composition of a set of events of the same type, or of different type.

For example, let us consider an event of type *creation of a new bank account* detected at the moment of an insertion in a relation *Accounts*. With an *instance-oriented* granularity, the event is produced every time that the operation "a tuple is inserted into the *Account* relation" is executed. On the contrary, what happens if N bank accounts are created? Would it be needed to produce an event *creation of a new bank account* for each insertion or just one event which represents the *creation of N bank accounts*? The choice is determined based on what the consumer wants to observe and according to the application context.

A set of granularities *of the same type* allows to produce events which group N occurrences of the same type. For example, all occurrences of type *creation of a new bank account* produced in the context of a single transaction or all the purchases made by a client within the last month.
A set of granularities *of different types* allows to produce composite events by combining the events with operators such as join, disjoint, sequence, etc. For example, *creation of a new bank account followed by a deposit of more than 1000 EUR*:

CreateAccount with delta(accountnumber:string,
owner:string, balance:real); [8]
UpdateAccount with delta(accountnumber:string,
oldbalance:real, newbalance:real)
where
CreateAccount.accountnumber=
UpdateAccount.accountnumber
and $oldbalance - newbalance > 1000$

In the two cases, it is necessary to specify the production conditions of the component events. For example, the component events should be produced by the same producer or within the same production unit. Such aspects are related to the semantics of the composition operators, but also to the *construction of the production environment*, as explained in the following.

The production process begins with the detection of basic events[9]. When a basic event is detected, it is necessary to verify if its occurrence initializes or triggers the production of a composite event. The production of such event depends on the occurrence of its component events and on the order with which the occurrences are produced.

[8] The operator ";" represents the *sequence* operator.

[9] We talk about *detection* of basic events and *production* of composite events.

$$Max\,(T(e_1), T(e_2)) = \begin{cases} T(e_1) & T(e_2) < T(e_1) \\ T(e_2) & T(e_2) < T(e_1) \\ T(e_1) \cup T(e_2) & T(e_2)\ and\ T(e_1)\ cannot\ be\ compared \end{cases} \quad (1)$$

The production of composite events is based on a *production mechanism*. This mechanism "knows" the structure of the composite event and the order with which its component events should have taken place for the composite event to be triggered. Everytime a simple event is detected, it is notified to the production mechanism. If the production can move forward, a new production state is derived. A certain *final state* indicates the triggering of a *composite event*.

The production of composite events has been deeply studied within the active databases domain [34], [35]. To date, most of the production mechanisms of composite events are based on the evaluation of *finite state automata*, *Petri nets* and *graphs*.

Let the reader remind that all events convey information which differentiates them and informs on the conditions under which they are produced: production instant, producer identifier, real parameters inherent to its type (see *production environment* within Section II. The *construction of the production environment* of an event is determined by the production granularity chosen to produce it. For example, let us consider an history $h = \{e_{12}, e_{13}, e_{24}, e_{15}, e_{26}, e_{27}, \ldots\}$, containing also occurrences of type *operation made on a bank account*:
E_1 *= Deposit with delta(accountnumber:integer, amount:real)*

The type E_1 conveys information concerning the account number and the amount of the operation which has been made. With a production granularity of *instances*, three events e_{12}, e_{13}, e_{15} will be produced and the production context of each contains the instant of production as well as the real parameter associated to its type, that are, bank account number and the amount of the operation which has been made.

When the construction of the production environment is by *set*, it is necessary to specify which events – produced previously and available in the history – participate in its construction. For instance, an event representing *n deposits made on the same bank account between* $[9:00, 17:00]$:
E_1 *= Deposit with n* × *delta(accountnumber:integer, amount:real)*
within [9:00,17:00] where same(accountnumber)

The CONSUMPTION SCOPE defines the selection policies of the events belonging to the history used for building the production environment of an event e_i. The scope of consumption can be:

– *local* with respect to (i) a producer, for example, when events of the same producer are selected; (ii) a specific set of producers, for example, when events produced by processes running on the same server are selected; (iii) to a time interval, for example, when only events produced within the same production unit are used (e.g. a transaction); or

– *global*, if all instances present in the history are used to build the production environment of an event (e.g. when the event contains information regarding bank transactions effectuated during the day).

The PRODUCTION MODE determines the consumption protocol of events to build composite events. It indicates the combinations of primitive events that participate in the composition of an event and clarifies the semantics of composite event types. The notion of *production mode* of events has been introduced by Snoop [36] with the name of *parameter context*.

Four production modes have been proposed in the database domain. *(i) Continuous:* all occurrences which time stamp the begin of an interval of a composite event type are considered as initiator events of composite events. *(ii) Recent:* only the most recent events are used to trigger occurrences of E. *(iii) Chronological:* the occurrences of events are considered in their chronological order of appearance. The component events are used with a "FIFO"–type strategy. *(iv) Cumulative:* when an occurrence of E is recognized, the context to which it is associated includes — cumulates — the parameters of all occurrences of events which participate to its construction. For instance, NAOS [35] implements the *continuous* mode, Sentinel [37] implements all the four modes, Chimera [38] supports the *recent* mode, SAMOS [39] implements the *chronological* mode.

VI. NOTIFICATION

The NOTIFICATION process deals with the notification of events to the consumers. The notification of an event can be made at specific instants in relation to their instant of production and considering temporal constraints. The events can also be filtered before being notified. The dimensions of the notification, corresponding to dimensions 21-27 in Table 5, characterize the aspects to be taken into account for the notification of events to the consumers. Such aspects are determined by the specific consumer needs in terms of information (validity interval, instant of notification, history management, communication protocol), but also by the autonomy and the isolation needs of the producers (granularity and range of selection, visibility of events in the history).

The VALIDITY INTERVAL (dimension 21), specifies an observation window on events belonging to the history. This interval allows to specify in which period of time a consumer is interested on events of a specific type. For example, the following expression:

TABLE V
DIMENSIONS OF EVENT NOTIFICATION.

	Dimension	**Domain**
21	Validity interval	{*implicit, explicit*}
22	Notification instant	{*immediate with [Δ]C [Comp], immediate without C, [Δ] differed wrt. an event*}
23	Selection granularity	{*instance, set*}
24	Selection range	{*local, interval, global*}
25	Visibility	{*local, global*}
26	Notification protocol	{*pull, push*}
27	Notification mode	{*synchronous, asynchronous*}

after UpdateAccount with delta(accountnumber:integer, oldbalance:real, newbalance:real), [9:00,17:00]

allows to focus on an *update on a bank account executed between* $9:00$ *and* $17:00$. Only events which take place in the active production unit(s) (a transaction, a program, etc.) in the interval $[9:00, 17:00]$ are considered. If the validity interval and the production unit correspond, the events are then relevant during the execution of the transaction (production unit by default) in which they are produced and are not visible outside such interval. Treating separately the event production unit (on the detection side) and the validity interval (on the notification side) contributes to the flexibility of the event notification mechanism, while allowing the consumers to decide the periods during which they want to observe events.

The NOTIFICATION INSTANT (dimension 22) specifies the instant when the notification mechanism must retrieve the events from the history to notify them to the consumers. The events are delivered at different instants according to the degree of reliability of the conveyed information, the notification rate required by the consumers, and so on. It is possible to notify them:

– *immediately after the production*;
– at a *precise instant* with respect to the production of another event, for instance at the end of the stock market session, every two minutes, on 10/09/00 at 14:30; or
– with respect to a *latency time* defined as follows: it exists a constant Δ such as, if the instant of production of an event e_i is t, then the notification mechanism sends the event after $t + \Delta$.

It is also possible to associate a *confirmation* to the event notification, to allow the validation – or invalidation – of the notified occurrences. For example, the event e_1 “an operation of purchase of actions has been executed” can be confirmed by the event e_2 “authorization confirmed by the bank”. It is possible to specify a temporal constraint (*latency time*) associated to the notification of the confirmation event. For example, e_2 should be notified with a *15 minutes delay*. If the constraints are never verified, it can also be defined a *compensation event* notified instead of the confirmation event. An instant of notification is associated to the compensation event too.

The EVENT SELECTION process specifies the policies to be used to choose the events belonging to the event history when it is necessary to notify them to the consumer. These policies are essentially determined by the consumers’ needs and the characteristics of the producers (for instance, autonomy), but also by the characteristics of the applications common to producers and consumers.

The SELECTION GRANULARITY (dimension 23) describes the selection criteria of history events:
(i) *instance–oriented*: only the last occurrence of an event is notified;
(ii) *set–oriented*: all instances available in the history are notified.

The SELECTION SCOPE (dimensions 24, 25) describes the visibility of history events with respect to the consumers. Two scopes of selection are possible based on applications’ needs:

– *local*: each consumer is granted access to a history events subset. There can be two different criteria: selection of events produced within a specified time interval (e.g., selection of the events detected in the same production unit (*intra–production unit*)), or considering a content–based filtering;
– *global*: events produced within other executions (*inter–production unit*) are selected, that is, beyond the limitations of applications and transactions. For example, the events produced within a given transaction are visible after the validation or before it.

The *inter–production unit* approach poses issues which still have to be treated, like: (i) what to do of history events which production unit has terminated?, and (ii) until which moment are they visible to the consumers?

The dimensions of notification (26 and 27) characterize also the communication protocol and mode used to notify events to the consumers. The NOTIFICATION PROTOCOL describes the way in which the notification mechanism interacts with the consumer during the event notification process, while the NOTIFICATION MODE define the way the operations are executed. Asynchronous notification assumes consumers and notification mechanism exchange events through a shared memory space. The *pull asynchronous* notification mode assumes that the notification mechanism stores the events in a memory space which may be queried by the consumer to retrieve the events. In case of a *push synchronous* notification, the notification mechanism must be able to stop the consumer(s) execution.

VII. History

An EVENT HISTORY is an ordered set of instances of events. The event history plays a fundamental role in the event production and notification. The management policies of the event history, summarized by means of dimensions 28-31 in Table 6 determine which events have to be inserted in the history and for how long, and when to insert and delete them. The management of the event history specifies then the insertion, selection and history update policies.

EVENTS INSERTION (dimension 28) in the event history is determined by an ordering function which allows to determine the position of an event (just occurred) in the history. According to specific algorithms like the precedence model $2g_g$ proposed by [40], we assume that it is always possible to determine an ordering (total or partial) between history events, based on comparison of timestamps. Let the reader refer to Appendix A for more details concerning timestamps comparison.

The HISTORY SELECTION is determined by the events *visibility* adopted for the production of events and by the *selection scope events* adopted for the notification. The *visibility* of events adopted for the production describes the policies to be used to consume the events part of the history during the construction of the production environment of an event e_i. With a policy:

– *intra–occurrence*, the event e_i is produced and ignored in the construction of the production context of other occurrences;

– *inter–occurrence*, after having been produced, the event e_i is kept visible for the construction of production environments of other occurrences.

The UPDATE of the history (dimension 29) is performed in the following cases:

– *Invalidation*: An event stored in the history can be invalidated by the production of a second event. The *net effect* allows to determine this situation. To be able to compute the net effect of an event of type E, its inverse type E^{-i}, as well as the net effect computation rules (defined by the model of event types) should initially be known. The invalidation of events affects the event notification process. In particular, the cancellation of events already notified but that wait for a confirmation. If the event is cancelled before the production of the confirmation event, strategies to notify such situation should be planned.

– *Expiration*: The consumers determine a validity interval for the various event types. When such deadline expires, event can be deleted from the history if they have been received by all consumers. A "sophysticated" system where the consumers grant different validity intervals to the same event type can be imagined. In this case, the definition of views within the history can be interesting.

– *Cancellation*: The production of an event can be cancelled by a particular situation. For example, when events are produced within transactions, their production may be cancelled if the transaction fails. In this case, it is necessary to delete from the history all cancelled events.

– *Explicit invalidation*: Instances of events can be deleted as a consequence of explicit requests formulated by clients or users [10].

A *transient* event has a duration equal to zero. On the contrary, a *persistent* event has a longer duration. So, for example, an event may persist as long as the control stream by which it has been generated exists, the object from which has been produced persists or another event still has to be produced. The PERSISTENCE (dimension 30) describes the time interval during which the events are kept in the history. Four policies can be adopted. An event e_i is kept:

1) *until its notification*: e_i is kept in the history until it has been notified to all its consumers, then it is deleted;
2) *during the validity interval*: events are kept in the history until the end of their validity interval;
3) *for the production unit*: events do not survive the execution of their producer. For example, event are kept in the history until the end of the transaction, as long as a program is running or an user is connected. As mentioned earlier, in many active DBMS event models the production unit (transaction) corresponds to the validity interval. In this case, events are kept until the end of the transaction;
4) *until the production of the next event of the same type*: a new occurrence of an event causes the deletion of the previous occurrence. In this case, two situations are possible: the new event can add up to the production environment of its predecessor (at the condition that it refers to the same data), or can ignore it.

Taking into account the net effect has a few implications on the persistence policies, since in the case of a persistence policy *until the notification*, an event e_i can be cancelled by another event of "inverse type" between the time of its production and the moment when the notification has taken place. In the other cases, an event of inverse type can always cancel e_i.

The events describe changes, transitional by nature and which can be assimilated to a transition between two states, generated within a system and which determine its evolution. PERMANENCE aims at making the evolutions permanent by storing in the disk the history of corresponding events, in validation points explicitly set by the application programmers.

A number of applications exist, such as workflow or data mining applications, for which the permanence of events could be useful — as when the so–called "durative" events have to be detected. For example, the detection of an event like *5 days after the creation of a bank account* requires the storage of the event *creation of a bank account* for at least five

[10] Which is undoubtedly an effective method to tamper with event history.

TABLE VI
DIMENSIONS OF EVENT HISTORY.

	Dimension	Domain
28	Insertion	{*ordering function*}
29	Updating	{*invalidation, expiration, cancellation, explicit cancellation, none*}
30	Persistence	{*validity interval, production unit, until a more recent e_i is produced* }
31	Permanence	{*validation point, explicit, none*}

consecutive days. The migration of processes can also need the permanence of events. A sequence of events reflects indeed the execution state of one or more processes. It is potentially desirable to store this information, migrate the process, then recover the events by means of which it can be relaunched with its last state.

VIII. Event Service Framework

An event service is a mediator in event-based systems enabling loosely coupled communication among producers and consumers. Producers publish events to the service, and consumers express their interest in receiving certain types of events by issuing event subscriptions. A subscription is seen as a continuous query that allows consumers to obtain event notifications [23], [24]. The service is then responsible for matching received events with subscriptions and conveying event notifications to all interested consumers [41]. Within our framework, the challenge is to design and implement an event service that implements event composition by querying distributed histories ensuring scalability and low latency.

We propose a generic event service that provides support for managing events in distributed systems. The service architecture consists of cooperative and distributed event detectors, which receive, process and notify events. The event service detects the occurrence of primitive and composite events and consequently notifies all the applications or components that have declared their interest in reacting to such events. Two main detector types are provided by the service: *Primitive Event Detectors*, which gather events from producers; and *Composite Event Detectors*, which process and produce events from multiple, simpler detectors.

Event Detectors are connected forming an event composition network. Events obtained from producers are propagated through the composition network and incrementally processed at each Composite Event Detector involved. Event processing functions can be separately configured for each Composite Event Detector. The most important configurable functions include event composition, correlation, aggregation, filtering according to the dimensions described in the previous sections.

Event Detectors are managed by the ***Service Manager*** component. They are dynamically created and configured in base on advertisements and subscriptions from producers and consumers respectively. Then, the Service Manager manages event advertisements and subscriptions. It maintains a list of all the detectors of the system and the event types (primitive or composite) that each one of them manages (receives and notifies).

The Service Manager implements the `DataCollector` interface, receiving advertisements and subscriptions from producers and consumers respectively, in order to create and configure Event Detectors. The Service Manager uses `EDManagement` interfaces in order to (re-)configure Event Detectors.

We propose a distributed event composition approach, done with respect to subscriptions managed as continuous queries, where results can also be used for further event compositions. According to the type of event composition strategy (i.e., aggregation, mining, pattern look up or discovery), event composition results can be notified as data flows or as discrete results. Event composition is done with respect to events stemming from distributed producers and ordered with respect to a timestamp computed with respect to a global time line. Provided that our event approach aims at combining different strategies that can enable dynamic and postmortem event composition, we assume that different and distributed event histories can be used for detecting composite event patterns. For example combining a history of events representing network connections with another history that collects events on the access to a given address in the Internet for determining the behavior of users once they are connected to the network.

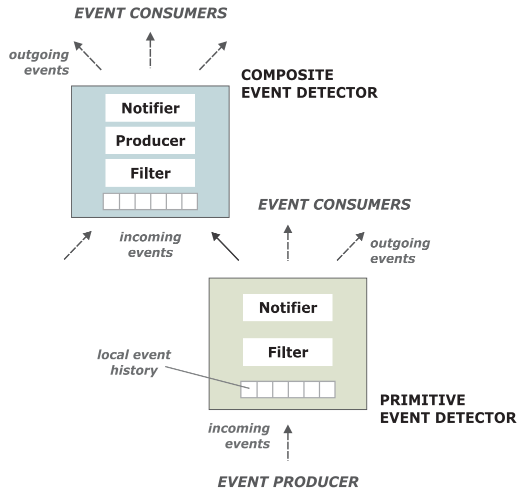


Fig. 1. *Primitive and Composite Event Detectors*

Therefore, both histories must be analyzed for relating the connection of a user to the sites he/she visits at different moments of the day.

The implementation of the event service extends the implementation of the Composite Probes framework [42]. The prototype implementation is based on Julia [43], a Java implementation of the Fractal component model [44], and on additional Fractal utilities, including Fractal ADL [45]. The service prototype implements Primitive and Composite Detectors. The service of communication is implemented via JORAM [46], a JMS implementation. Additionally, Fractal RMI and Fractal JMX should be used for the distribution and managing of the Fractal components. The event service implementation uses Fractal bindings for interconnections among Event Detectors. Using bindings allows both local communication inter-components via direct method calls, and distributed communication based on Fractal RMI [47]. In addition, maintaining the binding principle provides a true architecture among Event Detectors.

IX. Conclusions

A general framework for processing events and thereby supporting the communication among producers and consumers has been presented in the paper. In our approach we consider that event based communication must be adapted according to the kind of producers and consumers. For example, continuous detection/notification of events in sensor networks is not the same as detecting and notifying events concerning RSS flows of a web site or monitoring middleware infrastructures for supporting business services. Our framework and its associated event service can be personalized thanks to a general meta-model that provides dimensions for programming ad-hoc event management policies. The paper also describes our implementation experience of an event service that implements the meta-model and that is based on composite probes [42] and that can be configured for detecting, composing and notifying events.

We are currently developing a general event composition framework that can act in a completely distributed way and support dynamic event composition and event mining on post-mortem event histories. We are particularly interested in supporting monitoring for cloud-computing computing and business services, middleware and computer services monitoring, and for transmission of multimedia content over ad-hoc networks.

Acknowledgments

The authors would like to thank Javier A. Espinosa-Oviedo for his precious support during the writing of the paper.

References

[1] *ADEES: An Adaptable and Extensible Event Based Infrastructure*. London, UK: Springer-Verlag, 2002.

[2] Y. Yao and J. Gehrke, "The cougar approach to in-network query processing in sensor networks," vol. 31, no. 3. New York, NY, USA: ACM, 2002, pp. 9–18.

[3] D. J. Abadi, D. Carney, U. Çetintemel, M. Cherniack, C. Convey, S. Lee, M. Stonebraker, N. Tatbul, and S. Zdonik, "Aurora: a new model and architecture for data stream management," *The VLDB Journal*, vol. 12, no. 2, pp. 120–139, 2003.

[4] D. D. et al. (2010, February) Web services event descriptions (ws-eventdescriptions), w3c working draft. [Online]. Available: http://www.w3.org/TR/2010/WD-ws-event-descriptions-20100209/

[5] I. A. Ismail, I. F. Akyildiz, and I. H. Kasimoglu, "Wireless sensor and actor networks: Research challenges," 2004.

[6] T. M. I. F. Akyildiz and K. R. Chowdury, "Wireless multimedia sensor networks: A survey," *IEEE Wireless Communications*, vol. 14, no. 6, pp. 32–39, December 2007.

[7] L. Xie, H. Sundaram, and M. Campbell, "Event mining in multimedia streams," *Proceedings of the IEEE*, vol. 96, no. 4, pp. 623–647, 2008. [Online]. Available: http://dx.doi.org/10.1109/JPROC.2008.916362

[8] X.-j. Wang, S. Mamadgi, A. Thekdi, A. Kelliher, and H. Sundaram, "Eventory – an event based media repository," in *ICSC '07: Proceedings of the International Conference on Semantic Computing*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 95–104.

[9] J. Xu, W. Cheng, W. Liu, and W. Xu, "Xml based rfid event management framework," nov. 2006, pp. 1 –4.

[10] S. Bose and L. Fegaras, "Data stream management for historical xml data," in *SIGMOD '04: Proceedings of the 2004 ACM SIGMOD international conference on Management of data*. New York, NY, USA: ACM, 2004, pp. 239–250.

[11] U. Westermann and R. Jain, "Toward a common event model for multimedia applications," *IEEE MultiMedia*, vol. 14, no. 1, pp. 19–29, 2007.

[12] S.-K. Chang, L. Zhao, S. Guirguis, and R. Kulkarni, "A computation-oriented multimedia data streams model for content-based information retrieval," *Multimedia Tools Appl.*, vol. 46, no. 2-3, pp. 399–423, 2010.

[13] O. S. N.H. Gehani, H.V. Jagadish, "Composite event specification in active databases: model & implementation," in *18th*, Septembre 1992.

[14] S. Gatziu and K. R. Dittrich, "Detecting composite events in active database systems using Petri nets," in *4th International Workshop on Research Issues in Data Engineering: Active Database Systems*, 1994.

[15] C. Collet and T. Coupaye, "Composite Events in NAOS," in *7th(DEXA'96)*, Zurich - Switzerland, 9-13 Septembre 1996.

[16] P. P, S. B, and B. J, "Composite event detection as a generic middleware extension," *Network*, vol. 18, no. 1, pp. 44–55, 2004.

[17] D. C, "Alarm driven supervision for telecommunication networks. on line chronicle recognition," *Annales des Telecommunications*, pp. 501–508, 1996.

[18] E. Yoneki and J. Bacon, "Unified semantics for event correlation over time and space in hybrid network environments," in *OTM Conferences*, 2005.

[19] D. Luckham, *The Power of Events: An Introduction to Complex Event Processing in Distributed Systems*. Addison Wesley Professional, 2002.

[20] R. Agrawal and R. Srikant, "Mining sequential patterns," in *PROC 11th International Conference on Data Engineering*, IEEE, Ed., Taiwan, 1995.

[21] A. Giordana, P. Terenziani, and M. Botta, "Recognizing and Discovering Complex Events in Sequences," in *ISMIS 02: Proceedings of the 13th International Symposium on Foundations of Intelligent Systems*, London, 2002.

[22] E. Wu, Y. Diao, and S. Rizvi, "High-Performance Complex Event Processing over Streams," in *SIGMOD*, 2006.

[23] A. J. Demers, J. Gehrke, B. Panda, M. Riedewald, V. Sharma, and W. M. White, "Cayuga: A General Purpose Event Monitoring System," in *CIDR*, 2007.

[24] M. Balazinska, Y. Kwon, N. Kuchta, and D. Lee, "Moirae: History-Enhanced Monitoring," in *CIDR*, 2007.

[25] P. Caldirola, "The introduction of the chronon in the electron theory and a charged-lepton mass formula," *Lettere Al Nuovo Cimento (1971-1985)*, vol. 27, no. 8, pp. 225–228, February 1980.

[26] G. Iqbal-A, Y. Leontiev, M.-T. Ozsu, and D. Szafron, "Modeling temporal primitives: Back to basics," 1997.

[27] M. Dummett, *Elements of intuitionism*, 2nd ed., ser. Oxford logic guides 39. Clarendon Press, 2000.

[28] N. W. Paton, *Active Rules for Databases*. Springer Verlag, 1998.

[29] S. Berchtold, D. A. Keim, and H.-P. Kriegel, "Using extended feature objects for partial similarity retrieval," *The VLDB Journal*, vol. 6, no. 4, pp. 333–348, 1997.

[30] Y. Li, M. Potkonjak, and W. Wolf, "Real-time operating systems for embedded computing," in *International Conference on Computer Design*, 1997, pp. 388–392.

[31] J. pieter Katoen and L. Lambert, "Pomsets for message sequence charts," in *1st Workshop of the SDL Forum Society on SDL and MSC, SAM98*, 1998, pp. 291–300.

[32] D. A. Menascé, "Mom vs. rpc: Communication models for distributed applications," *IEEE Internet Computing*, vol. 9, no. 2, pp. 90–93, 2005.

[33] D. Flanagan, *Java In A Nutshell, 5th Edition*. O'Reilly Media, Inc., 2005.

[34] S. Chakravarthy, V. Krishnaprasad, Z. Tamizuddin, and R. H. Badani, "ECA Rule Integration into an OODBMS: Architecture and Implementation," University of Florida, Department of Computer and Information Sciences, Technical Report UF-CIS-TR-94-023, Mai 1994.

[35] C. Collet, "NAOS," in *In Active Rules for Databases*, N. W. Paton, Ed. Springer Verlag, 1998.

[36] S. Chakravarthy and D. Mishra, "Snoop: An Expressive Event Specification Language For Active Databases," University of Florida, Gainesville, Tech. Rep. UF-CIS-TR-93-007, Mars 1993.

[37] S. Chakravarthy, "Sentinel: an object-oriented dbms with event-based rules," in *SIGMOD '97: Proceedings of the 1997 ACM SIGMOD international conference on Management of data*. New York, NY, USA: ACM, 1997, pp. 572–575.

[38] S. Ceri and R. Manthey, "Consolidated Specification of Chimera," IDEA Esprit Project, Politecnico di Milano, Milano - Italy, Tech. Rep. IDEA.DE.2P.006.01, 1993.

[39] S. Gatziu, H. Fritschi, and A. Vaduca, "SAMOS an Active Object-Oriented Database System: Manual," Zurich University, Tech. Rep. Nr 96.02, Février 1996.

[40] S. Schwiderski, "Monitoring the behaviour of distributed systems," Tech. Rep., 1996.

[41] M. G, F. L, and P. P, *Distributed Event-Based Systems*. Springer-Verlag, 2006.

[42] "Composite probes," http://forge.objectweb.org/projects/lewys.

[43] "Julia," http://fractal.objectweb.org/julia/.

[44] "Fractal," http://fractal.objectweb.org/.

[45] "Fractal," http://fractal.objectweb.org/fractaladl/.

[46] "Joram," http://joram.objectweb.org/.

[47] "Fractal," http://fractal.objectweb.org/fractalrmi/.

[48] S. Chakravarthy and S. Yang, "Formal semantics of composite events in distributed environments," 1999.

APPENDIX

Let the reader remind that a timestamp $T(e)$ of an event e_i is a tuple $< I_{occ}(e_i)$, *site*, $I_{occ_{global}}(e_i) >$ [11]. Yang and Chakravarty define in [48] the temporal ordering relationship between the timestamps of primitive events as follows:

1) e_i before e_j:
   $T(e_1) < T(e_2) \iff$
   $(T(e_1).site = T(e_2).site) \wedge (I_{occ}(e_1) < I_{occ}(e_2)) \vee$
   $(T(e_1).site \neq T(e_2).site) \wedge (Iocc_{global}(e_1) < Iocc_{global}(e_2))$
   Event e_1 is produced before event e_2 if:
   (i) the two events are produced at the same location and the instant of production of e_1 is smaller than that of e_2; or
   (ii) if they are produced at different locations and the global instant of production of e_1 is smaller than that of e_2.
2) Simultaneous:
   $T(e_1) = T(e_2) \iff (T(e_1).site = T(e_2).site) \wedge (I_{occ}(e_1) = I_{occ}(e_2))$
   Events e_1 and e_2 are simultaneous if the two are produced at the same location and they have the same instant of production.
3) Concurrent:
   $T(e_1) \sim T(e_2) \iff (T(e_1) < T(e_2)) \vee (T(e_2) < T(e_1))$
   Events e_1 and e_2 are concurrent if e_1 was is produced before e_2 or vice–versa.

The relation $<$ defines a strict partial order [12] on a set of timestamps of primitive events. Then two events e_1 and e_2 can be ordered as follows:

e_1 *before* e_2*:*
$T(e_1).local < T(e_2).local \rightarrow T(e_1).global \leq T(e_2).global$
e_1 *simultaneous to* e_2*:*
$T(e_1).local = T(e_2).local \rightarrow T(e_1).global = T(e_2).global$
e_1 *concurrent to* e_2*:*
$T(e_1).local \sim T(e_2).local \rightarrow |\ T(e_1).global - T(e_2).global\ | \leq 1g_g$
(g_g represents the granularity of the global reference clock.)

It is worth noting that the difference between simultaneous and concurrent events consists in that simultaneous events are produced within the same location. For the ordering of composite events, the previous definitions are modified as follows [48]:

1) $T(e_1) < T(e_2) \iff (\forall\ t_2 \in T(e_2), \exists\ t_1 \in T(e_1))$ *such that* $(t_1 < t_2)$
   The composite event e_1 is triggered before the composite event e_2 if and only if for all events which trigger e_2 exists an event, which triggers e_1, that has been produced before.
2) Concurrent:
   $T(e_1) \sim T(e_2) \iff (\forall\ t_1 \in T(e_1), \forall\ t_2 \in T(e_2))$ *such that* $(t_1 \sim t_2)$
   The composite events e_1 and e_2 are concurrent if and only if all the triggering events of e_1 and e_2 are concurrent.
3) Not comparable:
   $T(e_1) \bowtie T(e_2) \iff \neg\ ((T(e_1) < T(e_2)) \vee (T(e_1) > T(e_2)) \vee (T(e_1) \sim T(e_2)))$
   The timestamps T(e_1) et T(e_2) of the composite events e_1 and e_2 are not comparable if it is impossible to determine an ordering relationship between e_1 and e_2.
4) $T(e_1) <\sim T(e_2) \iff T(e_1) \sim T(e_2) \vee T(e_1) < T(e_2)$
   The composite event e_1 is triggered approximately before the composite event e_2 if and only if they are concurrent or if e_1 is produced before e_2.

[11] The notation *T(e).*$< attribute >$ is used to denote the local (global) instant of occurrence and the timestamp location.

[12] Let the reader remind that an ordering relation $<$ on a set A defines a strict partial order if it is transitive and non–reflexive. It defines a total ordering if, moreover, $\forall$ *x,y,z* $\in A$ *either* $x < y$*, or* $x = y$*, or* $y < x$. In centralized systems, a strict total order of type $<$ is non-reflexive, transitive and asymmetric; on the contrary, a total order $\leq$ is reflexive, transitive and asymmetric.

Sistema Informático para Análisis de Cardiopatía Holter

Jesús Antonio Álvarez Cedillo, Juan Carlos Herrera Lozada y Patricia Pérez Romero

***Resumen*—El presente artículo, muestra los avances de desarrollo de una herramienta médica relacionada con los estudios de cardiopatía que esté disponible y al alcance de cualquier hospital, centro médico, o consultorio médico, que sea accesible en el costo, de fácil manejo y comprensible. Como un beneficio para el paciente, este proyecto le permitiría tener un mayor acceso a este tipo de estudios. Este proyecto también permite apoyar al médico profesional en cuanto a obtener en ciertos casos, un posible diagnóstico.**

***Palabras clave*—Holter electrocardiográfico, cardiopatía.**

Computer System for Analysis of Holter Cardiopathy

***Abstract*—This paper presents the development of a medical tool related to cardiopathy studies that is available and accessible to any hospital, medical center, or doctor's office, which is accessible at low cost, user friendly and understandable. As a benefit for patients, this project allows major accessibility of the corresponding medical studies. This project also allows to the medical professional obtaining in certain cases a possible diagnosis.**

***Index terms*—Holter electrocardiography, cardiopathy.**

I. Introducción

EN México, el estilo de vida acelerado en el que estamos inmersos ha sido motivo para que las enfermedades cardiovasculares tengan un mayor impacto entre la población. Y se han posicionado entre las principales causas de muerte en nuestro país.

Cualquier padecimiento del corazón o del sistema cardiovascular se puede registrar bajo el nombre de cardiopatía. Que abarca diversos padecimientos propios de la estructura del corazón. Dentro de estos padecimientos se encuentran las arritmias cardiacas. Las cuales son de gran importancia dentro de este proyecto.

La electrocardiografía, es un método para registrar gráficamente las señales eléctricas del corazón, utilizado ampliamente para detectar alteraciones en el ritmo cardiaco. Con este método es posible analizar el comportamiento del corazón del paciente, y apoyar al médico profesional a emitir el diagnóstico correcto.

Una de las herramientas importantes para este método de diagnóstico es el Holter electrocardiográfico, que pretende obtener un registro de la actividad eléctrica del corazón, por un periodo de tiempo, generalmente de 24 horas. A esta técnica se le denomina ambulatoria, ya que no es necesario estar en un hospital o en un consultorio médico. Esto con el fin de obtener datos más detallados, acerca de la actividad del corazón por un tiempo prolongado, y registra todas las señales en las actividades diarias y cotidianas de un individuo.

El electrocardiógrafo Holter, recoge los datos necesarios en unidades de almacenamiento, como puede ser una memoria USB o una ‘memory card’ con el fin de que, posteriormente, los datos puedan ser analizados por un médico.

Al utilizar un electrocardiógrafo Holter, se busca solucionar la deficiencia que se pueda tener en un electrocardiograma convencional, ya que en este último, pueden pasar desapercibidos algunos trastornos de la actividad eléctrica del corazón. El electrocardiógrafo Holter ofrece una lectura continua del ritmo cardiaco, de la frecuencia del corazón, y sus características eléctricas durante un periodo de 24 horas.

II. Diseño e Implementación del Sistema de Análisis

Para la implementación del piloto de la plataforma de educación virtual se tienen los siguientes requerimientos.

A. Hardware

La solución propuesta para este trabajo, fue implementar un sistema de adquisición de señales bioeléctricas, que principalmente consta de tres etapas importantes. La etapa de la alimentación, que es elemental para el funcionamiento del sistema, la etapa para la adquisición de la señal, y la etapa para la digitalización de dicha señal adquirida. Este sistema se elaboró, a partir del diagrama de bloques que se detalla a continuación en la figura 1.

1) Etapa para la Adquisición de la Señal

En la figura 2 se muestra el circuito que se implementó para adquirir la señal cardiaca.

Este circuito es un amplificador de instrumentación, y está constituido por dos seguidores de voltaje y un amplificador diferencial. Al momento de adquirir las pequeñas señales que provienen del corazón, pasan por este circuito, en donde son amplificadas y al mismo tiempo se disminuyen las señales de ruido, esto es posible porque el amplificador diferencial cuenta con una característica especial, tiene una muy baja ganancia en modo común, debido a su configuración. Los amplificadores operacionales que se utilizaron son de tecnología JFET, y presentan una alta impedancia de entrada y

Manuscrito recibido el 8 de febrero del 2010. Manuscrito aceptado para su publicación el 31 de mayo del 2010.

Los autores trabajan en el Centro de Innovación y Desarrollo Tecnológico en Cómputo del Instituto Politécnico Nacional, México, D.F. (e-mail: jaalvarez@ipn.mx, jlozada@ipn.mx, promerop@ipn.mx).

una corriente mínima de polarización, lo cual brinda un margen de seguridad eléctrica para el paciente.

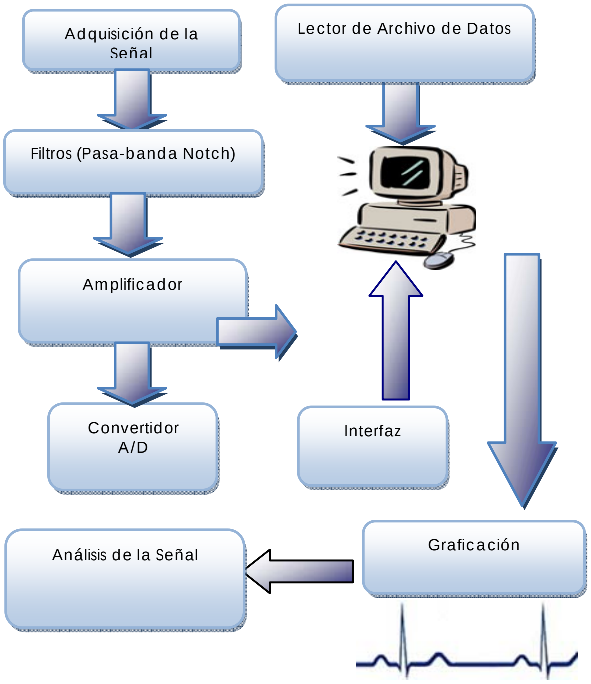


Fig. 1. Diagrama de bloques del sistema de adquisición de señales.

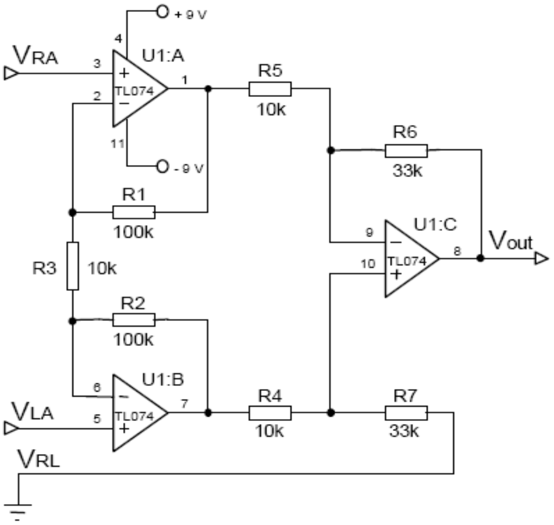


Fig. 2. Amplificador de instrumentación.

Para cada una de las etapas de este sistema de adquisición de señales, se realizó la simulación en el programa MULTISIM, para comprobar su buen funcionamiento y su modo de operación, antes de implementar el circuito físicamente. La figura 3, muestra la simulación de la etapa del amplificador de instrumentación.

En la figura 2, se muestra el circuito amplificador de voltaje, las entradas VRA y VLA, son los potenciales eléctricos de la mano derecha e izquierda respectivamente, y VRL es el potencial eléctrico que se mide en la pierna derecha, y se le utiliza como una referencia de los potenciales bioeléctricos.

Este circuito, se alimentó a la entrada con dos señales de 1 y 2 mV y a una frecuencia de 1 Hz, para que la frecuencia no salga de los límites normales, para frecuencias cardiológicas. El circuito funcionó de manera correcta, y la figura 4 muestra lo que se obtuvo en la pantalla del osciloscopio.

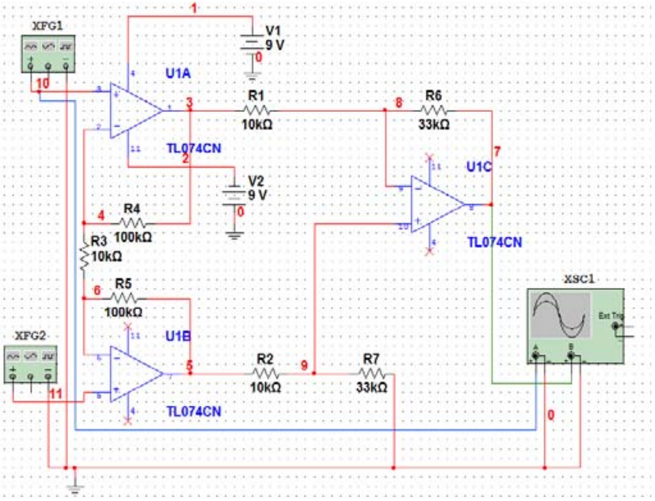
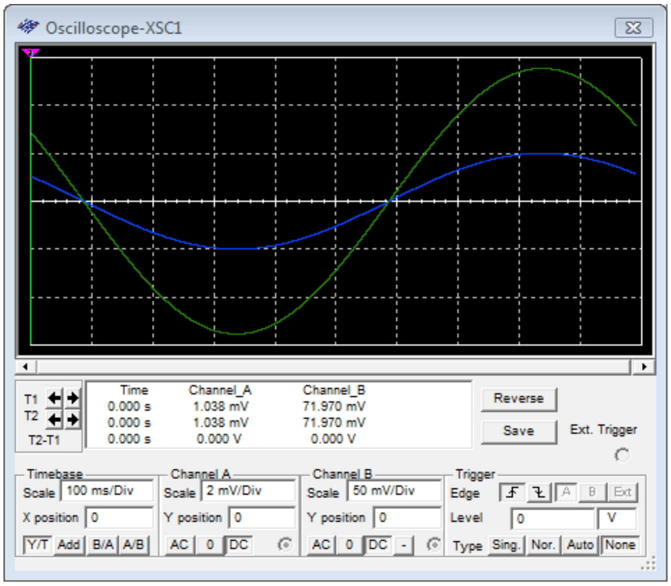


Fig. 3. Simulación de la etapa del amplificador de Instrumentación.



	Time	Channel_A	Channel_B
T1	0.000 s	1.038 mV	71.970 mV
T2	0.000 s	1.038 mV	71.970 mV
T2-T1	0.000 s	0.000 V	0.000 V

Timebase
Scale 100 ms/Div
X position 0

Channel A
Scale 2 mV/Div
Y position 0

Channel B
Scale 50 mV/Div
Y position 0

Trigger
Edge
Level 0 V
Type

Fig. 4. Señal de entrada tiene menor amplitud, señal de salida amplificada tiene mayor amplitud.

Para la implementación del amplificador de instrumentación, tanto R1 como R2 deben de ser de igual valor, porque haciendo esto se puede controlar la ganancia para esta parte del circuito, en conjunto con las resistencias R3 y R1 o R2. Para que se logre eliminar la señal de ruido también R6 y R7 deben ser iguales, de igual manera R5 y R4.

2) *Etapa de Filtro Pasa Banda*

La señal que se obtiene de la etapa anterior, debe pasar al filtro pasa banda, para asegurar que se encuentre dentro de la banda especificada por las normas médicas, que es entre 0.05 Hz y 100 Hz.

Se han realizado estudios que demuestran que las señales que tienen una frecuencia arriba de 100 Hz no son cardiológicas, y también, al filtrar las frecuencias que son menores de 0.05 Hz, se elimina una diferencia de potencial existente entre los electrodos y la superficie de la piel, que llegan a alcanzar niveles de hasta 300 mV, y esto puede ocasionar que se saturen los circuitos del amplificador. Al eliminar estas frecuencias, se puede asegurar una ganancia alta de la señal electrocardiográfica.

El circuito del filtro pasa banda se puede apreciar en la figura 5, en este circuito la resistencia R3 y el capacitor C2, funcionan como un filtro pasa altas y el valor de estos componentes, es lo que determina la frecuencia de corte inferior (fL), que es de 0.05 Hz. Se empleó la siguiente expresión para calcular los valores de los componentes.

$$fL = \frac{1}{2\pi R_3 C_2} \qquad (1)$$

De modo contrario la resistencia R2 y el capacitor C1, son los que forman el filtro pasa bajas que en este caso se necesita, este filtro define la frecuencia de corte superior (fH) de 100 Hz. A partir de la siguiente expresión, se encuentran los valores correspondientes a R2 y C1.

$$fH = \frac{1}{2\pi R_2 C_1} \qquad (2)$$

En esta etapa de filtrado, la señal obtiene una amplificación que es posible calcularla, si se eliminan los capacitores implicados. Esta operación se puede realizar solamente, debido a que en las frecuencias en las que opera el capacitor C2, funciona como un cortocircuito, y asimismo el capacitor C1 trabaja como un circuito abierto.

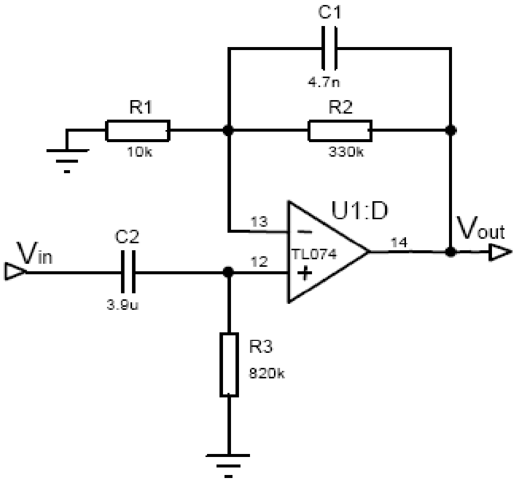


Fig. 5. Filtro pasa banda.

De esta manera el circuito se puede reducir a un amplificador no inversor, y su señal de salida se puede definir con la siguiente ecuación.

$$V_{OUT} = \left(1 + \frac{R_2}{R_1}\right) V_{IN} \qquad (3)$$

Empleando este circuito se logra amplificar la señal obtenida, y además se delimita la banda de frecuencia entre 0.05 Hz y hasta 100 Hz.

Aquí se muestra el circuito simulado con el programa MULTISIM.

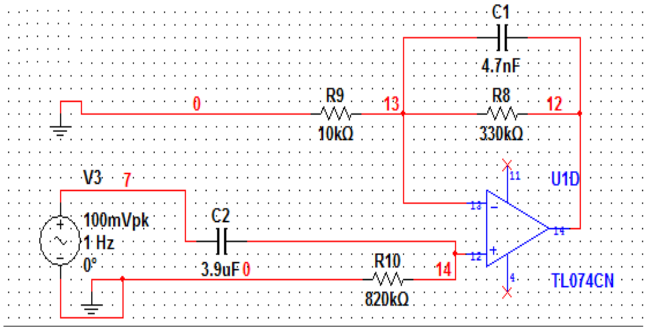


Fig. 6. Simulación de filtro pasa bajas.

3) *Etapa Filtro Rechaza Banda (Notch)*

Ya que se determinó el rango de frecuencias para la señal que se ha adquirido, lo que viene a continuación es implementar un filtro Notch e ingresar dicha señal. Esto se realiza porque la presencia del ruido en el registro de biopotenciales, no se puede evitar, y este tipo de filtro tiene la característica de eliminar señales de alguna frecuencia específica. El objetivo de realizar este filtro en este proyecto, es para eliminar ruido inducido a través de la red eléctrica, y demás aparatos como lámparas, computadoras, impresoras, y otros dispositivos que se alimentan de la red eléctrica doméstica de 60 Hz. Con esta información, se implementa el filtro Notch para la frecuencia de 60 Hz, eliminando así las señales de ruido que se producen a partir de esta frecuencia, y que distorsionan la señal electrocardiográfica.

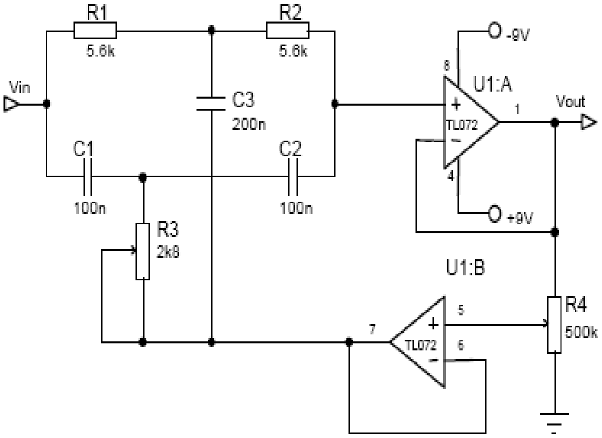


Figura 7. Filtro rechaza banda (Notch).

En la figura 7, se aprecia el filtro que se utilizó, en el cual el valor de R1 es igual a R2, al mismo tiempo que el valor de R3 es la mitad de éstos. Por otro lado los valores para los capacitores C1 y C2 es el mismo, y el de C3 es la suma de C1 y C2. El valor de la frecuencia que se desea eliminar se determina con la siguiente ecuación.

Simulación en MULTISIM del filtro rechaza banda o Notch se muestra en la figura 8.

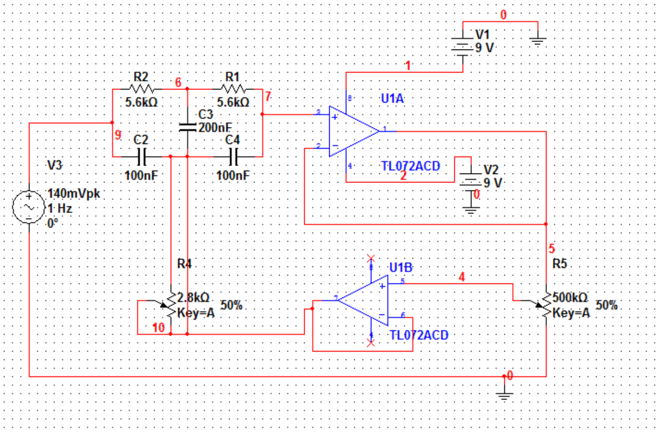


Fig. 8. Filtro rechaza banda.

4) *Etapa de Amplificación*

Ahora bien, ya que la señal ha pasado por todos los circuitos anteriores, necesita ser manipulada, para que pueda alcanzar una amplitud que se encuentre entre 0v y 5 v, esto es para poder digitalizarla con el ADC0809, que solamente acepta señales que estén comprendidas en este rango.

B. *Diseño del Software*

In software design, an important point to consider is the communication port to be used in this project is the parallel port of the computer, so below is a brief description of the most important in terms to this port for communication.

1) *Registros del puerto paralelo*

Este puerto recibe el nombre de paralelo porque tiene un bus de datos de 8 líneas, y además es posible escribir en él 8 bits al mismo tiempo. En las computadoras de escritorio, este puerto se encuentra en la parte posterior y es un conector DB25 hembra generalmente.

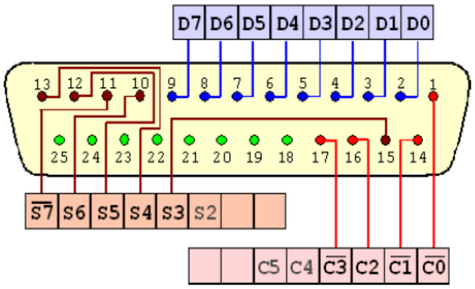


Fig. 9. Puerto Paralelo

Se podría profundizar de forma detallada, el uso específico de cada terminal del puerto paralelo, pero para este trabajo sólo es de interés conocer las terminales, en las que podemos escribir datos hacia el dispositivo, y en qué terminales podemos leer datos desde el hardware.

En la figura 9, se especifican principalmente tres registros:

- Datos (D0-D7) – tiene 8 terminales de salida,
- Estado (S2-S7) – tiene 5 terminales de entrada,
- Control (C0-C5) – tiene 4 terminales de salida,
- Tierra (18-25) – tiene 8 terminales aterrizadas.

Gracias a esta información, se observa que se puede utilizar el registro de Datos para escribir hacia el hardware, y el registro de Estado se utiliza para leer datos desde el hardware. Y las cuatro líneas de control, usualmente son salidas pero también se pueden utilizar como entradas, esto quiere decir que son modificables tanto por medio de software como por hardware.

C. *Implementación Física del Hardware*

Las imágenes a continuación, muestran la implementación física del sistema de adquisición de señales bioeléctricas del corazón (figuras 10, 11, 12), el cual se montó sobre una tablilla de pruebas (protoboard), y posteriormente se evaluaron los resultados, siendo éstos favorables, figura 13.

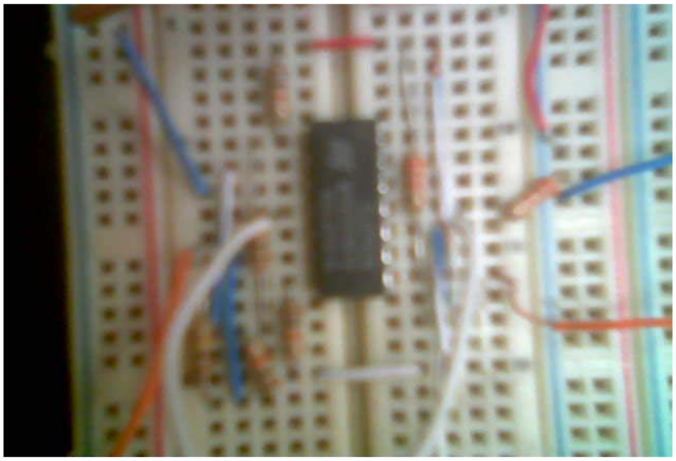


Fig. 10. Sistema de adquisición de señales biomédicas (1).

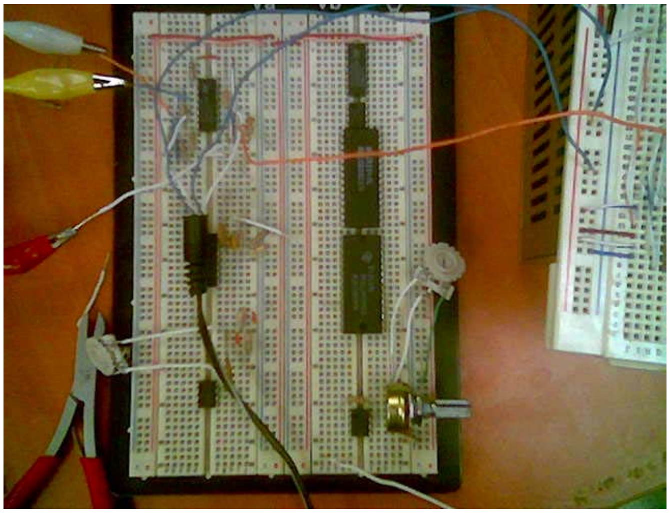


Fig. 11. Sistema de adquisición de señales biomédicas (2).

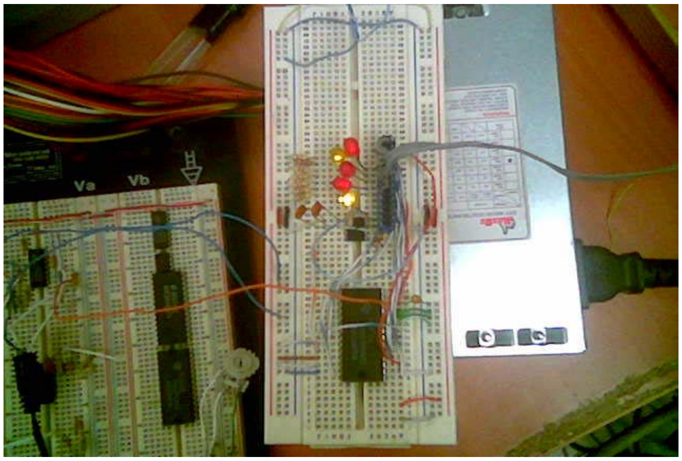


Fig. 12. Sistema de adquisición de señales biomédicas (3).

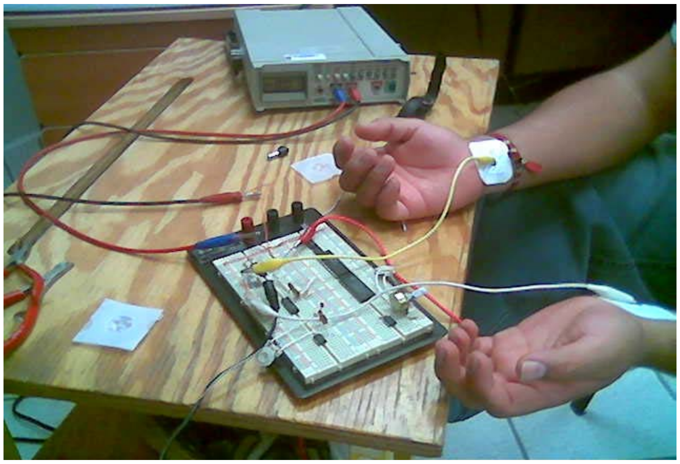


Fig. 13. Sistema de adquisición de señales biomédicas (4).

III. SOFTWARE

El lenguaje de programación que se eligió para desarrollar este proyecto de una manera visual amigable, fue Visual Basic 6.0. Las razones por la cual se eligió este entorno se mencionan brevemente:

- Es un leguaje sencillo y es fácil de aprender,
- Es un lenguaje popular,
- Existen diversos recursos para implementarlos en Visual Basic,
- Herramientas disponibles en Internet, como las librerías DLL o archivo OCX.

Para la implementación en código de este proyecto, es necesario contar con una librearía que trabaje con el puerto paralelo, las razones se explicarán a continuación.

Lectura y Escritura de datos en el puerto paralelo utilizando Visual Basic 6.0

Para realizar las operaciones de escritura y lectura en el puerto paralelo, utilizando el entorno de programación Visual Basic 6.0, es necesario controlar el puerto a través de una librería DLL, esto es, una librería de enlace dinámico, ya que Visual Basic 6.0 no cuenta con instrucciones propias para escribir o leer datos del puerto. Las librerías de enlace dinámico, forman parte de uno de los elementos primordiales del sistema operativo Windows. Básicamente las librerías DLL son archivos ejecutables independientes, que incluyen funciones y recursos para que puedan ser llamados por otros programas, e incluso por otras DLL, para llevar a cabo ciertos trabajos. No es posible ejecutar una DLL de manera independiente, sino que sólo se puede utilizar hasta que un programa u otra DLL, llamen a alguna de las funciones de la librería. El hecho de que sea una librería de “enlace dinámico”, hace referencia al código que contiene la DLL, es decir, al hecho de que el código que contiene la DLL se incorpora al programa ejecutable, y ésta es llamada sólo al momento en que es solicitada, esto es, en tiempo de ejecución.

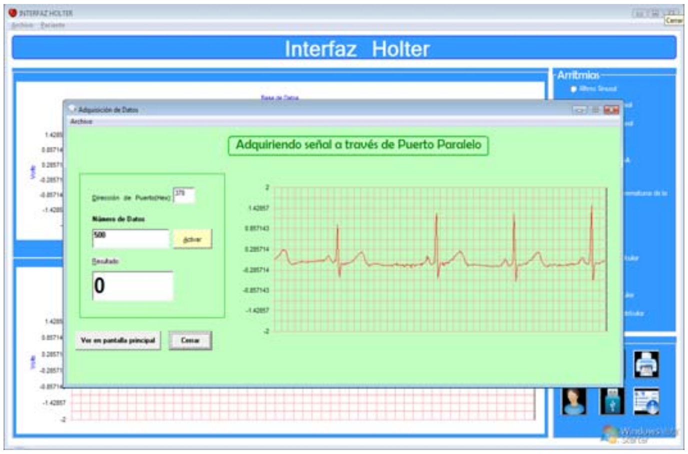
Dentro de la librería DLL existen funciones para controlar el puerto paralelo, y desde Visual Basic pueden ser fácilmente llamadas.

En este proyecto se trabajó con la librería *NTPort.dll*, la cual permite tener acceso a los puertos de entrada y salida de una computadora, sin la necesidad de utilizar el paquete Windows Drivers Development Kit (DDK).

Además la librería NTPort, brinda un soporte para los sistemas operativos Windows 95/98/Me y Windows NT/2000/XP/Server 2003.

A. *Interfaz Gráfica*

Para resolver de forma rápida y sencilla la estructura de esta aplicación, se desarrolló un diagrama de flujo, en el que se contemplan cada una de las etapas de adquisición de datos, y el proceso que conllevan. El diagrama de flujo, fue una herramienta de gran utilidad en este proyecto, para plantear los resultados que se deseaban obtener y cómo se alcanzarían. Principalmente se desarrollaron tres etapas importantes, dos para la adquisición de datos—éstas se describen más adelante—, y una para la base de datos, que servirá como punto de referencia, para el análisis de la señal electrocardiográfica adquirida, en cualquiera de las etapas anteriores. Se puede observar el diagrama de flujo en la figura 14.



Dirección de Puerto(Hex): 378

Número de Datos: 500

Resultado: 0

Fig. 15 a) Adquisición de datos por el puerto paralelo;
b) Datos en pantalla principal para su análisis.

La adquisición de las señales biomédicas a través del puerto paralelo en tiempo real, haciendo uso de la librería NTPort y el hardware del sistema de adquisición de datos, con el cual, el médico puede realizar una revisión al paciente que así lo requiera. Esto se muestra en la figura 15.

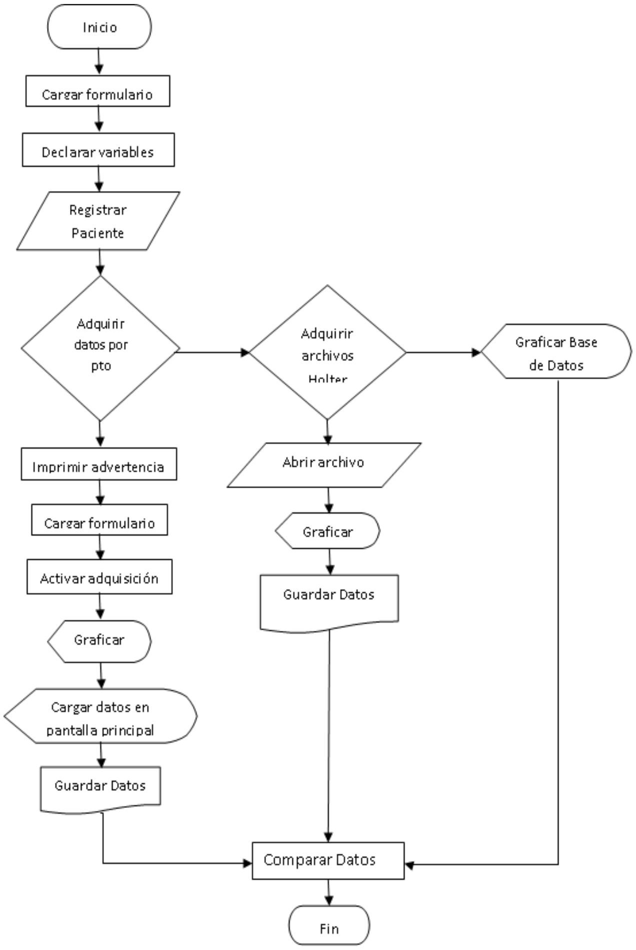


Fig. 14. Diagrama de flujo.

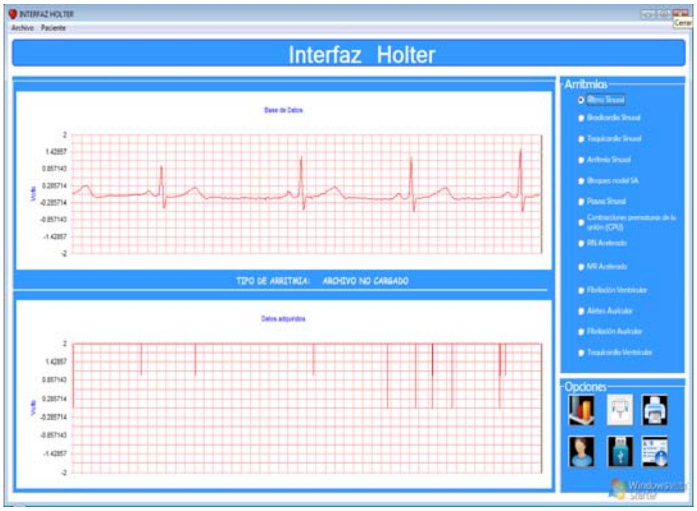


Fig. 16. Ritmo sinusal normal.

Por otra parte, se realizó la etapa de análisis de datos adquiridos, mediante archivos portables en algún dispositivo de almacenamiento, como una memoria USB o una 'memory card'.

Para este caso la aplicación realizada, tiene la opción para buscar el archivo creado por el dispositivo Holter, y cargarlo en una ventana para su graficación, con esto el médico puede identificar el tipo de anormalidad en el ritmo cardiaco del paciente.

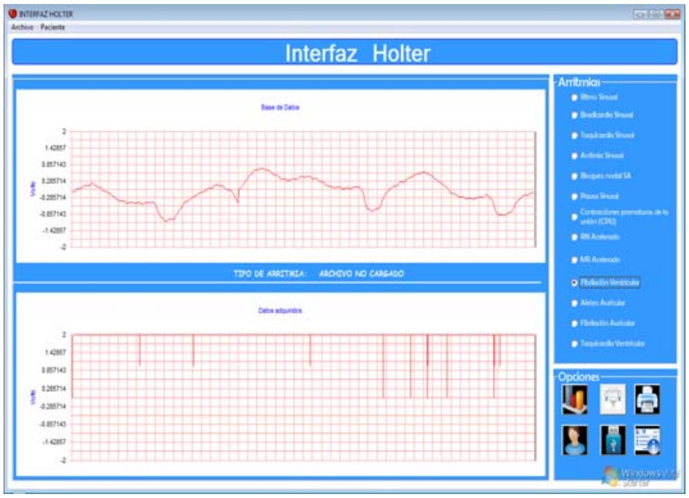


Fig. 17. Alteración cardiaca – Arritmia.

Para la ventana principal, se presenta una pantalla con las gráficas que muestran, según el caso, las señales para el análisis del médico, ya sea que la información se haya obtenido por medio de un archivo, o por el puerto paralelo.

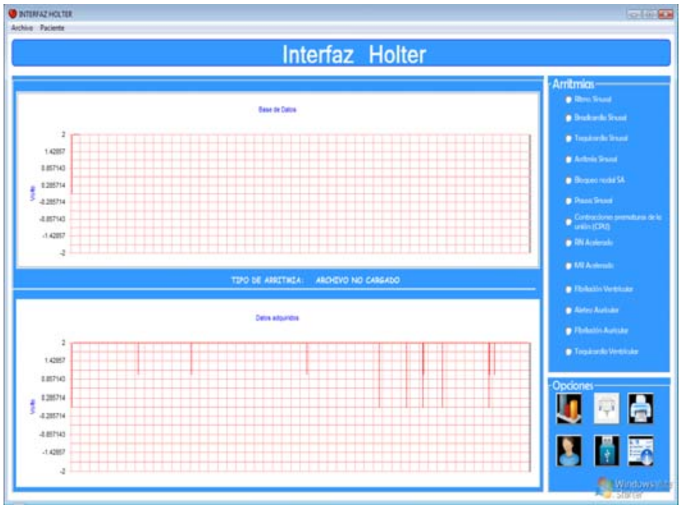
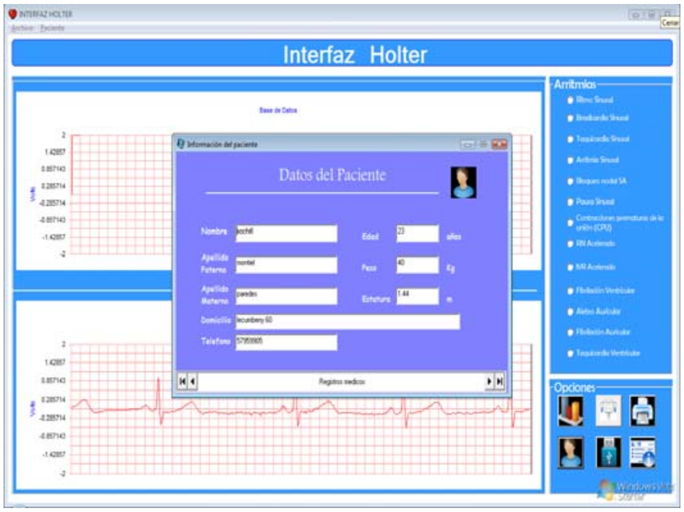


Fig. 18. Interfaz Holter.

También es posible, contar con los datos necesarios del paciente que se realiza el estudio Holter.

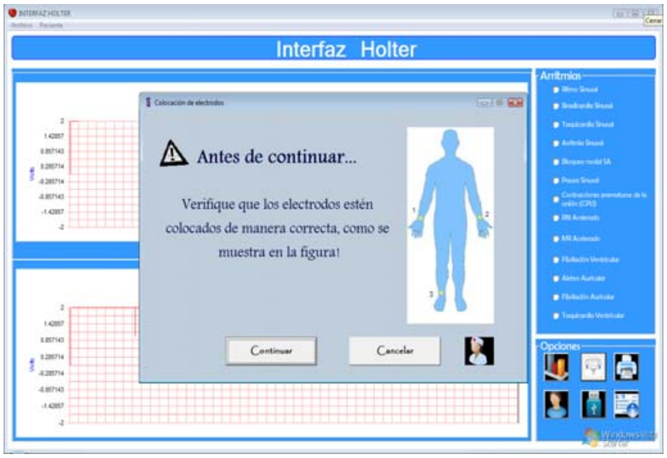
Antes de adquirir la señal a través del puerto paralelo, la aplicación lanza una advertencia, para verificar que el paciente tenga colocados los electrodos de manera correcta, como se muestra en la figura 20.

Se pueden observar tanto los datos adquiridos por el puerto paralelo, como los que se obtienen a través del dispositivo Holter, y al mismo tiempo tener una referencia, que sería la base de datos, para permitir al médico realizar un análisis de manera detallada. Comparando de manera visual las señales electrocardiográficas. Esto se puede apreciar en la ventana principal de la aplicación, la cual se muestra en la figura 21.



Nombre: xochil
Edad: 23 años
Apellido Paterno: montiel
Peso: 40 Kg
Apellido Materno: paredes
Estatura: 1.44 m
Domicilio: lecumbery 60
Telefono: 57959905

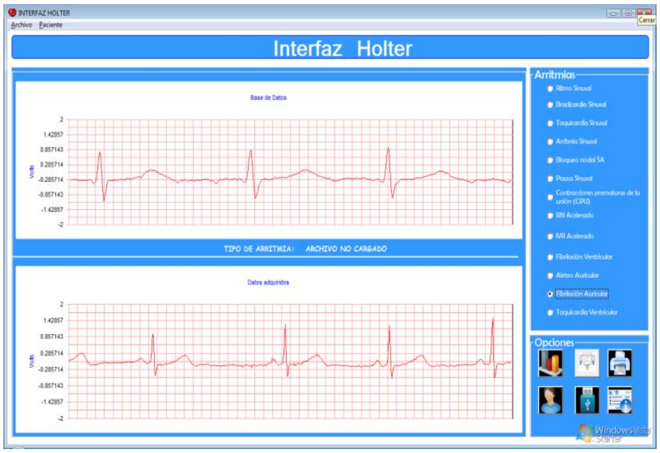
Fig. 19. Datos del paciente.



⚠ Antes de continuar...

Verifique que los electrodos estén colocados de manera correcta, como se muestra en la figura!

Fig. 20. Colocación de electrodos.



TIPO DE ARRITMIA: ARCHIVO NO CARGADO

Fig. 21. Pantalla principal para realizar comparaciones.

Se puede hacer uso de la base de datos, por medio del panel derecho, al seleccionar algún tipo de arritmia, ésta se visualizará en el gráfico superior. La figura 21, muestra lo que se ha descrito.

La aplicación ofrece otra opción, que es la de guardar el gráfico de los datos adquiridos, en un archivo de imagen, esto facilitará que cualquier señal electrocardiográfica guardada, pueda ser analizada posteriormente, si no es necesario hacerlo en el momento. Figura 22.

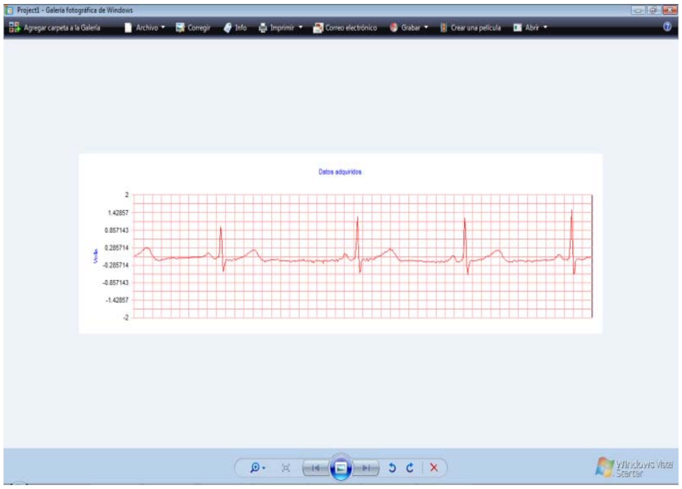


Fig. 22. Gráfica guardada como archivo de imagen.

IV. Conclusiones

En la realización de este proyecto se presentaron diferentes situaciones, y cada una implicaba resolver un conflicto, algunos más complejos que otros, en algunas etapas se tuvo que invertir un tiempo mayor del que estaba estipulado.

Al implementar el dispositivo, prácticamente no existieron problemas relevantes que impidieran el avance de este proyecto.

En cuanto al manejo del lenguaje Visual Basic 6.0, los problemas que se presentaron fueron en el manejo de ciertos controles, que se resolvieron fácilmente.

El software que se desarrolló, se llevó a cabo por etapas, de acuerdo a la planeación inicial y conforme se obtenían resultados exitosos. Como se mencionó al principio, uno de los objetivos fue realizar una aplicación que fuera amigable con el usuario, es decir, fácil de usar. Por esto se trabajó con muchos elementos visuales, para un mejor y más rápido aprendizaje acerca de la interfaz, por parte del personal que estará trabajando con ella.

Al finalizar este trabajo se logró el objetivo, que se planteó antes de iniciar la realización de este proyecto. Ya que al concluir se obtuvo un sistema de interpretación de datos, que está basado en un analizador de cardiopatías, mejor conocido como electrocardiógrafo Holter, y que cumple la función de ser una herramienta de apoyo médico.

Se aplicó la metodología médica, y se obtuvo la interfaz gráfica amigable con el usuario.

Con esto al finalizar este proyecto, se cuenta con una interfaz accesible y de bajo costo. Es un sistema confiable y muestra la respuesta de la actividad eléctrica del corazón, de manera fiel como lo haría cualquier electrocardiógrafo convencional.

V. Trabajo Futuro

Respecto a las mejoras que se le pueden hacer a este proyecto a futuro, la ciencia y la tecnología van en aumento, y cada vez más se desarrollan infinidad de aplicaciones, para cubrir las necesidades de los seres humanos, por lo tanto las modificaciones y mejoras que se puedan implementar en este proyecto, dependerán de las necesidades del usuario y de la imaginación del ingeniero a cargo del proyecto, y de esta forma se obtendrá la mejora de este software.

Una característica por mencionar, puede ser que el sistema de adquisición de datos, ya no trabaje utilizando el puerto paralelo, ya que cada vez más está en desuso, y está siendo reemplazado por nuevos dispositivos de comunicación con la computadora, de los cuales se puede echar mano.

Y en lugar de esto, realizar las modificaciones pertinentes para que pueda transmitir datos por vía USB, incluso se puede trabajar para que la transmisión de información sea posible, a través de infrarrojo o incluso utilizando la tecnología Bluetooth, con estas mejoras, sin lugar a dudas esta aplicación tendría una mayor aceptación, una mayor aplicación y grandes posibilidades de que se utilice, ya sea para innovar en nuevas aplicaciones, o para que el sistema tenga una mejora considerable en su rendimiento, y se adapte a las nuevas tecnologías.

Otra de las mejoras que se le podrían hacer a este proyecto, podría ser que transmitiera los datos o un informe detallado del paciente vía Internet, o también se podría implementar en un móvil o incluso en un dispositivo PDA.

Referencias

[1] C. M. Agulhari, R. M. R. Silveira, and I. S. Bonatti, *Lossless compression applied to sequences of bits*, Technical report, Unicamp, Brazil, 2007. Available: http://www.dt.fee.unicamp.br/~ivanil/lossless_bitmap_agulhari_2007.pdf.

[2] A. Alshamali and A. S. Al-Fahoum. “Comments on An efficient coding algorithm for the compression of ECG signals using the wavelet transform," *IEEE Transactions on Biomedical Engineering*, 50 (8), 1034-1037, Aug. 2003.

[3] C. Rodriguez, S. Borromeo, R. de la Prieta, J. A. Hernández, N. Malpica, "Wireless ECG based on Bluetooth protocol: design and implementation," in *Proc. of IEEE int. conf. on Information Technologies in Biomedicine*, Ionannina, Greece, Oct. 2006.

[4] P. Bifulco, G. Gargiulo, "Bluetooth Portable Device for Continuous ECG and Patient Motion Monitoring During Daily Life," *Medicon 2007 IFMBE Proceedings* 16, 2007, pp.369-372.

[5] V. Noparrat, P. Keeratiwintakorn "The Three-Lead Wireless ECG in Sensor Networks for Mobile Patients," in *SICE Annual Conference* Japan, August 20-22, 2008.

[6] Harri Kailanto, Esko Hyvärinen, and Jari Hyttinen, "Mobile ECG measurement and Analysis System Using Mobile Phone as the Base Station." in *Proc. of Second International Conference on Pervasive Computing Technologies for Healthcare,* 2008.

[7] F. Sufi, Q. Fang and I. Cosic, "ECG R-R Peak Detection on Mobile Phones," in *Proceedings of the 29th Annual International Conference of the IEEE EMBS Cité Internationale,* Lyon, France, August 23-26, 2007.

[8] S. Arslan and K. Kose, "A Design of DSPIC Based Signal Monitoring and Processing System," *Journal of Electrical and Electronics Engineering.* number 1, volume 9. 2009.

[9] F. Spadini and F. Vergari, "A Wireless and Context-Aware ECG Monitor : An iMote2 Based Portable System," *Computers in Cardiology* 35, pp. 997--1000, 2008.

[10] H. Ming , Z. Yajun , and H. Xiaoping, “Portable ECG Measurement Device based on MSP430 MCU,” in *Proceedings of the 2008 International Conference on BioMedical Engineering and Informatics*, May 27-30, 2008, p.667-671.

[11] J. Zhu , N. Rao , D. Liang , and W. Chen, “Design of Pre-processing Circuit for Wireless ECG Monitoring System,” in *Proceedings of the 2008 International Conference on BioMedical Engineering and Informatics*, May 27-30, 2008, pp. 598-602.

[12] A. Tahat, "Mobile Personal Electrocardiogram Monitoring System and Transmission Using MMS," in *Proceedings of the 7th International Caribbean Conference on Devices, Circuits and Systems,* Mexico, Apr. 2008, pp. 28-30.

[13] P. Frehill and D. Chambers. "Using Zigbee to Integrate Medical Devices," in *Proceedings of the 29th Annual International Conference of the IEEE EMBS Cité Internationale*, Lyon, France, August 2007, pp.23-26.

[14] S. Pavel, M. Pavlik, and R. Vrba, “Smart Differential Pressure Sensor with Bluetooth Communication Interface,” in *Proceedings of the Third International Conference on Systems*, April 13-18, 2008, pp.363-367.

[15] M. J. Moron, R. Luque, E. Casilari and A. Díaz-Estrella, "Analysis of Bluetooth Transmission Delay in Personal Area Networks," *IEEE*, 2008.

Predicción de Fallos en Redes IP empleando Redes Neuronales Artificiales

Gustavo A. García y Octavio Salcedo

***Resumen*—El presente artículo describe la implementación de un sistema de predicción de fallos en redes LAN (fallos de timeout y rechazo en las conexiones), utilizando redes neuronales artificiales Perceptrón Multicapa. Se describe como se implementó el sistema, las pruebas realizadas para la selección de los parámetros propios de la red neuronal, como del algoritmo de entrenamiento y los resultados de evaluación obtenidos.**

***Palabras clave*—Predicción de fallos, MIB, red neuronal artificial, perceptrón multicapa, backpropagation.**

Prediction of Failures in IP Networks using Artificial Neural Networks

***Abstract*—The paper presents the implementation of a system for predicting failures in LAN (timeout failure and rejection of connections), using neural networks (multilayer perceptron). It describes the implementation of the system, experiments conducted for the selection of specific parameters of the neural network, training algorithm and evaluation results.**

***Index Terms*—Prediction of failures, MIB, Artificial Neural Networks, multilayer perceptron, backpropagation.**

I. Introducción

LA idea principal en la predicción de fallos es predecir fallas catastróficas en la red, de manera que se pueda garantizar fiabilidad y calidad de servicio (QoS) en tiempo real para mantener la disponibilidad y fiabilidad de la red e iniciar apropiadas acciones de restauración de la "normalidad". Es por esto que surge la necesidad de implementar sistemas que por medio de análisis del tráfico de la red puedan predecir los fallos en servidores de archivos que se pudiesen presentar tales como time-out y rechazo de conexiones, existen diferentes técnicas de predicción que serán mencionadas en la siguiente sección, pero la utilizada en el sistema desarrollado está basada en redes neuronales artificiales, a las cuáles se les debe determinar de forma experimental y no teórica la arquitectura y los algoritmos de aprendizaje con los que se entrenará la red neuronal. A continuación se dará una breve introducción a las herramientas de predicción, posteriormente se mostrara el sistema de predicción de fallos de time-out y de rechazo de conexiones implementado, sus partes y las diferentes pruebas realizadas para encontrar los parámetros del sistema que brinden un mejor desempeño.

II. Marco Conceptual

A. *Herramientas Empleadas en la Predicción de Fallos*

Existes diferentes tipos de herramientas empleadas en la predicción tales como:

1) *Redes Neuronales Artificiales*

Según Simón Haykin [13] "Una red neuronal es un procesador masivamente paralelo distribuido que es propenso por naturaleza a almacenar conocimiento experimental y hacerlo disponible para su uso.

Este mecanismo se parece al cerebro en dos aspectos:

- El conocimiento es adquirido por la red a través de un proceso que se denomina aprendizaje.
- El conocimiento se almacena mediante la modificación de la fuerza o peso sináptico de las distintas uniones entre neuronas".

Las neuronas artificiales se conocen también como unidades de proceso, y su funcionamiento es simple, pues consiste en recibir en las entradas las salidas de las neuronas vecinas y calcular un valor de salida, el cual es enviado a todas las células restantes. Existen tres tipos de células o unidades [1]:

- Neuronas de entrada: reciben señales desde el entorno; estas entradas (que son a la vez entradas a la red) provienen generalmente de una serie de tiempo con datos anteriores al que se pretende predecir, resultado generalmente de preprocesamientos tales como normalizaciones, derivadas, umbralizaciones entre otros.
- Neuronas de salida: Las unidades de salida envían una señal fuera de la red; en la aplicación de predicción la salida correspondería al valor futuro o estimado.
- Neuronas ocultas: Son aquellas cuyas entradas y salidas se encuentran dentro del sistema; es decir, no tienen contacto con el exterior. Las redes neuronales pueden aprender de experiencias que son provistas como entrada-salida de la red sin necesidad de expresar la relación exacta entre la(s) entrada(s) y la salida, éstas pueden generalizar la experiencia aprendida y obtener la salida correcta cuando nuevas situaciones son encontradas [4].

Manuscripto recibido el 23 de enero de 2010. Manuscrito aceptado para su publciación el 31 de mayo de 2010.

Gustavo A. García es docente de la Universidad Central, Bogotá D.C., Colombia y candidato a Magíster en Ciencias de la información y las comunicaciones de la Universidad Distrital Francisco José de Caldas, Colombia (e-mail: gustavoandresg@gmail.com).

Octavio Salcedo es docente de la Universidad Distrital Francisco José de Caldas, Bogotá D.C, Colombia y candidato a PhD en ingeniería Informática Universidad de Salamanca, Colombia (e-mail: octavionetworking@gmail.com).

2) *Modelos autorregresivos (AR)*

Son modelos comúnmente usados para describir señales de series de tiempo estocásticas no estacionarias, y su característica principal es que van más allá de medidas estadísticas como la media y la varianza [5] [2], un modelo autorregresivo como lo menciona Proakis, es un proceso de solo polos cuya función de transferencia en Z se muestra en la ecuación 1 el cual es denominado proceso autorregresivo de orden p [3].

$$H(Z)=\frac{1}{1+\sum_{k=1}^{p} a_k Z^{-k}} \qquad (1)$$

3) *Autómatas de Aprendizaje*

De acuerdo a Kyriakakos *et al*. [14] los autómatas de aprendizaje (LA por sus siglas en inglés), son sistemas adaptativos de estados finitos que interactúan continuamente con un ambiente general. A través de la respuesta de un proceso probabilístico de ensayo y error, los LA aprenden a escoger o a adaptarse a un comportamiento que genera la mejor respuesta. Como primer paso de un proceso de aprendizaje una entrada es provista al autómata del medio en que se encuentra, esta entrada acciona uno de los posibles candidatos (estados) del autómata, el medio recibe y evalúa la respuesta, luego provee retroalimentación al autómata la cual altera la respuesta al estímulo del autómata [6]. Los autómatas de aprendizaje son generalmente considerados sistemas robustos pero no aprendices eficientes, son fáciles de implementar y cuyo funcionamiento generalmente se basa en un matriz de estados de transición, que contiene las probabilidades de transición de un salto, estando en el estado i al estado $j(P_{ij})$ [6].

4) *Circulant Markov Modulated Poisson Process (CMMP)*

Esta herramienta captura no sólo las estadísticas de segundo orden como lo hacen los procesos autorregresivos de media móvil (ARMA) sino que también las estadísticas de primer orden cuya distribución puede ser diferente a la Gaussiana, la técnica para construir dicho proceso se explica en detalle en [7]. Y Sang en su artículo [8] describe la manera de cómo emplear dicha herramienta en la predicción de tráfico comparando dicha investigación con los resultados obtenidos con un modelo ARMA.

Estos sistemas de predicción, muestran que ellos son comúnmente correlacionados [9] y el análisis de datos en redes de sistemas de gran escala, revela patrones de tiempo del día y día de la semana. Este tipo de correlación es comúnmente utilizado en proyectos de predicción como los desarrollados por Liang *et al*. [10], en el que analizaron los logs del supercomputador IBM BlueGene/L, con los que desarrollaron el sistema de predicción utilizando correlaciones temporales con los eventos de fallas presentados en las series de tiempo. Sahoo et al. [11], estudió las ocurrencias de eventos críticos en un cluster, realizando dos sistemas de predicción de fallos uno para cada uno, sin considerar que ellos se encontraban interrelacionados [9]. Wu et al [12], detectó fallas de nodo amplio en entre un ambiente de cluster, en donde la correlación temporal de los estados de los nodos fue usada para definir si el funcionamiento era normal.

III. Desarrollo Metodológico

En la figura 1 se muestra el sistema de predicción propuesto, el cual realizará predicciones de fallos de time-out y de conexiones rechazadas al servidor FTP tomando como entrada el conjunto de variables MIB (*Management Information Base*): *IpInReceives*, *IpInDelivers*, *IpOutRequests*, *tcpActiveOpens* y *tcpRetransSec*. Estas variables son tomadas utilizando un agente SNMP en el servidor FTP que se encuentra en la red mostrada en la figura 2 y son utilizadas en la etapa de preprocesamiento del sistema en donde se obtienen datos estadísticos de ellas (medias y desviaciones estándar) para posteriormente pasárselos a las neuronas de entrada de la red neuronal.

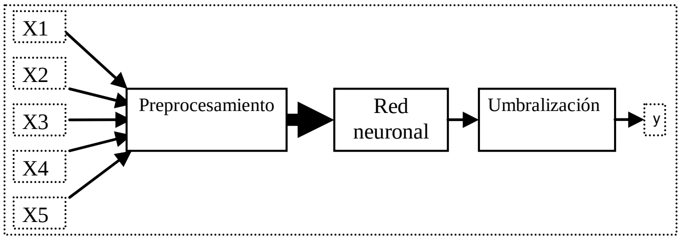


Fig 1 Sistema de predicción de fallos propuesto.

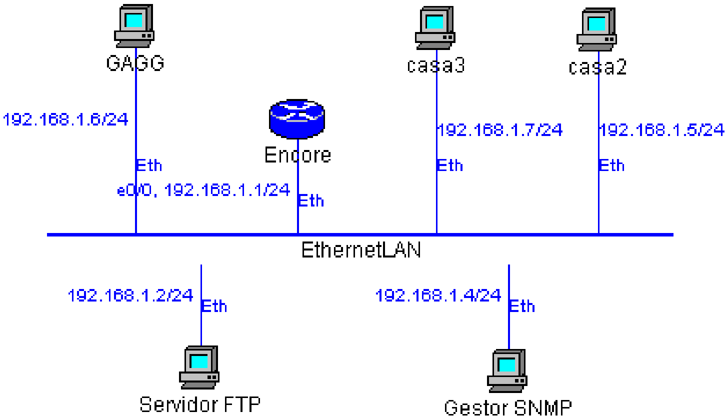


Fig 2. Red LAN de pruebas.

Para encontrar la configuración del sistema de predicción que presente el mejor desempeño en la predicción de fallos se realizaron pruebas de cada una de las siguientes variables del sistema.

- Número de épocas de entrenamiento,
- Variables de entrada a la red neuronal,
- Arquitectura de la red neuronal,
- Algoritmo de aprendizaje de la red neuronal,
- Parámetros del algoritmo de aprendizaje,
- Selección del nivel de umbralización.

A. *Segmentación de los Datos de Entrada*

Para encontrar los parámetros del sistema de predicción propuesto (ver figura 1), se contó con una base de datos que contenía muestras del tráfico entrante y saliente del servidor

FTP, así como sus conexiones activas y cantidad de retransmisiones (variables MIB mencionadas anteriormente), la base de datos contiene información de un mes de pruebas en la red y para motivos de entrenamiento y validación de la eficiencia del sistema propuesto, se segmentó el total de la base de datos en tres partes: 60% para entrenamiento de la red neuronal, 20% para validación del entrenamiento y el otro 20% para pruebas del sistema. Éste último fue utilizado para determinar la eficiencia del sistema implementado, ya que este segmento de datos no fue utilizado en el entrenamiento de la red neuronal, siendo desconocido por el sistema.

B. Selección del Número de Épocas de Entrenamiento de la Red Neuronal

La configuración de la red neuronal con la que se probó el número de épocas se encuentra en la tabla I, dicho parámetro fue variado desde un valor de 500 a 4500 en pasos de 500 épocas. Por cada época se realizaban 10 pruebas para verificar la repetitibilidad y consistencia de los resultados. Se seleccionó a 2000 épocas (ver tabla II) como el valor que mejor desempeño presentó. Como base para esta selección se tuvo en cuenta que la idea no es tener el menor error de entrenamiento sino que sea pequeño y que no requiera muchas épocas ya que esto afecta los tiempos de entrenamiento considerablemente.

TABLA I
PRUEBA DE SELECCIÓN DEL NÚMERO DE ÉPOCAS DE ENTRENAMIENTO.

No. Neuronas ocultas	8
Algoritmo de entrenamiento	BackPropagation (traingd)
Learning rate	0.05
No. neuronas de salida	1
No. neuronas de entrada	5
Tipo de red	Perceptrón Multicapa
No. épocas	500 – 4500
No. pruebas x época	10
Total pruebas	90

TABLA II
RESULTADOS DE PRUEBAS DE NÚMERO DE ÉPOCAS.

No Épocas	MseMin Prom	UltMse Prom
500	0,1143	0,114323222
1000	0,112908222	0,112908222
1500	0,112372556	0,112372556
2000	0,111893556	0,111893556
2500	0,110850333	0,110850333
3000	0,110573111	0,110573111
3500	0,110519444	0,110519444
4000	0,110476333	0,110476333
4500	0,110439889	0,110439889

C. Selección del Número de Entradas a la Red Neuronal

En esta etapa se realizaron pruebas con diferentes entradas a la red neuronal para determinar cuáles generaban un mejor desempeño en el sistema de predicción. Las entradas a la red neuronal probadas fueron: las variables MIB *IpInReceives*, *IpInDelivers*, *IpOutRequests*, *tcpActiveOpens*, *TcpRetransSec*; los valores medios, desviaciones estándar y valores anteriores de las variables *IpInReceives*, *IpInDelivers* e *IpOutRequests*. La conformación de las diversas entradas a la red neuronal, son generadas en la etapa de preprocesamiento (ver figura 1).

TABLA III
CONFIGURACIÓN DE LA RED NEURONAL, PRUEBAS DE LAS ENTRADAS.

No. Neuronas ocultas	8
Algorítmo de entrenamiento	BackPropagation con momento (traingd)
Learning rate	0.05
Momento	0.04
No. neuronas de salida	1
No. neuronas de entrada	5 a 26
Tipo de red	Perceptrón Multicapa
No. épocas	2000
No. pruebas x conjunto de entradas	10
Total pruebas	50

En la tabla III se encuentra la configuración de la red neuronal utilizada en las pruebas para las diferentes entradas. En el proceso de verificación de resultados por ser una prueba muy importante en el sistema de predicción, se realizó la verificación visual de las salidas del sistema. La figura 2 muestra las salidas del sistema en la que la línea verde cuando toma el valor de uno, representa los puntos en los cuáles el sistema debe indicar que se va a presentar un fallo en el servidor, y las líneas azules son aquellas en las que el sistema predijo que se iba a presentar una fallo en el servidor FTP. En esta figura la imagen *DatosOriMediaDesv* correspondiente a la salida del sistema se obtuvo que las entradas dadas por: las variables MIB, los valores medios y las desviaciones estándar de los últimos veinte minutos de adquisición, fueron las que mejor comportamiento presentaron utilizando como criterio la cantidad de veces que el sistema predijo correctamente vs. la cantidad de predicciones erradas.

Los resultados de las pruebas de variables de entrada a la red neuronal se presentan en la Figura 2, en el Anexo.

D. Selección del Número de Neuronas Ocultas en la Red Neuronal

La configuración empleada en la selección del número de neuronas ocultas se muestra en la tabla IV donde se observa que los resultados obtenidos hasta el momento fueron incluidos en las pruebas. Se realizaron pruebas variando el número de neuronas ocultas de dos a quince, y cuyos resultados se resumen en la tabla V. En esta tabla de resultados se adicionaron parámetros que ayudan en la selección de las pruebas que presentó mejor comportamiento en la predicción de fallos, no sólo teniendo en cuenta la cantidad de predicciones acertadas (*nOk*) sino también las erradas (*nErradas*).

La configuración empleada en la selección del número de neuronas ocultas se muestra en la tabla IV donde se observa que los resultados obtenidos hasta el momento fueron incluidos en las pruebas. Se realizaron pruebas variando el número de neuronas ocultas de dos a quince, y cuyos resultados se resumen en la tabla V. En esta tabla de resultados se adicionaron parámetros que ayudan en la selección de las pruebas que presentó mejor comportamiento en la predicción de fallos, no sólo teniendo en cuenta la cantidad de predicciones acertadas (*nOk*) sino también las erradas (*nErradas*).

TABLA IV
CONFIGURACIÓN DE LA RED NEURONAL
PARA PRUEBAS DE NEURONAS OCULTAS.

No. Neuronas ocultas	2-15
Algorítmo de entrenamiento	BackPropagation con momento (traingdm)
Learning rate	0.05
Momento	0.04
No. neuronas de salida	1
No. neuronas de entrada	11
Tipo de red	Perceptrón Multicapa
No. épocas	2000
No. pruebas x neurona	10
Total pruebas	140

Los resultados de las pruebas de selección de número de neuronas en la capa oculta se presentan en la Tabla V en el Anexo.

Se seleccionó la configuración de la red neuronal con 11 neuronas en la capa oculta, ya que el valor promedio de la relación entre el número de predicciones acertadas vs. el número de predicciones erradas es el cuarto más alto, pero la desviación estándar es la más baja de éstas, lo que representa una mayor homogeneidad de las predicciones, adicionalmente verificando la cantidad de predicciones acertadas vs. las erradas de la prueba en especifico se ve que esta prueba tuvo el 61 % de predicciones acertadas esta configuración.

E. Selección del Algoritmo de Entrenamiento

Se realizaron pruebas con los siguientes algoritmos de entrenamiento:

- **traingd:** Backpropagation de gradiente descendente;
- **traingdm:** Backpropagation de gradiente descendente y momento;
- **traingda:** Backpropagation de gradiente descendente con tasa de entrenamiento adaptativa;
- **trainrp:** *Resilient* Backpropagation.

A cada uno de los algoritmos se le realizaron diez pruebas para determinar cuál de los cuatro algoritmos probados presenta el mejor comportamiento en la predicción de fallos. En la tabla VI se muestra la configuración de la red neuronal como se realizaron las pruebas y en la tabla VII el resumen de resultados de éstas. En esta última se puede observar que el algoritmo seleccionado es el algoritmo **traingdm** (Backpropagation de gradiente descendiente y momentum**)** aunque el que mayor índice de predicciones correctas vs predicciones erradas fue el **trainrp.** La razón obedece a que este último presenta en promedio un número elevado de predicciones erróneas como lo son 389 predicciones erradas vs. las 204 erradas que presentó el algoritmo seleccionado.

TABLA VI
CONFIGURACIÓN DE LA RED NEURONAL,
PRUEBAS DEL ALGORITMO DE APRENDIZAJE.

No. Neuronas ocultas	11
Algorítmo de entrenamiento	traingd, traingdm, traingda, trainrp
Learning rate	0.05
Momento	0.04
No. neuronas de salida	1
No. neuronas de entrada	11
Tipo de red	Perceptrón Multicapa
No. épocas	2000
No. pruebas x Algorítmo	10
Total pruebas	40

Los resultados de las pruebas de la selección del algoritmo de entrenamiento se presentan en la Tabla VII, en el Anexo.

F. Selección del Momento en el Algoritmo de Entrenamiento

Después de haber encontrado que el algoritmo de entrenamiento de la red neuronal es el **traingdm**, sus parámetros son: el momentum y la tasa de aprendizaje (*learning rate*). A continuación en la tabla VIII se puede ver la configuración de la red neuronal con la que se probó el parámetro momento y en la tabla IX se observa el resumen de resultados de la prueba. El parámetro momentum seleccionado fue el de 0.10 ya tiene un índice de predicción alto (del 0.68), el número de predicciones es mucho mayor al de 0.72 que sólo tuvo 72 predicciones acertadas y el número de fallos no fue muy elevado (fue menor a 250 fallos que en la práctica se observó que era un número poco eficiente para la funcionalidad del sistema).

TABLA VIII
CONFIGURACIÓN DE LA RED NEURONAL,
PRUEBAS DE SELECCIÓN DE MOMENTUM.

No. Neuronas ocultas	11
Algorítmo de entrenamiento	BackPropagation con momento (traingdm)
Learning rate	0.05
Momento	0.01 - 0.15
No. neuronas de salida	1
No. neuronas de entrada	11
Tipo de red	Perceptrón Multicapa
No. épocas	2000
No. pruebas x momento	10
Total pruebas	150

Los resultados de las pruebas de momentum del algoritmo de aprendizaje se presentan en la Tabla IX en el Anexo.

G. Selección de la Tasa de Aprendizaje en el Algoritmo de Entrenamiento

Como se mencionó anteriormente el otro parámetro a definir en el algoritmo de aprendizaje es la tasa de entrenamiento, a continuación en la tabla X se encuentra la configuración de la red neuronal para las pruebas y en la tabla XI los resultados. En esta última se encuentra seleccionado la tasa de aprendizaje de 0.04 el cual comparado con los demás resultados que presentan el indicador de predicciones correctas vs. incorrectas.

TABLA X
CONFIGURACIÓN DE LA RED NEURONAL,
PRUEBAS DE SELECCIÓN DE LA TASA DE APRENDIZAJE.

No. Neuronas ocultas	11
Algorítmo de entrenamiento	BackPropagation con momento (traingdm)
Learning rate	0.01 - 0.15
Momento	0.01
No. neuronas de salida	1
No. neuronas de entrada	11
Tipo de red	Perceptrón Multicapa
No. épocas	2000
No. pruebas x learning rate	10
Total pruebas	150

Los resultados de las pruebas de selección de la tasa de aprendizaje del algoritmo de aprendizaje se presentan en la tabla XI en el Anexo.

H. Selección de Umbral

Como se observa en la figura 1, la umbralización es la última etapa del sistema de predicción y juega un papel muy importante en el sistema ya que es la encargada de seleccionar cuáles salidas de la red neuronal serán consideradas como un fallo en la red (Fallo en el servidor FTP) y cuáles no.

TABLA XII
CONFIGURACIÓN DE LA RED NEURONAL,
PRUEBAS DE SELECCIÓN DE UMBRAL.

No. Neuronas ocultas	11
Algorítmo de entrenamiento	BackPropagation con momento(traingdm)
Learning rate	0.04
Momento	0.01
No. neuronas de salida	1
No. neuronas de entrada	11
Tipo de red	Perceptrón Multicapa
No. épocas	2000
No. pruebas x umbral Umbral	10 1:0.1:2.5
Total pruebas	160

Es por esto que se realizaron pruebas para determinar el nivel de umbral con el que el sistema de predicción de fallos presenta un nivel alto de desempeño, en la tabla XII se encuentra la configuración de la red neuronal de las pruebas y en la tabla XIII (ver Anexo) los resultados.

Se escogió el umbral de 2.3 ya que éste presenta un nivel alto de predicciones correctas vs. las erradas (0.56) (comparado con las demás pruebas) y adicionalmente tiene desviación estándar baja y el promedio del indicador alto.

IV. RESULTADOS

En la tabla XIV se encuentra la configuración final del sistema de predicción de fallos propuesto en la figura 1, en ella se resume el trabajo desarrollado en la investigación para la configuración del sistema de preprocesamiento y en la tabla XIII se encuentran los resultados de las pruebas de predicción con datos de prueba diferentes a los de entrenamiento de la red neuronal. En ella se puede ver que el sistema presenta un índice de predicciones correctas vs. las erradas de un 66%, lo que nos dice que las redes neuronales perceptrón multicapa son herramientas válidas para la predicción de fallos en redes LAN aunque se debe buscar otra arquitectura de red que permita mejorar el desempeño del sistema.

TABLA XIV
CONFIGURACIÓN FINAL DEL SISTEMA DE PREDICCIÓN.

No. Neuronas ocultas	11
Algorítmo de entrenamiento	BackPropagation con momento (traingdm)
Learning rate	0.04
Momento	0.01
No. neuronas de salida	1
No. neuronas de entrada	11
Tipo de red	Perceptrón Multicapa
No. épocas	2000
No. pruebas x learning rate	10
Umbral	2.3
Total pruebas	40

En la prueba 36 como se observa en la tabla XV se obtuvo un sistema de predicción en el que el número de predicciones incorrectas es menor a 100 y el número de predicciones correctas fue de 65. La figura 3 muestra el resultado de esta prueba siendo la línea verde (punteada) en uno el intervalo de tiempo en el cual el sistema debe realizar predicciones y las líneas azules (líneas discontinuas) los momentos en los que el sistema realiza la predicción.

TABLA XV
RESULTADOS DE PRUEBAS DEL SISTEMA DE PREDICCIÓN.

Prueba	MseEntMin	nErradas	nOk	(nOk/nErradas)
1	0.109845	266	148	0,56
2	0.110150	225	113	0,50
3	0.111292	91	33	0,36

Prueba	MseEntMin	nErradas	nOk	(nOk/nErradas)
4	0.110363	182	110	0,60
5	0.110100	1	0	0,00
6	0.111106	2	0	0,00
7	0.110305	45	14	0,31
8	0.110122	4	2	0,50
9	0.110485	1	0	0,00
10	0.110099	176	83	0,47
11	0.110705	243	102	0,42
12	0.111991	1	0	0,00
13	0.110137	23	10	0,43
14	0.110764	107	70	0,65
15	0.110488	21	15	0,71
16	0.110766	191	77	0,40
17	0.109571	5	3	0,60
18	0.110636	244	104	0,43
19	0.110650	13	4	0,31
20	0.110410	215	103	0,48
21	0.111930	117	49	0,42
22	0.110180	185	85	0,46
23	0.110988	48	33	0,69
24	0.109985	18	12	0,67
25	0.109250	170	91	0,54
26	0.110232	21	17	0,81
27	0.110769	1	0	0,00
28	0.109736	1	0	0,00
29	0.109395	29	23	0,79
30	0.109827	1	0	0,00
31	0.109284	1	0	0,00
32	0.110412	38	28	0,74
33	0.110645	9	4	0,44
34	0.109890	35	23	0,66
35	0.110058	134	53	0,40
36	0.110605	99	65	0,66
37	0.110693	41	16	0,39
38	0.110444	22	13	0,59
39	0.111245	126	61	0,48
40	0.112721	160	71	0,44

V. Conclusiones

- Para el desarrollo de sistemas de predicción de fallos en redes se requiere que sean de baja complejidad computacional para que el tiempo utilizado en la predicción de fallos permita que el sistema sea implementable.
- Las redes neuronales perceptrón multicapa son una herramienta útil en la predicción de fallos, aunque se deben probar otros algoritmos de entrenamiento para mejorar el desempeño del sistema de predicción obtenido.
- Las variables MIB *IpInreceives*, *IpIndelivers*, *IpOutRequests*, *TcpActiveOpens*, *tcpRetranSec* permiten determinar las fallas de una red LAN, utilizando sus valores medios y desviaciones estándar a la entrada de la red neuronal.
- Con las entradas a la red neuronal *ipInreceives*, *ipIndelivers*, *ipOutRequests*, *tcpActiveOpens*, *tcpRetranSec*, sus valores medios y desviaciones estándar, la arquitectura que mejor se comporta en la predicción es la que tiene once neuronas en la capa oculta.
- De los algoritmos backpropagation de gradiente descendente, backpropagation de gradiente descendente y momento, backpropagation de gradiente descendente con tasa de entrenamiento adaptativa y resilient backpropagation, el mejor para la predicción de fallos de una red LAN utilizando red neuronal perceptrón multicapa de once neuronas en la capa oculta, es el backpropagation de gradiente descendente y momento, con parámetros tasa de entrenamiento de 0.04 y un momentum de 0.01.
- Para la determinación del fallo de la red LAN, el mejor parámetro para la umbralización de la salida de la red neuronal es que sea de 2.3 veces la desviación estándar de las últimas 120 salidas de ésta.
- Se podría mejorar el desempeño del sistema propuesto utilizando una arquitectura de red neuronal diferente o algoritmos que permitan un mejor aprendizaje a la red neuronal.

Referencias

[1] J. R. Hilera y V. J. Martínez, *Redes Neuronales Artificiales*, Alfaomega, 2000.

[2] G. Box, G. M. Jenkins, and G. Reinsel, *Time Series Analysis, Forecasting and Control*, Holden Day Series, 1976.

[3] J. G. Proakis y D. G. Manolakis, *Tratamiento Digital de Señales*, Pretince–Hall, 1998.

[4] Yen-Chieh Ouyang and Li-Bin Yeh, "Predictive bandwidth control for mpeg video: A wavelet approach for self-similar parameters estimation," in *IEEE International Conference on Communications ICC 2001*, vol. 5, 2001, pp. 1551–1555.

[5] M. Thottan and C. Ji, "Fault prediction at the network layer using intelligent agents," in *IFIP/IEEE Eighth International Symposium on Integrated Network Management*, 2003, pp. 745–759.

[6] R. W. Lucky, "Automatic equalization for digital communication," *Bell Syst. Tech. J.*, vol. 44, no. 4, pp. 547–588, Apr. 1965.

[7] N. Frangiadakis, M. Kyriakakos and L. Merakos, "Enhanced path prediction for network resources management in wireless LANs," *IEEE wireless communications*, pp.62–69, 2003.

[8] San Qi Li and Chia Lin Hwang, "On the convergence of traffic measurement and queuing analysis: a statistical-matching and queuing (SMAQ) tool," *IEEE/ACM Transactions on Networking*, vol. 5, no. 1, 1997, pp. 95–110.

[9] G. R. Faulhaber, "Design of service systems with priority reservation," in *Conf. Rec. IEEE Int. Conf. Communications*, 1995, pp. 3–8.

[10] Aimin Sang and San-Qi Li, "A predictability analysis of network traffic," in *IEEE INFOCOM*, 2000.

[11] W. D. Doyle, "Magnetization reversal in films with biaxial anisotropy," in *Proc. INTERMAG Conf.*, 1987, pp. 2.2-1–2.2-6.

[12] Ziming Zhang and Song Fu, "Failure prediction for automatic Management of networked computer systems with availability assurance," in *15th IEEE Workshop on Dependable Parallel, Distributed and Network-Centric Systems*, 2010.

[13] Y. Liang, Y. Zhang, A. Sivasubramaniam. M. Jette, and R. K. Shahoo, “BlueGene/L failure analisys and prediction models,” in *Proceedings of* international conference on dependable Systems and networks (DSN), 2006.

[14] R. K. Sahoo, A. J. Oliner, and I. Rish, “Critical event prediction for proactive management in large-scale computer clusters,” in *Proceedings of ACM International Conference on Knowledge Discovery and Data Dining (SIGKDD)*, August 2003.

[15] L. Wu, D. Meng, W. Gao, and J. Zhan, “A proactive fault-detection mechanism in large-scale cluster systems,” in *Proceedings of IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2006.

[16] S. Haykin, *Neural Networks for pattern recognition*, Oxford University Press Inc.

[17] N. Frangiadakis, M. Kyriakakos, and L. Merakos, “Enhanced path prediction for network resources management in wireless lans,” *IEEE wireless communications*, pp. 62–69. 2003.

ANEXO

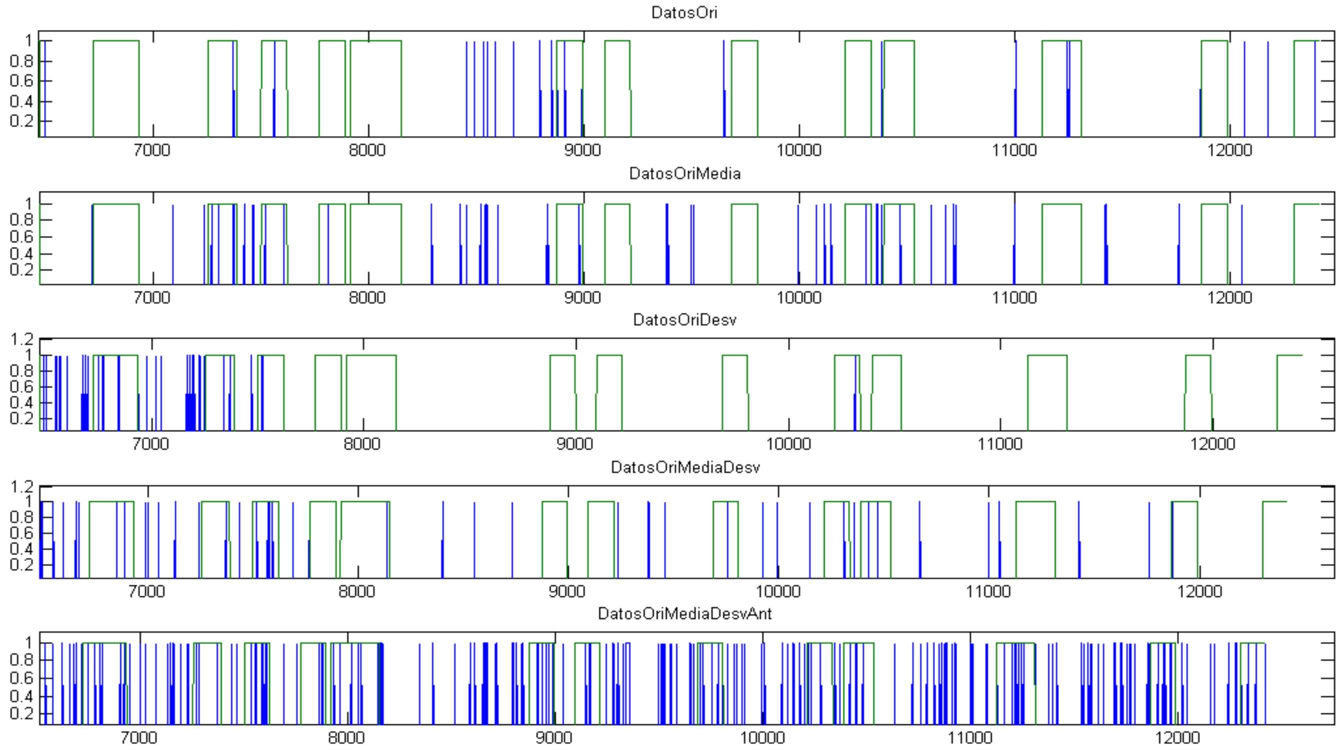


Fig 2. Resultados de pruebas de variables de entrada a la red neuronal.

TABLA V
RESULTADOS DE PRUEBAS DE SELECCIÓN DE NÚMERO DE NEURONAS EN LA CAPA OCULTA.

Prueba No	No. Neuronas	Mse Ent	nErradas	nOk	nOk Prom	(nOk/nErradas)	(nOk/n Erradas) prom	Desv Est nOk
9	2	0,110531	198	117	91,3	0,59	0,47	0,21
3	3	0,112086	133	93	82,8	0,70	0,44	0,21
3	4	0,110273	161	107	91	0,66	0,59	0,16
9	5	0,110105	233	122	125,5	0,52	0,44	0,16
3	6	0,111664	186	133	70,3	0,72	0,49	0,24
1	7	0,109914	227	144	63,2	0,63	0,52	0,27
3	8	0,110727	181	114	68,7	0,63	0,48	0,19
4	9	0,109966	94	69	38,5	0,73	0,32	0,31
2	10	0,109988	180	111	85,6	0,62	0,41	0,23
6	11	0,110726	219	134	96,2	0,61	0,51	0,07
3	12	0,109578	158	105	107,4	0,66	0,45	0,14
9	13	0,112075	233	139	90,4	0,60	0,62	0,16
4	14	0,110857	247	149	95,3	0,60	0,47	0,15
9	15	0,110076	171	103	58,9	0,60	0,37	0,23

TABLA VII
Resultados de Pruebas de Selección del Algoritmo de Entrenamiento

Prueba	Algoritmo	MseEntMin	nErradas	nOk	nOk / nErradas	Prom Ind	Prom Nok	DesvEst Ind
2	traingd	0,1103	315,00	162,00	0,51	0,37	56,60	0,15
1	traingdm	0,1107	204,00	124,00	0,61	0,54	100,20	0,16
4	traingda	0,1074	306,00	161,00	0,53	0,44	86,80	0,05
5	trainrp	0,1035	389,00	241,00	0,62	0,54	234,90	0,06

TABLA IX
Resultados de Pruebas de Momentum del Algoritmo de Aprendizaje.

PruebaNo	mc	MseEntMin	nErradas	nOk	(nOk/nErradas)	Promedio Ind	Desv Est	Error Promedio
10	0,01	0,110705	204	124	0,61	0,40	0,19	4325,3
8	0,02	0,110027	305	149	0,49	0,51	0,17	4368,7
8	0,03	0,110327	109	78	0,72	0,55	0,32	4292,6
2	0,04	0,110501	196	122	0,62	0,50	0,20	4345,1
5	0,05	0,111212	303	159	0,52	0,55	0,19	4386,6
5	0,06	0,109703	236	124	0,53	0,41	0,19	4347,1
6	0,07	0,111071	201	127	0,63	0,40	0,23	4345,2
3	0,08	0,110303	227	133	0,59	0,47	0,21	4332,4
4	0,09	0,109769	235	140	0,60	0,50	0,23	4335,7
7	0,10	0,110687	159	108	0,68	0,45	0,19	4368
8	0,11	0,110848	236	111	0,47	0,39	0,22	4334,9
4	0,12	0,109861	283	142	0,50	0,44	0,17	4333,4
9	0,13	0,110393	189	120	0,63	0,49	0,23	4332,9
4	0,14	0,110068	168	108	0,64	0,45	0,18	4363,4
7	0,15	0,109735	261	137	0,52	0,41	0,17	4384,2

TABLA 11
Resultados de Pruebas de la Tasa de Aprendizaje del Algoritmo de Aprendizaje.

Prueba	lr	MseEntMin	nErradas	nOk	(nOk / nErradas)	Promedio Ind	Desv Est	Promedio indicador	Error Promedio	NErr Promedio
5	0,01	0,110231	2	2	1,00	0,41	0,26	90,7	4386,8	206,5
7	0,02	0,109600	38	30	0,79	0,51	0,19	95,3	4368,9	193,2
10	0,03	0,109871	5	5	1,00	0,54	0,25	72,6	4335,9	137,5
4	0,04	0,109890	106	70	0,66	0,51	0,20	74,2	4337,1	140,3
7	0,05	0,110586	1	1	1,00	0,51	0,26	79,4	4356,2	164,6
4	0,06	0,110092	71	56	0,79	0,49	0,21	105,3	4373,2	207,5
10	0,07	0,110963	201	127	0,63	0,35	0,25	53,6	4321,6	104,2
2	0,08	0,110089	218	133	0,61	0,48	0,11	85,7	4348,3	163
4	0,09	0,109995	3	3	1,00	0,51	0,26	62,7	4322,5	114,2
7	0,10	0,109770	79	58	0,73	0,38	0,25	42,6	4319,5	91,1
2	0,11	0,110474	100	74	0,74	0,44	0,27	78	4340,8	147,8
7	0,12	0,109633	194	116	0,60	0,48	0,12	61,9	4333,9	124,8
8	0,13	0,109807	22	22	1,00	0,40	0,32	29,9	4297,3	56,2
4	0,14	0,110667	34	28	0,82	0,49	0,24	60,9	4333	122,9
1	0,15	0,109805	82	55	0,67	0,41	0,21	65,6	4338,8	133,4

TABLA XIII
RESULTADOS DE PRUEBAS DE LA SELECCIÓN DEL UMBRAL.

Prueba No	Umbral	MseEntMin	nErradas	nOk	(nOk/nErradas)	Promedio Ind	Desv Est	Promedio indicador	Error Promedio	NErr Promedio
5	1,9	0,110100	5	4	0,80	0,39	0,23	90,9	4378,5	198,4
1	2	0,109845	382	198	0,52	0,38	0,17	78,3	4356,4	163,7
4	2,1	0,110363	249	134	0,54	0,35	0,20	66,6	4339,9	135,5
1	2,2	0,109845	298	161	0,54	0,31	0,24	57,5	4329,4	115,9
1	2,3	0,109845	266	148	0,56	0,33	0,24	50,3	4320	99,3
1	2,4	0,109845	228	128	0,56	0,38	0,32	42,1	4313	84,1
4	2,5	0,110363	131	79	0,60	0,33	0,24	36,8	4306,5	72,3

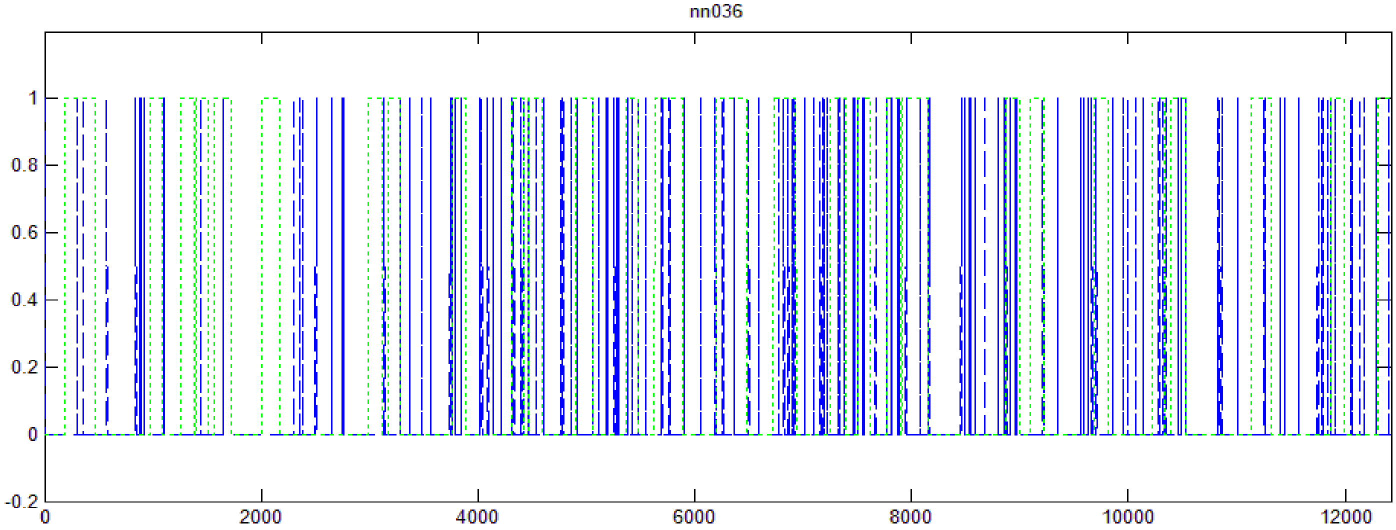


Fig. 3. Gráfica de resultados del sistema de predicción.

Journal Information and Instructions for Authors

I. Journal Information

"*Polibits*" is a half-yearly research journal published since 1989 by the Center for Technological Design and Development in Computer Science (CIDETEC) of the National Polytechnic (Technical) Institute (IPN) in Mexico City, Mexico. The journal solicits original research papers in all areas of computer science and computer engineering, with emphasis on applied research.

The journal has double-blind review procedure. It publishes papers in English and Spanish.

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.

More specifically, the main topics of interest include, though 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, Virtual Reality
- Machine Learning
- Pattern Recognition
- Intelligent Tutoring Systems
- Semantic Web
- 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
- Geo-processing
- Networks and Connectivity
- Cryptography
- Informatics Security
- Digital Systems Design
- Digital Signal Processing
- Control Systems
- Robotics
- Virtual Instrumentation
- Computer Architecture
- other.

B. Indexing

LatIndex, Periódica, e-revistas.

II. Instructions for Authors

A. Submission

Papers ready to review are received through the Web submission system www.easychair.org/polibits. See also the updated information at the web page of the journal www.cidetec.ipn.mx/polibits.

The papers can be written in English or Spanish.

Since the review procedure is double-blind, the full text of the papers should be submitted without names and affiliations of the authors and without any other data that reveals the authors' identity.

For review, a file in one of the following formats is to be submitted: PDF (preferred), PS, Word. In case of acceptance, you will need to upload your source file in Word or TeX. We will send you further instructions on uploading your camera-ready source files upon acceptance notification.

Deadline for the nearest issue (July-December 2010): September 1, 2010. Papers received after this date will be considered for the next issues.

B. Format

Please, use IEEE format[1], see section "Template for all Transactions (except IEEE Transactions on Magnetics)". The editors keep the right to modify the format and style of the final version of the paper if necessary.

We do not have any specific page limit: we welcome both short and long papers, provided the quality and novelty of the paper adequately justifies the length.

In case of being written in Spanish, the paper should also contain the title, abstract, and keywords in English.

[1] www.ieee.org/web/publications/authors/transjnl/index.html