Advances in Computational Linguistics and Intelligent Decision Making

Research in Computing Science

Research in Computing Science es una publicación trimestral, de circulación internacional, editada por el Centro de Investigación en Computación del IPN, para dar a conocer los avances de investigación científica y desarrollo tecnológico de la comunidad científica internacional. **Volumen 86,** noviembre de 2014. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No. : 04-2005-121611550100-102, expedido por el Instituto Nacional de Derecho de Autor. *Certificado de Licitud de Título* No. 12897, *Certificado de licitud de Contenido* No. 10470, expedidos por la Comisión Calificadora de Publicaciones y Revistas Ilustradas. El contenido de los artículos es responsabilidad exclusiva de sus respectivos autores. Queda prohibida la reproducción total o parcial, por cualquier medio, sin el permiso expreso del editor, excepto para uso personal o de estudio haciendo cita explícita en la primera página de cada documento. Impreso en la Ciudad de México, en los Talleres Gráficos del IPN – Dirección de Publicaciones, Tres Guerras 27, Centro Histórico, México, D.F. Distribuida por el Centro de Investigación en Computación, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, México, D.F. Tel. 57 29 60 00, ext. 56571.

Editor responsable: *Grigori Sidorov, RFC SIGR651028L69*

Research in Computing Science is published by the Center for Computing Research of IPN. **Volume 86**, November 2014. Printing 500. The authors are responsible for the contents of their articles. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior permission of Centre for Computing Research. Printed in Mexico City, in the IPN Graphic Workshop – Publication Office.

Advances in Computational Linguistics and Intelligent Decision Making

Sofía Galicia-Haro (ed.)

SEP
SECRETARÍA DE
EDUCACIÓN PÚBLICA

Instituto Politécnico Nacional
"La Técnica al Servicio de la Patria"

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2014

ISSN: 1870-4069

Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal
Unidad Profesional "Adolfo López Mateos", Zacatenco
07738, México D.F., México

http://www.rcs.cic.ipn.mx
http://www.ipn.mx
http://www.cic.ipn.mx

The editors and the publisher of this journal have made their best effort in preparing this special issue, but make no warranty of any kind, expressed or implied, with regard to the information contained in this volume.

Indexed in LATINDEX and Periodica / Indexada en LATINDEX y Periódica

Printing: 500 / Tiraje: 500

Printed in Mexico / Impreso en México

Editorial

This volume of the journal "Research in Computing Science" contains selected papers on the modern interdisciplinary research areas: computational linguistics (when the computer model the linguistic abilities of humans) and reasoning (when computers model another human activity: thinking processes).

The papers were carefully chosen by the editorial board on the basis of the at least two reviews by the members of the reviewing committee or additional reviewers. The reviewers took into account the originality, scientific contribution to the field, soundness and technical quality of the papers. It is worth noting that various papers for this special issue were rejected.

As far as computational linguistics is concerned, this volume contains papers on topics like automatic font identification, HPSG grammar for syntactic coordination in Arabic, textual entailment in Bengali, advertising in social networks, and speech synthesis in Mexican Spanish. The part of the reasoning contains papers on decision support system, agents, and route recommendation for a robot based on pheromons.

I would like to thank Mexican Society for Artificial Intelligence (Sociedad Mexicana de Inteligencia Artificial), MICAI 2014 conference, Instituto Tecnológico de Tuxtla Gutierrez (Chiapas, Mexico), and Universidad Autónoma de Chiapas for their support during preparation of this volume.

The papers were collected and the reviewing process was organized using the system EasyChair.

Sofía Galicia Haro
November 2014

Table of Contents

Page

Automatic Bilingual Legacy-Fonts Identification and Conversion System

Gurpreet Singh Lehal[1], Tejinder Singh[2], and Saini Pretpal Kaur Buttar[1]

[1] DCS, Punjabi University, Patiala,
India

[2] ACTDPL, Punjabi University, Patiala,
India

{gslehal, preetpalkaur15}@gmail.com, tej@pbi.ac.in

Abstract. The digital text written in an Indian script is difficult to use as such. This is because, there are a number of font formats available for typing, and these font-formats are not mutually compatible. Gurmukhi alone has more than 225 popular ASCII-based fonts whereas this figure is 180 in case of Devanagari. To read the text written in a particular font, that font is required to be installed on that system. This paper describes a language and font-detection system for Gurmukhi and Devanagari. It also explains a font conversion system for converting the ASCII based text into Unicode. Therefore, the proposed system works in two stages: the first stage suggests a statistical model for automatic language-detection (i.e., Gurmukhi or Devanagari) and font-detection; the second stage converts the detected text into Unicode as per font detection. Though we could not train our systems for some fonts due to non-availability of font converters but system and its architecture is open to accept any number of languages/fonts in the future. The existing system supports around 150 popular Gurmukhi font encodings and more than 100 popular Devanagari fonts. We have demonstrated the effectiveness of font detection is 99.6% and Unicode conversion is 100% in all the cases.

Keywords: n-gram language model, Gurmukhi, Devanagari, Punjabi, Hindi, fonts, font detection, font conversion, Unicode.

1 Introduction

The text on the Internet is available in numerous languages and encodings. These encodings are often not based on any standards. In any NLP application for Indian or any other language, the input text can be processed only after knowing its language and the underlying font-encoding. In many cases this language-encoding is not known in advance and has to be determined. This problem can be viewed as font as well as language identification problem. For languages like Punjabi, Hindi and others, there is no standard font encoding followed by everyone. A large amount of digital text written in Indian languages is in ASCII-based font-formats. It has been found there are many fonts which belong to the same keyboard-mapping, which can be grouped together. Gurmukhi alone has more than 225 popular ASCII-based fonts with 41 keyboard-mappings. In Devanagari, there are more than 180 popular Devanagari ASCII-based font-formats and with 52 different keyboard-mappings. According to Raj and Prahallad[3] the problem of font-identification could be defined as: given a set of words or sentences, identify the font-encoding by finding the minimum distance between the input glyph codes and the models representing font-encodings.

To solve this problem of mutual incompatibility among various ASCII-based font-formats, Unicode was developed as a standard that would assign a unique number known as a code point to every letter in every language. But still the popularity of Unicode is not fully accepted. There are some reasons for this:

- Lack of awareness
- Typing issues in Unicode standards
- Non availability of Unicode typing tools for Indian language
- Media Printing/Publishing houses do not have full support for Unicode
- Lack of variety in Unicode fonts for Indian Languages

In order to find a solution of this problem, we have developed an automatic system to bridge the gap between legacy-fonts and Unicode. Currently, it is working on Gurmukhi and Devanagari languages, but the system and its architecture is open to accept any number of languages/fonts in the future. The proposed system works in two stages: the first stage suggests a statistical n-gram model for automatic font-detection as well as language-detection (i.e., Gurmukhi or Devanagari); the output of this stage is ranked weighted list of trained fonts, from where the topmost font, which has the maximum weight, is selected by the system as the detected font. The second stage converts the text into Unicode as per detected font.

2 Related Work

The problem of font and language identification is addressed by many researchers in the literature [1-5]. The earliest approaches used for automatic language identification were based on unique strings. The proposed mathematical language models relied on orthographic features like characteristic letter sequences and frequencies for each language.

Singh and Gorla [5] presented their work on identifying the languages and encodings of a multilingual document. It involved the steps of monolingual identification, enumeration of languages and then identification of the language of every portion. For enumeration, they have been able to get a precision of 96.20%. Raj and Prahallad [3] have discussed the Term Frequency - Inverse Document Frequency (TF-IDF) weights based approach for font identification. They have modeled a vector space model and TF-IDF weights for each term in the font-data according to its uniqueness. In experiments, they have demonstrated the effectiveness of font data conversion to be as high as 99% on 10 Indian languages and for 37 different font-types. As described earlier, this font size is very small as compared to our research problem.

Font identification in monolingual, bilingual or multilingual system can be seen as a classification task. Like other NLP tasks, we can think of using some sophisticated pattern classification technique such as maximum entropy, for solving this task. But maximum entropy would require training and testing data which is not easy to prepare in our case. As described by Lehal et al. [2], [5] we found that a simpler n-gram model based similarity method is more suitable for this purpose. The advantage of this method is that only a small amount of training data per font-encoding is enough and yields excellent results without using any specially selected features.

3 Problem Complexity

The lack of a standard poses difficulties in processing text written in some non-standard font-formats. The major issues we found are:

- There is no uniform mapping of a character to a code value in Indian languages. Most Indian language fonts assign different codes to same character. For example, consider the word ਪੰਜਾਬੀ, it is internally stored at different keys in different fonts as shown in Table 1.

Table 1. Internal representation of word ਪੰਜਾਬੀ /punjabi/ in different Gurmukhi fonts.

Sr.	Font	ASCII Character Code
1	Akhar	112, 077, 106, 119, 098, 073
2	Gold	102, 046, 117, 106, 087, 103
3	AnandpurSahib	112, 181, 106, 059, 098, 073
4	Asees	103, 122, 105, 107, 112, 104
5	Sukhmani	080, 094, 074, 065, 066, 073
6	Satluj	234, 179, 220, 197, 236, 198

- There is no standard which defines the number of glyphs per character or word in that language, and hence it differs between fonts of a specific language itself. The glyphs are shapes, and when 2 or more glyphs are combined together, they form a character in the scripts of Indian languages. The underlying codes for the individual characters are according to the glyphs they are broken into. This problem is more prevalent in Devanagari as compared to Gurmukhi fonts. The decomposition of glyphs and the codes assigned to them are both different in Devanagari. Table 2 shows how the same word is internally coded in two separate Devanagari fonts, viz., Chanakya and DV-TT-Surekh. In Chanakya, it was five character codes whereas it used seven character codes for same words in DV-TT-Surekh.

Table 2. Code mapping for Devanagari word स्थिति into two different fonts.

Font	Chanakya	DV-TT-Surekh
Char Code	231 83 205 231 204	202 186 108 201 202 105 201

- The third problem is that there is no standard procedure to align the characters while rendering. For example, consider the Gurmukhi word ਠੂੰ. The order of rendering the glyphs can be: first pivotal character, then bottom character, and then top character; or the top character can be rendered before the bottom character. However, Unicode allows only specific sequence of rendering and hence, some rendering may not be supported in Unicode text and needs to be handled in the final output as shown in figure 1.

n + ¨ + = n¨µ
µ

n + µ + = n¨µ (Not in Unicode)
¨

Fig. 1. Different orders of rendering in Gurmukhi fonts.

4 The Proposed System

The proposed system supports around 150 popular Gurmukhi font-encodings and around 100 Devanagari font-encodings. Some of the popular fonts supported are *Akhar, Anmol Lipi, Chatrik, Joy, Punjabi, Satluj, Chanakya, Agra, DV-TT-Yogesh, KrutiDev, Shusha*, etc. A detailed analysis of font encodings showed that many fonts belonging to same keyboard-map have same internal mappings for all the characters. For example, *Akhar* and *Akhar2010* font belong to same keyboard-map family. In fact, we analyzed that all these fonts correspond to 81 unique keyboard-mappings.

The Gurmukhi fonts correspond to 41 keyboard-mappings and Devanagari fonts correspond to 40 keyboard-mappings. It means, if k_0, k_1 ..... k_{80} be the 81 keyboard mappings and f_0, f_1 ...... f_{250} be the fonts, then each of fonts f_i will belong to one of the keyboard mapping k_i. The problem is thus reduced from 250 distinct fonts to just 81 group classes corresponding to each keyboard map. Therefore, our font detection problem is to classify the input text to one of these 81 keyboard mappings. It could be thought of as an 81 class pattern recognition problem. The system for font-detection is based on character-level trigram language model (figure 2).

A raw corpus of around 66,000 words for Gurmukhi and 50,000 words for Devanagari has been used for training the system. Corresponding to each keyboard-map, trigrams have been trained. Font identification is done by extracting the character-level trigrams from the input text and then a score is calculated for each keyboard-mapping which indicates the probability of the keyboard-mapping to be the font of the input text. The keyboard-map having maximum score is identified as the font of the input text and conversion to Unicode text is then performed.

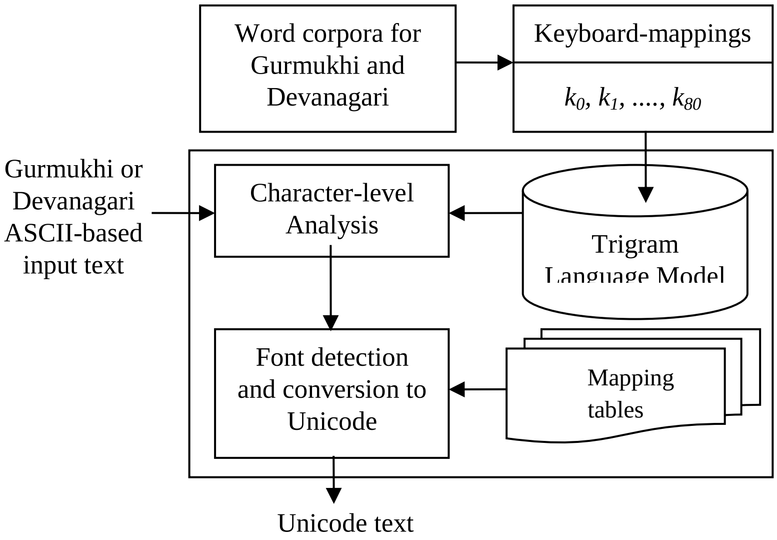

Word corpora for Gurmukhi and Devanagari
Keyboard-mappings
k0, k1, ..., k80
Gurmukhi or Devanagari ASCII-based input text
Character-level Analysis
Trigram Language Model
Font detection and conversion to Unicode
Mapping tables
Unicode text

Fig. 2. Framework for font detection and conversion.

4.1 The Proposed Data Structure

Language and encoding are closely interconnected in such a way that if we could identify the font-encoding, we would most probably have also identified the language. The font detection problem is formulated as character level trigram language model. The single font has 256 character code points. Therefore, in a trigram model, 256x256x256=256^3 code points need to be processed for a single keyboard map. But in order to deal with 81 distinct keyboard maps, the memory requirements will further increase to process and hold 81x 256^3 code points. After detailed analysis, it has been found that the array representation of this task is sparse in nature i.e. the majority of code points in omni fonts have zero values. It has been observed that each keyboard map has around 26,000 non-zero code points which is 0.156% of the original code points. Hence, to avoid sparse array formation, binary search tree representation has been created. Each node of the tree contains single unique trigram that exists in one or more of the keyboard-mappings' training data. Every tree node has a linked list associated with it. Each node of this list contains the keyboard-mapping number (k_0, k_1, ..., k_{80}) of the training data in which the corresponding trigram exists and the probability of the trigram in the training data of that keyboard-mapping. The structures of the nodes are expressed in figure 3.

The total number of nodes in the tree are 4,41,648. The height of the tree is 80. The shortest list linked to a node in the tree has length 1, i.e., the shortest linked list contains 1 node. The longest node contains 76 nodes. The average number of nodes in the linked lists is found to be 4.

The major advantage of using this representation is performance enhancement. As we know, searching in a binary search tree is faster as compared to that in a linked list. The structure is dynamic, as the lists linked to the nodes of the trees have their sizes dependent on the number of fonts which have non-zero probability of the trigram stored at the node. In other words, length of the list of fonts is not fixed to be equal to the total number of fonts. This is because most fonts can have non-zero probability for that trigram at the node. Moreover, by combining the list of those fonts in which the trigram stored at the tree-node exists, we access the probabilities of that trigram in all the fonts right there. Additionally, this architecture is open to add more language/fonts without affecting the performance.

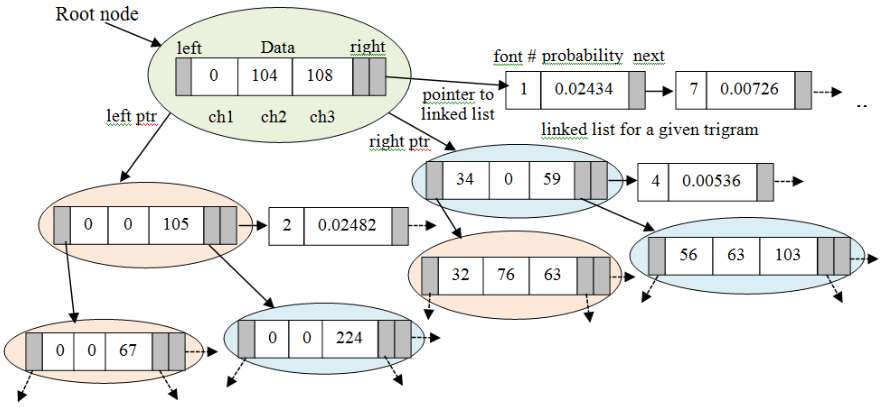


Fig. 3. Binary search tree node structure.

Adding Tree Node: To store trigrams and their probability in the data structure, the tree is searched for each trigram one by one. Let us suppose, the tree is being searched for the trigram (0, 104, and 108). If the trigram is found in the tree, we access the linked list of that trigram and add a new node containing the font number and the probability of that trigram in that font. If the trigram is not found, a new node of the tree is created containing that trigram and a linked list containing the font number and probability of the trigram.

Searching the Binary Tree: Suppose, we have a word ਕਲਮ, whose font is to be detected. First, the system will break down this word into trigrams. Then, the system will search for each trigram in the binary search tree one by one. Thus, first it searches for the trigram (0, 0, and 107). On finding the trigram, it goes to the list linked with that trigram's node. There it finds the probabilities of that trigram in all the fonts from 0 to 80. For example, the trigram (0, 0, and 107) has non-zero probabilities in the font-numbers 1, 2, 11, and 23 which are 0.2434, 0.435, 0.04364, and 0.06433 respectively. For all the other fonts, probability of this trigram is considered to be zero. Similarly, the system searches all other trigrams and collects their probabilities.

5 Font Detection

The task of font detection can be classified as an n-gram based text similarity method. The advantage of this method is that only a small amount of training data per font encoding is enough to get the desired results. As discussed earlier we have created a raw corpus of around 50000 words for training the system. The proposed methods of font detections are:

- Word level prediction
- Character level prediction
- Hybrid Approach using Unseen probability

The following two methods have been used to calculate the score of a keyboard-mapping for the prediction of font of the input text.

5.1 Word Level Prediction

The trigram probability of a word w^l of length l is calculated as a product of trigram probability for all possible combinations as shown in equation 1. Clearly, if any trigram has zero probability then the probability of the word becomes zero. The total probability of the input text of n words, for defined keyboard-mapping k is calculated as sum of all words probabilities as shown in equation 2 and the best prediction for keyboard-mapping index is detected corresponding to the maximum weight of probability as shown in equation 3.

$$w^l = \prod_{i=1}^{l} P(c_i \mid c_{i-1}, c_{i-2}) \qquad (1)$$

$$W_k = w_{1,n} = \sum_{1}^{n} w^l, \qquad (2)$$

$$K = argmax\ W_k \qquad (3)$$

For example, consider a sentence of four words ਪਾਣੀ ਵਿਅਰਥ ਨਾ ਵਹਾਓ. In the word level prediction, the probability of the word ਪਾਣੀ can be calculated as:

$$w^4 = \prod_{i=1}^{4} P(c_i \mid c_{i-1}, c_{i-2}) = 0.009001276 * 0.000873424 * 0.000085518$$
$$* 0.000198584 * 0.000591988 * 0.023062862 = 1.822875e\text{-}18$$

Similarly, the total probabilities of remaining three words are calculated and the overall probability of the whole sentence is:

$$W_1 = w_{1,4} = 1.822875e\text{-}18 + 8.063544e\text{-}28 + 2.290603e\text{-}10 + 4.744855e\text{-}28$$
$$= 2.290603e\text{-}10$$

In this way, the probability of all input words is calculated for all trained fonts and the detected keyboard-mapping index pops up corresponding to the maximum weight of probability.

5.2 Character Level Prediction

Unlike word level prediction, this method considers all trigram probabilities at character level not at word level. In other words, the total probability of the input text for keyboard-mapping k is calculated as sum of all valid trigrams of the input text as shown in equation 4 and the detected font index is corresponding to the maximum weight of probability as $K = argmax\ C_k$.

$$C_k = \sum_{i=1}^{n} P(c_i \mid c_{i-1}, c_{i-2}) , \quad (4)$$

Using the same example as given above, there are 23 trigrams corresponding to sentence of four words and the probability of the text comes out to be:

$$C_1 = \sum_{i=1}^{23} P(c_i \mid c_{i-1}, c_{i-2}) = 0.102334869$$

5.3 Hybrid Approach using Unseen probability

Unseen Probability factor (*B*$_k$). The above two methods detect the probability of the input text in the training data. On the other hand, we can think of another way out to calculate the probability, i.e., how much of the input text does not belong to the training data. This can be seen as a probability of unseen trigrams. We can quantify this by using a variable 'B_{ck}' for each trained font, initialized to 0, which is incremented by one for an unseen trigram, as shown in equation 5. Then the unseen probability can be calculated with the equation 6.

$$B_{ck} = \sum_{i:P\ (c_i \mid c_{i-1}, c_{i-2})=0} 1 \quad (5)$$

$$B_k = 1 - [B_{ck} / \text{(Total number of trigrams)}] \quad (6)$$

Now, the two proposed methods have been configured to incorporate unseen probability factor, so that each contributes towards the final selection. Now, the overall probability of the word- and character-level methods will be combined as shown in equation 7 and 8 respectively. The value of α is used to determine each factor's contribution towards the final outcome.

$$K = argmax\ (\alpha * W_k + (1 - \alpha) * B_k), \text{ where } 0 \leq \alpha \leq 1 \quad (7)$$

$$K = argmax\ (\alpha * C_k + (1 - \alpha) * B_k), \text{ where } 0 \leq \alpha \leq 1 \quad (8)$$

It has been concluded from the test results that the hybrid methods gave more accurate results when the value of α was kept equal to or slightly less than 0.3. That is, the unseen-trigram factor was contributing more towards the oveall probability.

6 Language Detection

The total number of keyboard-mappings supported by the system is 81. The keyboard mappings ranging 0 to 40 correspond to Gurmukhi fonts and the rest 41 to 80 correspond to Devanagari. Thus, the language of the input text is Punjabi, if the detected keyboard mapping is lying anywhere in the range 0 to 40, and Hindi if otherwise.

7 Conversion to Unicode

Unicode conversion is performed by mapping all fonts to a single intermediate form with the help of mapping tables generated for each font. Then, the rule based approach is followed for conversion of text from intermediate form to Unicode. Many rules have been formulated for proper rendering of text in Unicode format. The steps for font-data conversion are explained as follows:

7.1 Mapping Tables and Intermediate Form

For performing the conversion of input text into Unicode text on the basis of detected font encoding, first, a mapping table of the detected font encoding is used. The mapping tables are static alignments between all font glyphs and an intermediate form. The intermediate form is a list of all the glyphs which exist in a particular language. These glyphs are assigned an internal code. A codepoint corresponding to a gylph in a font is mapped to an internal code corresponding to the same glyph in the intermediate form. In this manner, the mapping tables have been built for all the trained fonts. The advantage of using intermediate form is that it reduces the complication, as we have to create less number of rules because similar glyphs are mapped onto single internal code in the intermediate form.

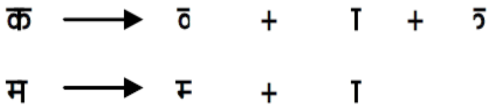
For example, consider the *Chanakya* font's glyph % and another glyph combination ^Ù which means the same, but there is minor difference in shape. But Unicode transformation of both the glyphs is same. Hence, these types of glyphs are treated as one in our intermediate and Unicode transformations. Also, it is easy to add a new font to the system as we need only a training file and a mapping table of that font. There is no need to create new transformations rules for the newly added font as transformation to Unicode is not done directly from font to Unicode, but from intermediate form to Unicode. We only need to build a trigram-probability training file and a mapping table for the new font.

The input text written in the detected font is converted into the intermediate form using the mapping table for that font.

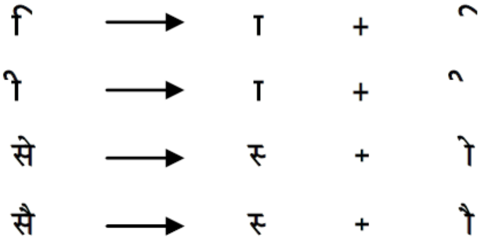
7.2 Intermediate Form to Unicode Conversion

The next step is the conversion of intermediate form into Unicode text. The conversion to Unicode is not straight forward due to complex Unicode transformation rules. We have crafted various conversion rules to perform this transformation.

Handling full characters in Devanagari fonts. In some Devanagari fonts, there are no separate glyphs for full-characters. They are formed by combining other glyphs together. For example, in *APS-C-DV̲-Prakash* font,



Similarly, other glyph combinations are formed, such as:

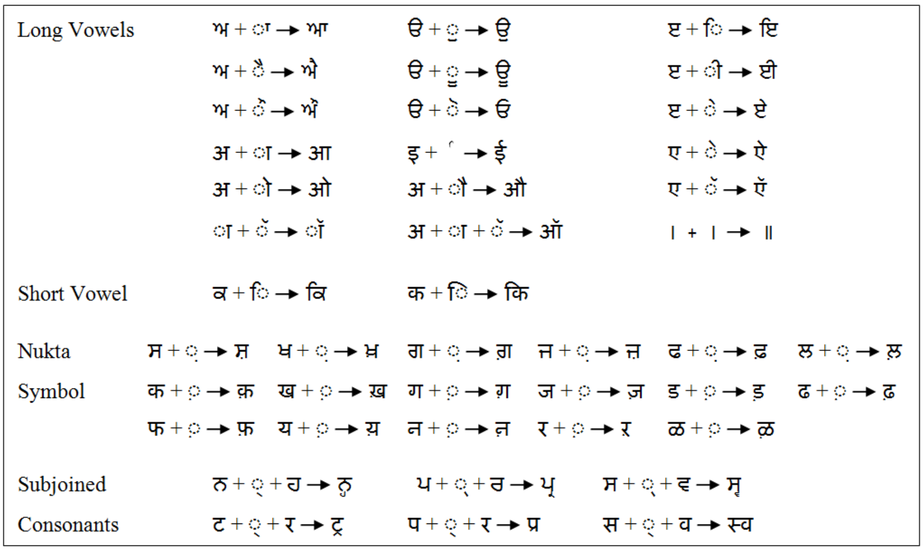


To handle such combinations, rules have been formulated, for example, we know that a half-character cannot be combined with a *matra*, it exists only in conjunction with a full-character. Thus, if there exists a half-character in combination with a *matra* i.e., ◌ा, ◌ो, or ◌ौ, then it should be converted into the corresponding full-character applying corresponding *matra*, if any, such as none, ◌े, ◌ै respectively.

Transforming Long Vowels. Next, rules have been devised for transforming Gurmukhi long vowels ਉ[ʊ], ਊ[u], ਓ[o], ਆ[ɑ], ਇ[ɪ], ਈ[i], ਏ[e], ਐ[æ], ਔ[ɔ]. This is because, in Unicode these nine independent vowels with three bearer characters Ura ੳ[ʊ], Aira ਅ [ə] and Iri ੲ[ɪ] have single code points and need to be mapped accordingly as shown in Figure 5. Similarly, Devanagari independent long vowels आ[ɑ], ई[i], ऊ[u], ऍ[eɪ], ऐ[æ], ऑ[ɒ], ओ[o], औ[ɔ] with four bearer characters अ[ə], इ[ɪ], उ[ʊ] and ए[e], and short vowel ◌ॉ[ɒ] were mapped accordingly.

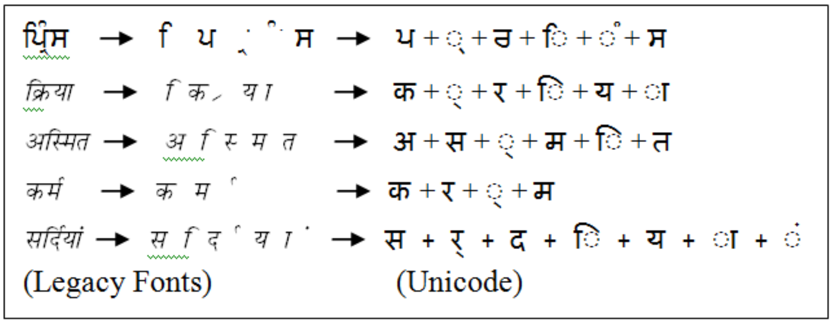
Rendering half-character in Unicode. There are no explicit code points for half-characters in Unicode. They will be generated automatically by the Unicode rendering system when it finds special symbol called *halant* or *virama* [◌्]. Therefore, Gurmukhi/Devanagari transformation of subjoined consonants is performed by prefixing *halant* ◌् symbol along with the respective consonant, as shown in Figure 4.

Handling of short vowel ਿ◌. Mapping of short vowel sign ਿ◌[ɪ] has been done according to Unicode rendering system. The Unicode transformation becomes complex when short vowel ਿ◌[ɪ] and subjoined consonants come together at a single word position.



Long Vowels		ਅ + ਾ → ਆ	ੳ + ੁ → ਉ	ੲ + ਿ → ਇ		
		ਅ + ੈ → ਐ	ੳ + ੂ → ਊ	ੲ + ੀ → ਈ		
		ਅ + ੌ → ਔ	ੳ + ੋ → ਓ	ੲ + ੇ → ਏ		
		अ + ा → आ	इ + ॅ → ई	ए + े → ऐ		
		अ + ो → ओ	अ + ौ → औ	ए + ॅ → ऍ		
		ा + ॅ → ॉ	अ + ा + ॅ → ऑ	। + । → ॥		
Short Vowel		ਕ + ਿ → ਕਿ	क + ि → कि			
Nukta	ਸ + ਼ → ਸ਼	ਖ + ਼ → ਖ਼	ਗ + ਼ → ਗ਼	ਜ + ਼ → ਜ਼	ਫ + ਼ → ਫ਼	ਲ + ਼ → ਲ਼
Symbol	क + ़ → क़	ख + ़ → ख़	ग + ़ → ग़	ज + ़ → ज़	ड + ़ → ड़	ढ + ़ → ढ़
	फ + ़ → फ़	य + ़ → य़	न + ़ → ऩ	र + ़ → ऱ	ळ + ़ → ऴ	
Subjoined		ਨ + ੍ + ਹ → ਨ੍ਹ	ਪ + ੍ + ਰ → ਪ੍ਰ	ਸ + ੍ + ਵ → ਸ੍ਵ		
Consonants		ट + ् + र → ट्र	प + ् + र → प्र	स + ् + व → स्व		

Fig. 4. Unicode transforming rules.



ਪ੍ਰਿੰਸ	→	ਿ ਪ ੍ਰ ੰ ਸ	→	ਪ + ੍ + ਰ + ਿ + ੰ + ਸ
क्रिया	→	ि क ्र या	→	क + ् + र + ि + य + ा
अस्मित	→	अ ि स् म त	→	अ + स + ् + म + ि + त
कर्म	→	क म ्	→	क + र + ् + म
सर्दियां	→	स ि द ् य ा ं	→	स + र् + द + ि + य + ा + ं
(Legacy Fonts)				(Unicode)

Fig. 5. Complex Unicode transformations.

For example, consider the Gurmukhi word ਪ੍ਰਿੰਸ .In omni fonts, [ਿ] appears as the first character. But according to Unicode rendering, it must be after the bearing consonant. Therefore, it must go after the subjoined consonant ੍ + ਰ, as shown in Figure 5.

Similarly, when half-character comes in between [ि] and a full-character, then according to Unicode rendering system, half-character must be written first, then the full-character and [ि] at the last position as shown in the word अस्मित in Figure 5.

Other Issues. Some Devanagari fonts consist of glyphs depicting some combinations of matras, such as ਿੰ ਿੱ ਿ. . Consider the word शर्मिंदा this word uses matra ਿੰ . In order to handle this word as per Unicode rendering, the following rule-based transformations are done:

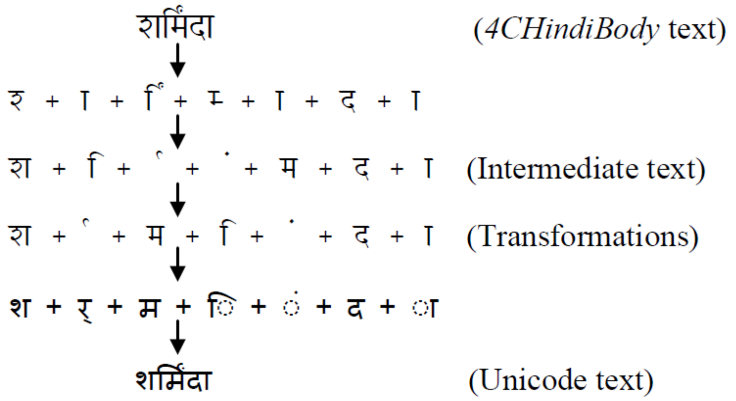


Fig. 6. शर्मिंदा Word transformations while converting to Unicode.

In Unicode, there are no explicit glyphs for rendering complex consonants, such as क्ष, श्र etc. These are generated by using combinations of consonants with halant (्) as shown in Figure 7.

क्ष = क + ् + ष   त्र = त + ् + र

ज्ञ = ज + ् + ञ   श्र = श + ् + र

Fig. 7. Rendering of complex consonants in Unicode.

8 Evaluation and Results

8.1 Test Data Preparation

For evaluation purpose, we collected random text written in Gurmukhi and Devanagari from 27 different sources, for example, AmarUjala, BBC Hindi News, Punjabi Tribune, etc. The data was categorized under five sections namely, articles, books, news, poems and stories. For each section, 4 sets of about 1000 words each were prepared for each trained font. The fonts for which real ASCII-based data was not available, font-converters were used to convert Unicode text into font-data.

8.2 Font Detection Results

All the methods are then tested over the input-text and the results of font detection among proposed methods are shown in Table 3. Clearly, character-level prediction method has shown better results in both Gurmukhi as well as Devanagari font detection as compared to word-level prediction.

Table 3. Font detection accuracy of different methods.

Data Set Domain (200 words)	**Word-level**		**Char-level**		**Hybrid Word-level (α=0.3)**		**Hybrid Char-level (α=0.3)**	
	Gur	**Dev**	**Gur**	**Dev**	**Gur**	**Dev**	**Gur**	**Dev**
Articles	51	75	95	90	100	98	100	100
Books	51	68	83	93	98	100	100	100
News	76	78	66	90	100	100	100	100
Poems	51	73	73	90	98	98	98	100
Stories	78	50	76	90	98	100	100	98

The hybrid approach has shown the overall improvement in font detection as compared to single method approach. Unseen probability factor has shown better results. The optimal value of system combination factor α is found to be 0.3. Again, it has been found that hybrid character-level method is the best among all the other methods we have discussed. The average detection results of this method are 99.6% for both Gurmukhi and Devanagari fonts. It is also come out from the results that the proposed n-gram approach with limited training data has successfully overcome the confusion amongst Gurmukhi and Devanagari fonts. It has been found that for the input text of any size, we need at most first 200 words for the task of font-detection. This factor contributes towards the improvement of performance of the system.

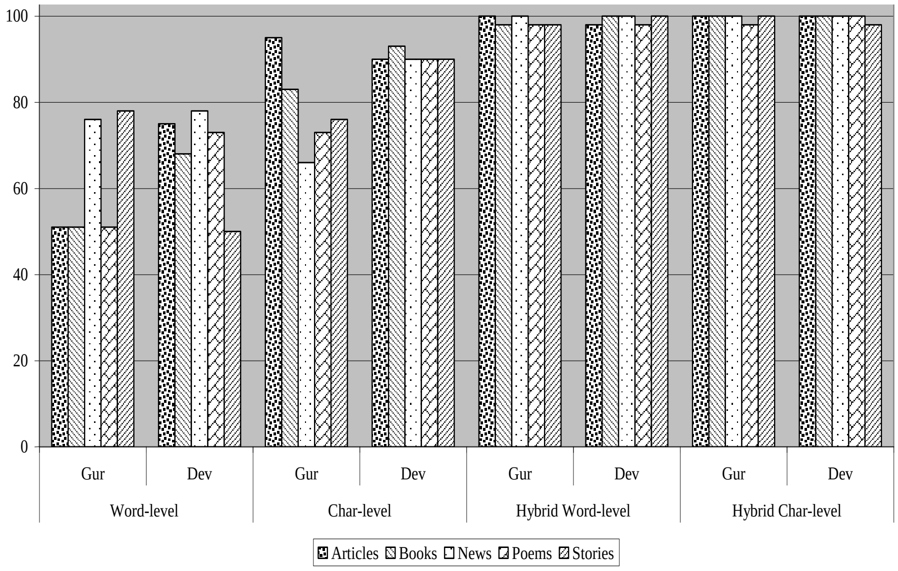


Fig. 8. Font detection accuracy of different methods.

Conversion Accuracy of the system is found to be 100% in all the test data.

Font Detection vs. Input word length. The figure 8 shows the trend of detection of Gurmukhi and Devanagari fonts when varying number of most frequent words in a language are taken as input to the system. The results shown in the following graph are based on the best detection method.

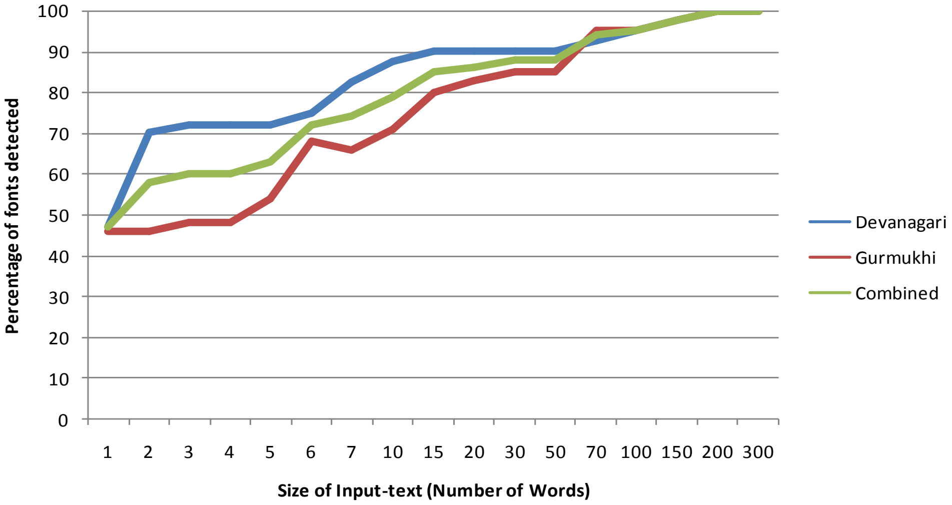


Fig. 9. Comparison of font-detection at varying sizes of input-text.

The different word sizes are from 1 to 300 words with a random interval. According to our observation the system is capable to identify 45% fonts correctly with just single word as input. Next, with two input words, there is sharp rise in Devanagari font detection from 45% to 70% while the Gurmukhi font detection is not affected. There is nearly 70% detection in Gurmukhi fonts is when the input size is at least six words. Unlike Devanagari fonts Gurmukhi fonts, detection decreases when input words are increased from six to eight. In overall trend we can say that during the input word size 1 to 70, the percentage of Devanagari font detection is always higher than the Gurmukhi font detection and reverse trend is seen between input word sizes greater than 70 to 125. Finally, it has been observed that input word size of around 200 words is reasonably sufficient for the system to give best detection results.

9 Conclusion

This paper describes a language and font detection system for Gurmukhi and Devanagari. It also delineates a font conversion system for converting the ASCII based text into Unicode. Hence, the proposed system works in two stages: the first stage suggests a statistical model for automatic language detection (i.e., Gurmukhi or Devanagari) and font-detection; the second stage converts the detected text into Unicode. The existing system supports around 150 popular Gurmukhi font-encodings and more than 100 popular Devanagari fonts. We have demonstrated the effectiveness of font detection is 99.6% and Unicode conversion is 100% in all the cases. Though we could not train our systems for some fonts due to non-availability of font converters but system and its architecture is open to accept any number of languages/fonts in the future without affecting its speed and performance.

References

1. Chaudhury, S., Sen S., Nandi, G.R.: A Finite State Transducer (FST) based Font Converter. International Journal of Computer Applications, Volume 58, No. 17, pp. 35–39 (2012)
2. Saini, T. S., Lehal, G. S., Chowdhary, S.K.: An Omni-font Gurmukhi to Shahmukhi Transliteration System, In: Proceedings of Conference on Computational Linguistics (COLING), pp. 313–319, Mumbai (2012)
3. Raj, A.A., Prahallad, K.: Identification and Conversion of Font-Data in Indian Languages, In: International Conference on Universal Digital Library (ICUDL2007), Pittsberg, USA (2007)
4. Singh, A.K.: Study of some distance measures for language and encoding identification, In: Proceedings of the Workshop on Linguistic Distances, Sydney, Australia, pp. 63–72, ACL (2006)
5. Singh, A.K., Gorla, J..: Identification of languages and encodings in a multilingual document. In: Proceedings of the 3rd ACL SIGWAC Workshop on Web As Corpus, Lovain-la-Neuve, Belgium (2007)
6. Cavnar, W. B., Trenkle J. M.: N-Gram-Based Text Categorization, In: Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, pp. 161–175 (1994)
7. Davis, M., Whistler, K. (Eds.): Unicode Normalization Forms. Technical Reports, Unicode Standard Annex #15, Revision 33. (2010) Retrieved February 22, 2011, from http://www.unicode.org/reports/tr15/tr15-33.html
8. Bharati, A., Sangal, N., Chaitanya, V., Kulkarni, A.P., Sangal, R.: Generating converters between fonts semi-automatically, In: Proceedings of SAARC conference on Multi-lingual and Multi-media Information Technology, CDAC, Pune, India (1998)
9. Kikui, G.: Identifying the coding system and language of on-line documents on the Internet. In: Proceedings of Conference on Computational Linguistics (COLING), pp. 652–657 (1996)

HPSG Grammar Treating of Different Forms of Arabic Coordination

Sirine Boukedi[1,3] and Kais Haddar[2,3]

[1] Faculty of Science Economy and management, University of Sfax, France

[2] Faculty of Sciences, University of Sfax, France

[3] MIR@CL laboratory, University of Sfax, France

sirine.boukedi@gmail.com, Kais.haddar@yahoo.fr

Abstract. Researchers working in Natural Language processing (NLP) found many problems, at different levels. The main problem encountered is the treatment of complicated phenomena, essentially the coordination. This phenomenon is very important. In fact, it is very frequent in various corpora and has always been a center of interest in NLP. Unfortunately, the few works working on this structure treated only some coordinated forms using constructed parsers which are generally so heavy. In this context, our work aims to develop a Head-driven Phrase Structure grammar (HPSG) representing all the different forms of Arabic coordination, based on a proposed typology. The constructed grammar was validated with Linguistic Knowledge Building (LKB). This system is designed for grammars specified in Type Description Language (TDL).

Keywords: HPSG grammar, coordination in Arabic, NLP.

1 Introduction

The coordination is an important linguistic phenomenon. It joins two or several compounds using conjunctions. However, there exist some cases where the elements composing a coordination structure are joined implicitly. This phenomenon interacts with many other syntactic phenomena, such as ellipsis and relatives. Therefore, there exist a large number of coordinated forms.

To treat the different cases of the coordination phenomenon, we should use a reliable formalism. In fact, a great representation leads to a correct syntactic analysis. In this context, appears the HPSG [14]. The choice of this formalism is justified. It is a unification grammar characterized by a reliable modeling and a complete representation of linguistic knowledge. Besides, HPSG proposes a modularized organization of linguistic knowledge. It minimizes the syntactic rules and attributes a great importance to the lexicon.

Therefore, our work aims to construct an HPSG grammar treating the simple forms of Arabic coordination, the interaction with ellipsis and relatives and some embedded forms. Thus, we start by proposing a typology classifying the coordinated constructions. Afterward, we adapted the HPSG grammar to represent the different forms of Arabic coordination. The established grammar was specified in TDL [12]. Indeed, resources developed in a specification language are easy to extend. Moreover, TDL is designed to support essentially the lexicalized grammatical theories such as HPSG formalism. Finally, our grammar was experimented with LKB. This system represents a parser generation tool, proposed by [6]. It is ergonomic and used standard parser algorithm, "chart parsing".

In this paper, we present some related works treating coordination structure. Then, we give the different forms of Arabic coordination and we studied some delicate cases. After that, we introduce the HPSG representation of the different forms. Then, we present the experimentation of the constructed grammar with Linguistic Knowledge Building (LKB) system and we evaluate the obtained results. This paper is enclosed by a conclusion and some perspectives.

2 Related Works

Researchers on coordination phenomenon started since 1970 for various languages. Our study showed that each work focused on some particular forms of coordination using different grammars. Indeed, this phenomenon has a very complicated structure and has different forms. Most of the related works considered that the coordination can be subdivided in two categories: constituent and non constituent coordination.

Biskri treated French coordination, essentially constructions based on the conjunction "et, *and*". In their work, they used the Applicative Combinatory Categorical Grammar (ACCG). Indeed, they conceived a schema for coordination of compounds having similar or different categories. Referring to the obtained results, they concluded that ACCG grammar is not reliable to treat complex phenomena. Indeed, the authors treated some forms and neglected other cases. Therefore, they didn't treat coordinated cases when the compounds have different function and nature.

Other researchers like [8] used Lexicalized Tree Adjoining Grammar (LTAG). They presented a general approach for elliptical constructions of coordination. The constructed grammar used trees to represent syntactic structures. This approach is based on fusion operations. According to the authors, this grammar has a delicate process of treatment. Moreover, it is expensive in terms of efforts and response time. Based on the obtained results, they concluded that the complexity of this process is exponential and depends of the number of derivations.

Unlike all these grammars, researchers working with HPSG found a great success in terms of reliability and complexity. Therefore, most of them such as [2], [4] and [5] used it in Natural Language Processing (NLP).

In [4], the author examines different coordination constructions. They treated Constituent Coordination, where the joined clauses are complete. Besides, their study covers also some cases of interaction with ellipsis phenomenon. In fact, they treated Argument Cluster Coordination (ACC), Right Node Raising (RNR) and the combination of ACC and RNR. To justify their choice of the used formalism, the authors started by a comparative study between CCG and HPSG. Based on their paper, they said that CCG were not sufficient, essentially for the ACC. Thus, they proposed an HPSG schema to represent the coordination forms mentioned below. Indeed, their work was inspired from some related works such as [7] and [16]. The obtained results were perfect. In fact, with HPSG grammar, the representation of the coordinated forms was perfectly clear. Besides, the number of ambiguities was reduced.

Other works, working with HPSG, focused on the coordination of particular categories such as [5]. In fact, he treated only Nominal Phrases (NP) coordination. Therefore, he proposed a compositional and constraint based approach for processing these constructions with HPSG framework with the goal of capturing complex semantic interactions that can arise in such structures. The obtained results were also encouraging. However, their study was interested on some particular categories and limited cases of interaction with ellipsis phenomenon. Their established grammar was insufficient to cover all the constructions.

For [2], she worked on coordination. According to her, the coordination has always been a problem for syntactic models and many problems were encountered. Indeed, there exists a problem in the treatment of coordination of constituents having different categories and of elliptical constructions. In this context, Abeillé proposed two different solutions: Categorical Grammar (CG) and HPSG grammar. The first solution inserts some predicates using operators like the logical ones. This solution has several disadvantages: the appearance of many ambiguities, the difficulty of using the operators and can't represent elliptical constructions. By the way, HPSG has a clear description of linguistic objects using SAV. The different representations were based on a detailed type hierarchy.

For Arabic language, some works treated Arabic coordination like [10]. The first contribution of this work consists of introducing a formal characterization of the ellipsis phenomenon interacting with the coordination one. The authors present a clause grammar to distinguish between well formed clauses and the uncompleted ones. To prove the feasibility of the proposed approaches, they developed a prototype called ERASE (Ellipsis Resolution of Arabic Sentences) and tested it on a corpus of elliptical Arabic sentences. The results obtained are satisfactory but the study on coordination phenomenon was done superficially. In conclusion, there is no existing work treating Arabic coordination adequately. Their study were incomplete and treats some forms of Arabic coordination.

Therefore, in this work we aim to construct an HPSG grammar treating all the possible forms of Arabic coordination. This specification is based on the proposed classification of Arabic coordination presented in the next section.

3 Classification of Arabic Coordination

According to the related works, the coordination can be subdivided in two categories: constituent and non constituent coordination. Like any grammar, these two kinds of coordination exist in Arabic language.

However, referring to much linguistics such as [1], the coordination joins the different compounds with two different ways: explicit relation (1) or implicit one (2). Therefore, the coordination can be classified on two principle categories: Coordinating attraction and explicative attraction.

(1) Taafa [‘alrijaalu fa ‘alnisaa’u] hawla ‘alk`abati
[Men and women] turned around the Kaaba
(2) Marartu bi [al faarisi `antara]
I passed by [the escapee Antara]

As representing in examples, the first category, coordinating attraction, requires particles. Already, the coordinated particles are called particles of attraction. In the following paragraph, we start by presenting the Arabic conjunctions.

3.1 Arabic Conjunctions

In Arabic grammar, the linguists such as [1] argued that there exist nine particles: و، فـ، ثم، حتى، أو، بل، أم، لكن، لا, (*wa, fa, thumma, hattae, ‘aw-*, *bal-, ‘am-*, *lakin-*, *lae*).

In some previous works, we have considered the conjunctions as non operative particles. In fact, this type of particle didn’t have any influence on the joining element. It only brings a semantic to the sentence. However, referring to some recent linguistic [11], the particles « حتى, *hattae*», « لكن, *lakin-* » and « بل, *bal-*» require some syntactic conditions.

The particle « حتى, *hattae*» requires that the attracted must be singular and not composed of any words (3). Moreover, it must be a part of the attractant (4) and achieve an augmentation or a diminution (5).

The particle « لكن, *lakin-* » requires also that the attractant must be singular; none attached to the particle « و, *wa* » and exists after a negation or interdiction (6). The same constraints for the particle « بل, *bal-*»: a singular attracted and after a negation, an interdiction or an affirmation.

(3) ‘akaltu [‘alsamakata hattae ra’sa haa]
I ate [the fish until her head]
(4) kadima [‘alhujjaju hattae ‘almuchaatu]
[The pilgrims even pedestrians] come
(5) maata [‘alnaasu hattae ‘alru’asaa’u]
[people even presidents] died
(6) maa dharabtu [zayda lakin- `amra]
I didn’t hit [zayd but amru]

All these examples represent some forms of coordinating attraction. We present below this type of coordination.

3.2 Coordinating Attraction

As we have mentioned above, the coordinating attraction is constructed with conjunctive particles. For Arabic language, the elements composing a coordinated structure can be complete or incomplete. Therefore, there exist two different categories: constituent coordination and non constituent coordination. The study on Arabic grammar showed that these two categories require particles. Therefore, we considered them as subtypes of the coordinating attraction.

Constituent coordination

The constituent coordination represents the case when the compounds composing a coordination phrase are complete. In fact, there is no lack in the coordination clause. The joined elements can have similar or different categories, as represented respectively in examples (7) and (8).

(7) [‘akala thumma naama] fi ‘aalmanzili
He [ate then slept] at home
(8) [‘akala wa bi sor`atiN dhahaba] ‘ila ‘al madrasati
He [ate and quickly went] to school

In fact, as represented in sentence (7), the conjunction “thumma, *then*” joins two similar categories (two verbal phrases). However, in the second sentence, it joins a sentence “bi sor`atiN dhahaba, *quickly went*” and a verb “‘akala, *ate*”.

Non constituent coordination

The non constituent coordination describes the interaction with ellipsis phenomenon, i.e., the case when one of the coordination structures lacks an element. According to [10], there exist four forms of ellipse: Right Node Raising (RNR), Left Node Raising (LNR), Gapping and VP-ellipse.

RNR represents cases of right factoring (8) in a sentence. In fact, the component factor is at the right of the sentence. Contrariwise, LNR designed the case when the component factor is at the left of the sentence (9). For the third form: Gapping, it represented discontinuities in the second compound of the coordination phrase (10).

Finally, the VP-ellipse represents the case when the verbal phrase is missed and replaced by a proverb (11).

(8) [‘akala] Mohamed tufaahataN wa ∅ ‘akhouhu ijaaSataN,
Mohamed ate an apple and his brother a pear
(8’) Mohamed [‘akala] tufaahataN wa ‘akhouhu ∅ ijaaSataN,

Mohamed ate an apple and his brother a pear
(9) ‘akalat- thumma naamat- [hadhihi ‘alkittatu],
She ate then she slept, this cat
(10)’istaykadha [‘aalwaladu] fa ghassala ∅ wajhahu,
The boy is waked up so hi washed his face
(11) ‘akala ‘aalwaladu wa kadhalika [faàla] ‘akhouhu,
The boy ate and so his brother

The study on Arabic grammar shows that sometimes when we transform a verbal sentence to a nominal one, we can switch from a form to another. (See example (8’)). In fact, after transformation, the example (8’) is no longer an RNR but a gapping form. Besides, there exist some cases when there is no particle in the coordination structure. It represents the explicative attraction. See section 3.3.

3.3 Explicative Attraction

The explicative attraction is a coordinated form which is not frequent in Arabic grammar. It is possible when the attracted is inert and represents an adjective to explain the attracted. Referring to [11], there exist four cases of coordination representing the explicative attraction. The first case represents the last name after the first name as represented in the next sentence.

(12) marartu bi ‘ahmadu helmy
I passed by Ahmed Hilmi

In fact, as we can see in this example the last name “helmy, *Hilmi”* comes to more precise the person that we have seen. The second case illustrates a name explained via a nickname. We give in the following an example of this case.

(13) ‘aldhakiyatu Amina
The smart Amina

As represented in this example, “Amina” which represents a name of a person is recognized through the nickname “‘aldhakiyatu, *the smart*”. Another case of explicative coordination is represented by the described after an adjective as represented in the following sentence, example (14).

(14) ‘alfaarisu àntara
The escapee Antara

Indeed “Antara” is described by an escape. Therefore he is called “‘alfaarisu, *the escape*”. Therefore, the adjective brought more clarity to the sentence. For the last case of explicative attraction, it is the easier case. Indeed, it illustrates the case of an explication after the explicated compound, as we can see below.

(15) àndy àusjuduN ‘ay dhahabuN
I have Asjudon i.e gold

As represented in (15), there exists an explicit explication of the term “àusjuduN, AsjuduN” with the conjunction “‘ay, *i.e*”. In all these cases, this type of coordination is very similar to the substitution phenomenon on the syntactic level that makes several cases of ambiguities. Indeed, it requires the same constraints to develop the HPSG schema representing the substitution phenomenon. Besides the two classes of coordination, there exist some delicate forms of this phenomenon. In the next section, we present the cases that we have treated through our work.

4 Delicate Forms

Like any grammar, Arabic grammar contains a variety of forms. This leads to several syntactic ambiguities. In the following, we present some delicate forms of Arabic coordination that we have treated.

4.1 Similar forms

The study on Arabic grammar shows that there exist many similar forms in syntax point of view. As example, the explicative attraction is completely similar to the phenomenon of substitution. Indeed, it requires the same syntax constraints for the phrase composition (16).

(16) maa ajmala [faatimatu bintu ‘alrasuli]
What a beautiful girl, [Faatimatu the prophet’s daughter]

In this example, the phrase putted between brackets represents a conflict case. Grammatically, it represents a phenomenon of substitution. However, at the syntactic point of view, it can also be an explicative attraction. Referring to [1], to solve this problem, there exist some criterions able to resolve this problem. Indeed, the explicative attraction is generally defined by components accompanied by others to specify them. Besides, the attracted must always be clearer that the attractive. Indeed, the second compound represents an explication of the first one. However, in a phrase substitution, it is possible to eliminate the substituted. This is obviously impossible in a coordination phrase.

4.2 Embedded Forms

In Arabic corpora, like in any other grammar, we can find many coordinated structures in the same sentence. This case illustrates the embedded forms. The sentence (17) represents an example of this kind of coordination.

(17) [Taafa ‘alrijaalu fa ‘alnisaa’u hattae ‘alSibyatu hawla
‘alkaàbati] thumma [Sallaw fi ‘albayti ‘alharaami]

[Men, women until boys turned around the Kaaba] then [pried in the Sacred house]

This sentence is composed from two sentences putted between brackets. They are joined with the conjunction "thumma, *then*". As we can see, in this example, the first sentence contains also another coordinated structure "[‘alrijaalu fa ‘alnisaa’u] hattae [‘alSibyatu]", where the two compounds are joined with the conjunction "hattae, *even*". Moreover, the first compound "‘alrijaalu fa ‘alnisaa’u" represents another coordinated structure using the conjunction "fa, *and*". So there exists encapsulation of three constructions in a same sentence. This illustrates the case of embedded forms.

It should be noted that an embedded form can be homogeneous: similar structures, heterogeneous structures of different or mixed nature. This type of structure is very delicate and leads to a great number of ambiguities.

4.3 Interaction with Others Phenomena

According to our study on Arabic grammar, we concluded that the coordination structure interacts with many others phenomena. Among these phenomena, we can mention ellipsis and relatives.

For ellipsis, as we have already mentioned, it represents non constituent forms of coordination. The different forms of this case is detailed in section 3.2 of the present paper

For relatives, the interaction with this phenomenon is very frequent in Arabic corpora. Referring to some works, the Arabic relative clause is a subordinate clause that can has all grammatical functions of a noun. This phenomenon has always many embedded forms that augment the degree of ambiguities (18).

(18) Jaa’a ‘alrajulu ‘alladhy `arafa ‘anna ‘albayta ‘alqadima ‘alladhy fy wasati ‘alqaryati mahjuwrun wa quara’a maa fy ‘alrisaalati ‘allaty wajadahaa
The man, who knew that the old house which is in the center of the village is deserted, came and read what in the letter that he found

The example above illustrates a combination of coordination and relatives. The coordination structure joins two verbal sentences with the conjunction "*wa*". The first sentence represents an embedded form of relatives. Indeed, the subject "*‘alrajulu ‘alladhy `arafa ‘anna ‘albayta ‘alqadima ‘alladhy fy wasati ‘alqaryati mahjuwrun*" contains two other relative clauses: "*’anna ‘albayta ‘alqadima ‘alladhy fy wasati ‘alqaryati mahjuwrun*," and "*‘albayta ‘alqadima ‘alladhy fy wasati ‘alqaryati*".

Based on the large study done on the Arabic coordination, we represented the different forms with HPSG formalism. The choice of this grammar is justified. In the next section, we give an overview on the HPSG representation.

5 HPSG for Arabic Coordination

HPSG is a unification grammar [14]. It is characterized by a reliable modeling of the universal grammatical principles and a complete representation of linguistic knowledge. This grammar is based on Attribute Value Matrix (AVM) for representation and a set of immediate domination schemata (DI schemata). The composition of the different structures is based on a set of principles (i.e., HFP Head Feature Principle).

According to some references working on coordination [2] and [15], this phenomenon was considered as a non-headed structure. In fact, the conjunction is a weak head. It inherits an important number of properties from its complement, essentially its head features. Moreover, it didn’t bring any modification on the adjoined compound.

For Arabic grammar, this criterion is also true. Indeed, an Arabic conjunction didn’t have any specification on the adjoined compound. It only differs at the level of VALENCE feature. We present, in the next section, the HPSG representation of Arabic conjunctions. Then we give an HPSG representation of two different conjunctions having different valence.

5.1 Conjunction SAV

To represent adequately the coordination particles, we have brought some modifications on the type hierarchy. Indeed, we have subdivided the unity on three categories: word, conjunction-word and phrase. The phrase, in its turn, is subdivided on coordinated and non coordinated phrases. In the following figure, we present the general AVM representing a coordinated particle.

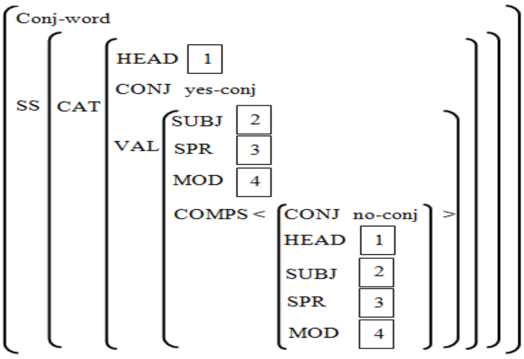


Fig. 1. General representation of an Arabic conjunction.

As we can see, in the figure below, a conjunction word follows the adjoined element. It has the same categorization like represented in the valence feature. The feature CONJ has two different values “yes-conj’, to specify the conjunction and “no-conj” for its complement. Moreover, the conjunction must have the same valence as its complement. In section 3 of the present paper, we have mentioned that there exist three conjunctions having influence on the adjoined compound. This difference appears on the VAL feature. The Fig. 2 gives an example of two different conjunctions.

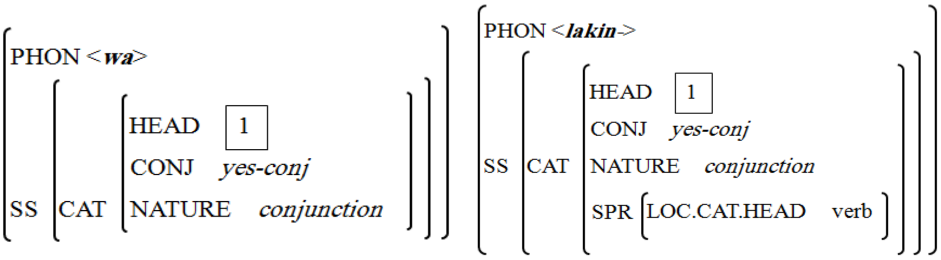


Fig. 2. HPSG representation of two different conjunctions.

As we can see, in Fig. 2, the conjunction "lakin-, *but*" is one of the operative particles. It requires as specification a negative verb. Taken into account these different constraints, we constructed different coordinated schemas. In the next section, we present an overview about the HPSG representation of the coordination schemata.

5.2 Coordination Schemata

As we have mention above, the conjunction is a weak constituent in a coordination structure. Therefore, we have constructed two different schemata. The following figure gives the general schema.

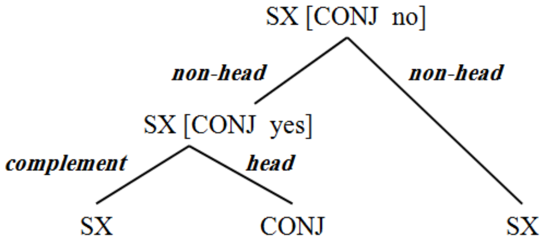


Fig. 3. Coordination structure.

Referring to [2], the conjunction is attached to the last compound to inherit its proprieties, using a head complement relation. This sub structure is relied with the other element of the coordination structure using a non-head relation. Therefore, we conceived two different schemas. The first one represents a headed structure. It represents a complement relation. The second schema joins this structure with the other elements composing the coordinated structure. Fig. 4 represents an example illustrating the general schema of a coordinated structure. This structure is based on an operative conjunction, the particle “lakin-, *but*”.

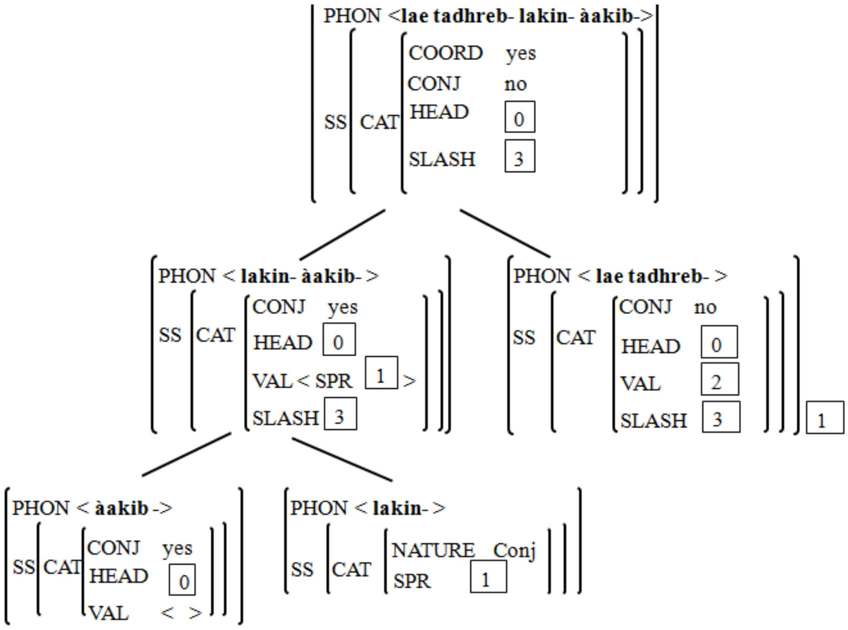


Fig. 4. HPSG representation of a coordinated structure.

In fact, we have added the feature COORD which expresses that the phrase is a coordination structure. Indeed, all the syntactic rules representing the coordination forms respect the proprieties presenting in the Figure above. To experiment the elaborated grammar, we have specified it in TDL. See next section.

6 Experimentation with LKB System

The experimentation of a constructed grammar is done throughout some stages. The modeling of HPSG grammar is based on a type hierarchy and a set of principles. To validate the constructed grammar, we should start by specifying it to proceed to the experimentation phase. In our work, we used Type Description Language (TDL) and the Linguistic Knowledge Builder (LKB) system.

6.1 TDL Specification

As we have mentioned above, HPSG formalism is based on AVMs (Attribute Value Matrix), to describe the different lexical entries and schemata representation. Each

AVM is composed from a set of features. The values attributed to each feature have a type. The different types are grouped hierarchically in the file “types.tdl”.

Besides this file, there exist others TDL files to specify the constructed grammar, essentially, “lexique.tdl”, “rlex.tdl” and “rsynt.tdl”. We give, in the following an extract from each file”.

TDL Specification of a Lexical Entry

To validate the constructed HPSG schemata, we need to add all the unities composing the different sentences in the file “lexique.tdl”. In Fig. 5, we give an example of the conjunction “lakin-, *but*”, specified in TDL.

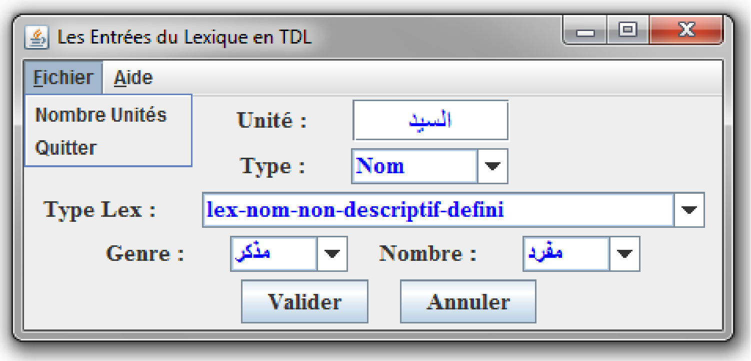
```
لكن := lex-conjonction-operative &
      [PHON <! “لكن”!>,
       SS.LOC.CAT.TETE[MAJ particule,
                        DEC non-decline,
                        PFORM conjonction,
                        NATURE حرف-عطف]].
```

```
lex-conjonction-operative := lex-conj &
   [SS.LOC.CAT|TETE conjonction-op,
                VAL [SPR<LOC.CAT.TETE
                     operative-verbe>]]].
```

Fig. 5. TDL specification of a conjunction.

As shown in Fig. 5, the lexical entry "lakin-, *but*" represents an instance of the type “lex-conjonction-operative”. This type regroups conjunctions having some constraints on the following compound. It should be noted that for more clarity, we specified the different constraints of each type of unity in the file “type-lex.tdl”.

The addiction of the different words in the lexicon is an easy task since the TDL specification is very similar to HPSG representation. However, this task requires many time. Therefore, we developed an application in JAVA “lex-editor”. In Fig. 6, we present the interface of this application.



Unité : السيد

Type : Nom

Type Lex : lex-nom-non-descriptif-defini

Genre : مذكر

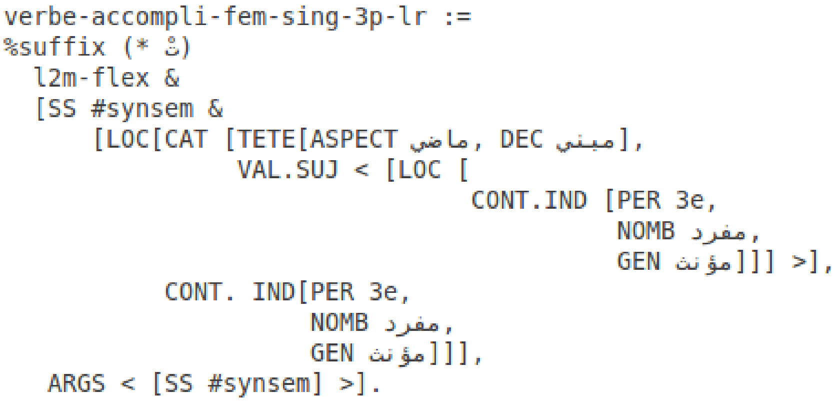
Nombre : مفرد

Fig. 6. Lex-editor interface.

Indeed, “Lex-editor” adds automatically the unities in the lexicon. All you have to do is to write the unity, specify its type and validate. Moreover, it checks the presence of the unity in the lexicon and accounts the number of entries.

TDL specification of a lexical rule

Besides, to make the lexicon extensional, we developed some lexical rules that generate automatically the derived forms of an entry. As example, we take the case of verbs. It is sufficient to give the root on the lexicon. The derivate forms after conjugation are done automatically via lexical rules as represented in Fig. 7.



```
verbe-accompli-fem-sing-3p-lr :=
%suffix (* تْ)
  l2m-flex &
  [SS #synsem &
      [LOC[CAT [TETE[ASPECT ماضي, DEC مبني],
                VAL.SUJ < [LOC [
                                CONT.IND [PER 3e,
                                          NOMB مفرد,
                                          GEN مؤنث]]] >],
           CONT. IND[PER 3e,
                     NOMB مفرد,
                     GEN مؤنث]]],
   ARGS < [SS #synsem] >].
```

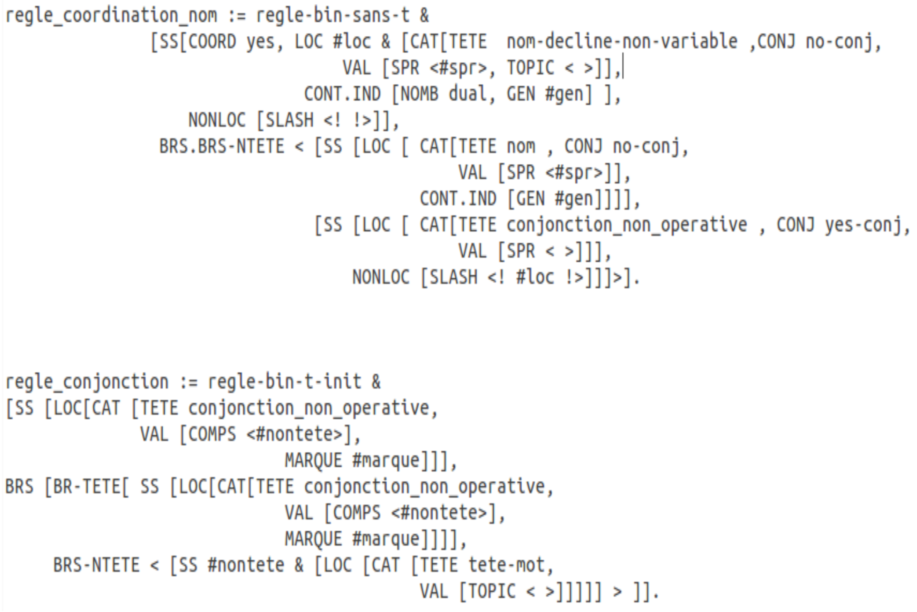
Fig. 7. Example of a lexical rule.

In fact, this rule is used to conjugate an infinitive verb in the third singular feminine word. Indeed, the term %suffix adds to the canonic form designed in the lexicon a termination. In the next section, we present the TDL specification of a syntactic rule treating Arabic coordination.

TDL Specification of a Syntactic Rule

In section 5, we have mentioned that to represent the coordination schema, we conceived two different schemas. The first one represents a headed relation composing the conjunction with the last compound. The second schema is a non headed structure that composes this phrase with the other elements.

To validate the constructed grammar, we used LKB system [6]. Indeed, this system is specialized for unification grammars such as the HPSG grammar. Moreover, many researchers like [9] and [13] used LKB to experiment their work and they obtained reliable results in a short time of response. Besides, this system is ergonomic and very easy to use. Indeed, LKB used standard parser algorithm, the “Chart parsing”.



```
regle_coordination_nom := regle-bin-sans-t &
               [SS[COORD yes, LOC #loc & [CAT[TETE  nom-decline-non-variable ,CONJ no-conj,
                                   VAL [SPR <#spr>, TOPIC < >]],
                               CONT.IND [NOMB dual, GEN #gen] ],
                   NONLOC [SLASH <! !>]],
                BRS.BRS-NTETE < [SS [LOC [ CAT[TETE nom , CONJ no-conj,
                                               VAL [SPR <#spr>]],
                                           CONT.IND [GEN #gen]]]],
                                [SS [LOC [ CAT[TETE conjonction_non_operative , CONJ yes-conj,
                                               VAL [SPR < >]]],
                                    NONLOC [SLASH <! #loc !>]]]>].



regle_conjonction := regle-bin-t-init &
[SS [LOC[CAT [TETE conjonction_non_operative,
              VAL [COMPS <#nontete>],
                             MARQUE #marque]]],
BRS [BR-TETE[ SS [LOC[CAT[TETE conjonction_non_operative,
                             VAL [COMPS <#nontete>],
                             MARQUE #marque]]]],
     BRS-NTETE < [SS #nontete & [LOC [CAT [TETE tete-mot,
                                           VAL [TOPIC < >]]]]] > ]].
```

Fig. 8. A coordination schema in TDL.

6.2 Evaluation of the Obtained Results

According to the related results, LKB was considered as the best system to validate constructed grammars. Therefore, in the present work, we used it to experiment the developed HPSG. This grammar was tested based on test corpus extracted from the Arabic Tree Bank, ATB. This corpus contains several texts covering a great number of syntactic forms. The table below presents the different treated forms and gives the result of each form.

Table 1. Obtained results.

Treated forms		**Number of sentences**	**Results**	
			Fail	**Success**
Coordinated structures	*Simple*	40	4	36
	With relatives	100	30	70
	With ellipsis	180	105	75
	Embedded	50	5	45
Simple structures		230	21	209
		600	165	435

Before commenting these results, it should be noted that the success cases represent sentences having one analysis tree similar to the syntactic representation. Otherwise, the failure represents ambiguities or sentences didn't have any tree parse.

As highlighted in Table 1, the simple structures are treated very well. In fact, 91% represents success cases. The fail forms are due to the lack of words in the lexicon. For the coordination structures, among 370 structures, there exist 226 having one parse tree. So the recall has as value 61%. However, among these 226 sentences, only 157 are correct grammatically that makes the precision as 69%. The following sentence represents one of the successful cases.

> (19) Taafa [‘alrijaalu fa ‘alnisaa’u hattae ‘alSibyatu *‘alladhyna taìbue fy ‘albayti ‘alharaami*]
> *Men and women even little boys, who were tired in the scared house, walk.*

This sentence represents an embedded form and interacts with the relative phenomenon. As we can see in (19), the principle coordinated structure is constructed with the conjunction “hattae, *even*”. The first compound represents another coordinated structure. The second one interacts with relatives represented in italics. The result of this sentence is represented with the following parse tree.

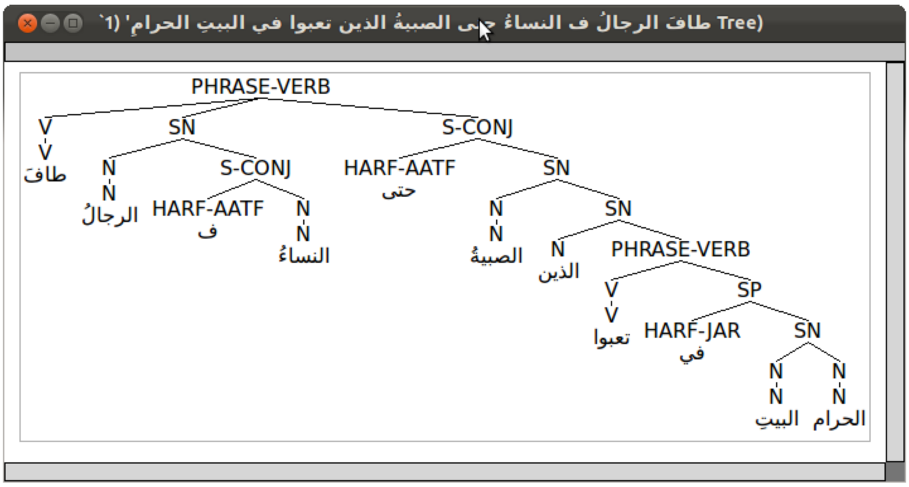


Fig. 9. Parse result of the example (19).

The ambiguity cases come from the similarity of many Arabic phenomena. In section 4.1 of the present paper, we have mentioned an example. For the sentences which didn't have any parse, there exist different reasons. Among these causes, we can mention the absence of different entries in the lexicon. Besides, there exist some others phenomena which are not treated, essentially the juxtaposition. This structure joins many compounds via comma. It is very frequent in Arabic corpora and interacts sometimes with coordination.

7 Conclusion and Perspectives

In the present paper, we started by presenting some related researchers working on coordination. Then, based on a large study, we give a classification for Arabic coordination. Indeed, we studied the different Arabic conjunctions and many delicate forms of this structure. Among these forms, we focused on the embedded forms and the interaction cases with the other phenomena. After that, based on the proposed classification, we represented the different coordination schemata in HPSG. The constructed grammar was specified in TDL and validated with LKB system. The experimentation was done on a corpus of 600 sentences. According to the obtained results, we evaluated our grammar.

As perspectives, we are working on the juxtaposition since it is very frequent in Arabic grammar. This phenomenon is very delicate and represents a resource of ambiguities. Moreover, we are working to ameliorate the syntactic rules to give best results. Thus, we will treat other particular phenomena and specify more constraints to eliminate the ambiguous cases. Furthermore, we aim to construct a converter permitting to convert the lexical entries of XML in TDL in order to facilitate the development of the lexicon.

References

1. Abdelwahed A.: *الكلمة في التراث اللساني العربي*, Librairie Aladin 1ère édition (2004)
2. Abeillé A.: Coordination: two challenges for syntactic theories, LLf University, Paris 7 (2006)
3. Biskri I., Desclés J.: Coordination of different categories in French, Quebec University, Canada (2006)
4. Breavers J., Sag I.: Coordinate Ellipsis and Apparent Non-Constituent Coordination. In: Proceedings of the HPSG04 Conference (2006)
5. Chaves R.: Underspecification and NP coordination in constraint based grammar. In: Workshop on Empirical challenges and Analytical alternatives to strict compositionality. Heriotwatt university, Edinburgh, Scotland, pp. 38–58 (2005)
6. Copestake A.: Implementing Typed Feature Structure Grammars. CSLI publications (2002)
7. Crysmann B.: An Asymmetric Theory of Peripheral Sharing in HPSG: Conjunction Reduction and Coordination of Unlikes in formal Grammar. (2003)
8. Djamé S., Benoît: Modélisation et analyse des coordinations elliptiques par l'exploitation dynamique des forêts de dérivation. In: TALN, Lleuven (2006)
9. Garcia, O.: Une introduction à l'implémentation des relatives de l'espagnole en HPSG-LKB. Research memory (2005)
10. Haddar K., Ben Hamadou A.: An Ellipsis Resolution System for the Arabic language. The International Journal of Computer Processing of Languages, 22(4):359–380 (2009)
11. Hamad Kh., Aidi H.: أثر العطف في التماسك النصي في ديوان على صهوة الماء . Journal de l'Université islamique de recherches en sciences humaines, pp. 327–356 (2012)
12. Krieger H., Schäfer U.: TDL: A Type Description Language for HPSG. Part1: Overview. Technical reports, Deutsches Forschungszentrum für Künstliche Intelligenz, Saarbrücken, Germany (1994)

13. Laurens F.: Implémentation des types de phrases et des types de constructions coordonnées du Français avec la plateforme LKB. Technical Report, laboratoire LLF (2007)
14. Pollard C., Sag I.: Head-Driven Phrase Structure Grammars. CSLI Lecture Notes, Chicago University Press (1994)
15. Tseng J.: La grenouille: Grammar report, Delph-In summit (2007)
16. Yatabe S.: A Linearization Based Theory of Summative Agreement in Peripheral-Node Raising Constructions. In: Proceedings of the 2002 HPSG Conference, Stanford, CA: CSLI Publications (2002)

Exploring the Partial Textual Entailment Problem for Bengali News Texts

Amitava Das[1] and Dwijen Rudra Pal[2]

[1] Department of Computer Science and Engineering,
University of North Texas, Denton, Texas,
USA

[2]Computer Science and Engineering Department,
National Institute of Technology, Agartala,
India

Abstract. Formal definition textual entailment implies strict meaning relationship of meaning in its totality between text (*T*) and hypothesis (*H*). Even if the text have main ideas of a hypothesis, but lacks minor information or have additional information then treated as an entirely unrelated text. In these cases we are left with no sense of how close (*T*, *H*) were to entailment. In various applications of entailment, major attention has given on strict entailment only. However, in reality strict entailment cases are relatively lower in compared to partial entailment cases, are prevalent. We introduce the idea of partial entailment in this paper and defining it empirically. We have developed corpus and finally proposed baseline architecture for automatic identification of partial textual entailment. This work is on Bengali news texts.

Keywords: Partial textual entailment, Bengali news texts.

1 Introduction

The automatic recognition of textual entailment is one of the difficult under research Natural Language Processing (NLP) problems. In the last decade, automatic textual entailment research received significant research attentions but majority of the works dealt with strict definition of entailment, whereas in reality strict entailment cases are relatively lower. Moreover, majority of such prior works concentrated on English. Here in this paper we are proposing the idea of partial textual, as a bidirectional relationship between pairs of statements for Bengali. The standard definition of textual entailment has been considered as a unidirectional problem so far where a given text *T* would be considered as entailed to another text *H* (hypothesis) if the meaning of the *T* could be completely inferred from the *H*. We are extending the formal definition of the entailment and empirically defining the concept of partial entailment. Let us consider "*Text1*" or *(T1)* and "*Text2*" or *(T2)* are partially entailed with each other. Both the statements *T1* and *T2* would be considered as entailed to each other while partial meaning of *T1* could be inferred from the partial or complete meaning of *T2* or vice versa. We are also proposing different classes of partial entailment by breaking down both the statements *T1 and T2* into additional information to compare partial matching. We preserve the original definition of the complete entailment. Our main motivation for this work was to investigate the idea of partial textual entailment, and assess credibility of existing automatic complete textual entailment detection methods for the partial entailment problem, otherwise finally explore for new methodologies.

The rest of the paper is organized as follows. In the next sections, we formalize our procedure by empirical definition of partial entailment in Section 2, Corpus Acquisitioning in Section 3, a baseline system and performance in Section 4, preparing related work on partial textual entailment in Section 5, and finally, we draw our conclusions in Section 6.

2 Partial Entailment: The Empirical Definition

We define following four detailed categories to represent partial entailment.

1. **Type1:** If both the sentences are having same information and meaning same, then it is a case of direct entailment and should be noted as YES (X=Y). This category is the perseverance of the original entailment definition. Example:

   Sentence 01: মৃত্যুদণ্ড নয় যাবজ্জীবন আফতাবের।

   Eng. Gloss: Aftaab has been life sentenced instead of sentence to death.

   Sentence 02: ফাঁসি রদ করে আফতাবকে যাবজ্জীবন দিল সুপ্রিমকোর্ট।

   Eng. Gloss: Supreme Court has cancelled aftaab hanging and had given him life sentenced

   Entailment Status: *YES (X=Y).*

2. **Type2:** If the second sentence has all the information of the first sentence and has some extra information, then it is a case of partial entailment of type1 (X=X+Z). System also marked the repeated information section in the second sentence. Example:

   Sentence 01: (ফাঁসি থেকে রেহাই পেয়েছে এই ঘটনার আর এক দোষী জামিলউদ্দিননাসির।)

   Eng. Gloss: Jamiluddin Nasir, another offender of this incident is exempted from hanging

   Sentence 02: আজ দেশের সর্বোচ্চ আদালত আফতাব ও তার সঙ্গী (জামিলউদ্দিন নাসিরের ফাঁসি রদ করল।)

   Eng. Gloss: Today Supreme court has cancelled hanging of Aftaab and his companion Jamiluddin Nasir

   Entailment Status: YES (X=X+Z)

3. **Type3:** If the first sentence has all the information of the second sentence and has some extra information, then it is a case of partial entailment of type3 (X+Z=X+Y). Moreover please mark the repeated information section of the first section. Example:

   Sentence 01: দীর্ঘ শুনানির পর (নিম্ন আদালত দোষীদের মৃত্যুদণ্ড ঘোষণা করে।)

   Eng. Gloss: After a long hearing lower court has declared sentenced to death of the offenders.

   Sentence 02: জঙ্গিদল আসিফ রেজা কমান্ডো ফোর্সের সদস্য আফতাব জামিলউদ্দিন সহ সাতজনকে (নিম্ন আদালত মৃত্যুদণ্ড দেয়।)

   Eng. Gloss: Lower court has declared sentence to death to Seven members of the terrorist group of Asif reja force along with Aftaab, Jamiluddin.

   Entailment Status: YES (X+Z=X+Y)

4. **Type4:** If both the sentences are not having same information then it is a false entailment and marked NO status. Example:

   Sentence 01: ফাঁসি থেকে রেহাই পেয়েছে এই ঘটনার আর এক দোষী জামিলউদ্দিন নাসির।

   Eng. Gloss: Jamiluddin Nasir, another offender of this incident is exempted from hangingSentence 02: প্রায় এক বছর চলা শুনানি শেষ হয় এমাসেই।

   Eng. Gloss: Almost one year hearing has finished in this month

   Entailment Status: NO

Here in all the cases X, Y and Z are abstract representation of a block of information. The 4th category is basically the negative example. We marked the common information boundaries for all the sentence pairs. For our automatic partial entailment detection task we prefer to detect common information boundaries for the both sentences beyond the original binary classification, because then it will be useful to use those outputs further for any NLP task like summarization, QA, or else. The empirical question we asked to ourselves is how much extra information should be the upper limit for the partial entailed sentence pairs. For example we cannot claim that the following two sentences are partially entailed.

Sentence 01: গত তিন দিন ধরে চলা যাবতীয় জল্পনার অবসান ঘটিয়ে পাকিস্তান আজ জানিয়ে দিল সোমবার নরেন্দ্র (মোদীর শপথগ্রহণ অনুষ্ঠানে যোগ দিতে আসছেন নওয়াজ শরিফ)।

Eng. Gloss: Over three days, ending speculation Pakistan confirmed that Nawaz Sharif would attend the oath ceremony of Narendra Modi on Monday.

Sentence 02: মোদির শপথে নওয়াজ শরিফ।

Eng. Gloss: Nawaz Sharif at Modi’s oath ceremony.

Entailment Status: NO

Here in the first sentence there is lots of more information than the second one. So, we define out upper threshold as the following equation.

$$\frac{\left|n_1^w - n_2^w\right|}{(n_1^w + n_2^w)/2} * 100 \pounds 35\%$$

Here n_1^w is the total number of words in the sentence one and n_2^w is the total number of words in the second sentence. In our definition of partial entailment we kept the number of words differences within 35%. To compare we checked the mean word count difference in the standard RTE (Recognizing Textual Entailment) corpus[1] and we found it is to be 75-80% on an average. So, empirically we have restricted more than 2 times for of the original textual entailment definition.

3 Corpus Acquisition

We designed a semi-automatic corpus acquisition process, because it helps on removing rigorous manual efforts and expedite the overall process. We collected news texts on specific events from multiple Bengali news sources, i.e. news stories on same event published in different newspapers on the same day. Two most popular Bengali newspapers Aajkaal[2], and Anandabazar[3] are chosen for this task. During the selection of the source texts, we gathered Bengali news text corpus of 25 topics containing news stories on those events in the cited two newspapers. From the original HTML text we kept only the unformatted content text, without any images, tables or links. Further, while choosing topics we made sure those topics covering various domains like international politics, national politics, sports, natural disasters, political campaigns and elections.

Here from the next paragraph onwards various steps of automatic semi-automatic annotation task have been discussed. We have also included some useful negative examples, are lexically very similar but not actually entailed.

3.1 Stop Word Removal

Stop/junk words such as অবশ্য *(sure)*, অনেক *(many)*, অন্তত *(at least)*, অথবা *(or)*, অথচ *(still)*, আজ *(today)* are removed automatically. The stop word list for Bengali has been collected from [1].

3.2 Tokenizing and Part-of-Speech (POS) Tagging

A tokenizer has been developed for Bengali text. The Bangla POS-Tagger, developed by [2, 3] has been used for the present task.

[1] http://pascallin.ecs.soton.ac.uk/Challenges/RTE/Datasets

[2] http://www.aajkaal.net/

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

3.3 Stemming

Stemming is the process of generating surface word forms to their root forms. For example, the plural forms of a noun such as ‘সেন্টারের’ (center’s) are stemmed into ‘সেন্টার’, (Center) and ‘আফতাবের’ (Aftab’s) are stemmed into ‘আফতাব’ (Aftab) for the present task. Some of the most frequent Bengali suffixes are ‘ই’, ‘গুলো’, ‘টা’, ‘টি’, ‘রা’, ‘হীন’. We have used the system described in the [4], with some simple additional modifications.

3.4 Content Words Extraction

At this stage bag of content words have been collected from each sentence to further measure cosine similarity between sentences. Here bag of content words defined [5] by only four open POS classes: nouns, verbs, adverbs and adjectives. The used POS tagger [4] generates two sub-categories for Noun; Verb has two sub-categories as verb finite and verb auxiliary. Adverb and Adjective does not have any more subcategories.

3.5 Measuring Cosine Similarity

The simplest way to describe a binary sentence vector is as the set of its non-zero values. Cosine similarity is a measure of similarity between two *n*-dimensional vectors obtained by finding the cosine of the angle between them. It is often used to compare documents in text mining. In addition, it is used to measure cohesion within clustering data mining. Cosine similarity is also widely used in information retrieval to calculate the similarity between documents or sentences. Given two vectors of attributes, *A* and *B*, the cosine similarity θ is calculated using the dot product and magnitude as:

$$\cos(A,B) = \frac{|\mathrm{A} \cup \mathrm{B}|}{\sqrt{|\mathrm{A}| \times |\mathrm{B}|}} \tag{1}$$

We consider binary vectors, that is, vectors with entries that are either 0 or 1. We converted each sentences into binary vector. Then calculates the cosine similarity for all the sentences present in file 01 with other files within the same topic cluster. One example given below to show the similarity value. These lines are from the original texts after stemming of content words. For example,

sentence 1: মৃত্যুদণ্ড আফতাব যাবজ্জীবন,

sentence 2: আফতাব যাবজ্জীবন দেওয়া হয়েছ,

Cosine Similarity Score (1,2) = 67.082.

Then we ended up with various sub-groups of possible partial entailed pairs. For further manual checking, we chose a cosine similarity threshold of ^{3}15 experimentally. It has found that almost all the actual entailment cases where cosine similarity value is less than the 15 of maximum cosine similarity value, no entailment relation comes.

3.6 Manual Annotation and Agreement

For the human annotation we involved two different human annotators, they are undergraduate students (not linguist) and native Bengali speakers. To assess annotation agreement Cohen's Kappa [6] coefficient has been measured on a small subset. We have chosen one topic: two files having 144 comparisons altogether tagged by the two annotators separately. A detailed categorical distribution of the two annotator's markings is reported in the following Table 1.

Table 1. Categorical distribution for the agreement annotation.

	Categories			
	X=Y	**X=X+Z**	**X+Z=X+Y**	**NO**
Annotator 01:	4	7	4	129
Annotator 02:	5	8	2	129

We found the sentence level *kappa* is 0.92. To understand the common information boundary detection agreement we choose Mean Agreement precision (MAP) metric. For the Type 2 (X=X+Z) it is 0.98 and for the Type 3 (X+Z=X+Y) it is 0.976, which is indeed higher implies that the task is not much ambiguous.

3.7 Corpus Statistics

Finally we ended up with 245 tagged pairs, it is an ongoing task. Here are the details of corpus statistics. All the negative examples, marked as not entailed by the annotators are been kept for further evaluation during training and testing. For the negative example inclusion cosine similarity threshold is $^{3}10$. A detail of the generated corpus is reported in the Table 2. This is an ongoing task.

Table 2. Categorical distribution for the agreement annotation.

Categories	**Sentence Pairs**	**Avg. CS**
X = Y	102	54.68
X=X+Z	127	16.41
X+Z=X+Y	16	23.12
Neg. Exmp.	7,349	10.19

4 The Baseline System and Performance

At this stage our motive is to develop an automatic system, can identify partial textual entailment (PTE) and can classify them into defined classes. We have developed a very basic system and the accuracy is not very encouraging but we are reporting the initial results to establish the fact that the partial entailment detection is more challenging than the standard definition of the entailment. This is an ongoing task.

Pakray et al. (2011) [10] reported decent performance for their rule based automatic textual entailment system using lexical and syntactic features. Reported lexical features were WordNet based Unigram Match, Bigram, Longest Common Subsequence (LCS), Skip-grams and they stemmed throughout before each of the feature compilation. Syntactic features were Subject, Object, Noun, Verb, Preposition, Determiner and Number. We drew our inspiration from this task and applied those lexical features on our data to observe the effect. We are unable to use syntactic features because there is no good quality dependency parser available for Bengali. Syntactic features extraction is our future target.

There is no WordNet (Bengali) available publicly so we are unable to use that feature. Therefore, we did our experiment with only Unigram Match, Bigram, Longest Common Subsequence (LCS), Skip-grams and we have used stemming before each feature extraction. All the features are self explanatory except Skip-grams. A skip-gram is any combination of n words in the order as they appear in a sentence, allowing arbitrary gaps. In the present work, only 1-skip-bigrams are considered where 1-skip-bigrams are bigrams with one word gap between two words in order in a sentence. Our strategy is relatively simple. Pakray et al, 2011 [10] reported their accuracies on the RTE 1-5 datasets as in the following Table 3.

Table 3. Pakray et al. (2011) reported RTE accuracies.

Dataset	Accuracy	Baseline PTE
RTE1	0.537	0.49
RTE2	0.592	0.51
RTE3	0.610	--
RTE4	0.554	--
RTE5	0.603	--

Pakray and his colleagues did not mention any implementation details how these features helped on the final entailment decision and how all these features values accumulated to reach out the final result. Moreover they did not provide any feature ablation to understand what is the effect of a particular feature. We replicated the system using these formulations.

Table 4. Lexical features meanings.

Type	**Unigram**	**Bigram**	**LCS**	**SkipGram**
	$(u_m/n)*100$	$(b_m/n)*100$	$(lcs_m*l^{avg}/n)*100$	$(sg_m*l^{avg}/n)*100$
X = Y	52	33	16	20
X=X+Z	14	9	4	10
X+Z=X+Y	21	12	9	14
Neg. Exmp.	10	8	2	6

where u_m is the total number of matched unigrams and n is the average number of words in both the sentences. b_m is total number of matched bigrams. lcs_m is the numbers of LCS matched whereas l^{avg} is the average length of those matched strings. sg_m is the numbers of skip-gram matched and l^{avg} is the average length of those skip-grams. Reported numbers are the mean values of those features learned from the training set.

Table 5. Baseline PTE with basic lexical features.

Type	Accuracies
X = Y	0.47
X=X+Z	0.39
X+Z=X+Y	0.35
Neg. Exmp.	0.56

We split our data into training (65%) and test set (35%). This split was class wise. Those learned feature wise mean values have been used further to detect partial entailment classes on the test set. Feature values exceeding these means resulting *yes* decision and feature values below the means is resulting a *no* decision. Initial results reported in the following table 5. We even tried the same setup on RTE 1 and 2 data as reported in the last column in the table 3.

5 Related Works

The concept of the partial textual entailment was first presented by Nielsen and his colleagues [7] in the year of 2009. Their work was on student's responses to an automated tutor's question. Partial entailment was used to understand the overlap between student answers. To detect proposed system broke sentences into fine-grained semantic facets, derived roughly from syntactic dependencies, and checked whether those facets were overlapping. Their work provides a finer-grained annotation schema to indicate more precisely the entailment relationship between the student's answers and that facet of the reference answers.

Instead of binary textual entailment decision in the form of yes or no, the proposed method in Nielsen et al work break down reference answer into semantic facets which refer to some part of a text's meaning. They also propose more expressive annotations labels in order to specify entailment relationship more clearly. They used eight finer annotation labels named: Assumed (facets that are assumed), Expressed (Facet that are directly expressed), Inferred (Facets inferred), Contra-Expr (Facets directly contradicted by negation), Contra-Infr (Facets contradicted by pragmatics), Self-Contra (Facets that are contradicted and implied), Diff-Arg (Facets where core relation expressed) and Unaddressed (Facets not addressed at all). In this model of facets, where each such facet is a pair of words in the hypothesis and the direct semantic relation connecting those two words. In comparison, we identified common information between *T* and *H* in terms of semantic similarity, which defines semantic inference more precisely for the sake of partial entailment.

Agirre et al. explicitly defined in their work [8] in 2012, different levels of semantic text similarity between two sentences. This system proposed 5 levels of similarity starting from 0 to 5. Level 0 defines no similarity, 1 defines not equivalent but same topic, 2 defines not equivalent but share same details, 3 defines roughly equivalent with missing of important information, 4 as mostly equivalent but some unimportant information differ and 5 as completely equivalent having same information. Though this model provides finer grained similarity notions, it is still not appropriate for semantic inference, as similarity was not well defined enough.

After these works, there is no more work on partial textual entailment until Omer Levy et al work [9] published last year i.e. 2013. In this work, they investigate the idea of partial textual entailment, and assess whether existing complete textual entailment methods can be used to recognize it. In their work partial textual entailment has defines as breaking down the hypothesis into components, and attempting to recognize whether each one is individually entailed by text. This definition concentrated on whether a single element of the hypothesis is entailed or not.

In our work, we proposed two detailed categories of partial entailment with further identification of common information boundaries in both the entailed sentences, which is a first approach in the area of partial textual entailment. This identification will be helpful for any NLP task like summarization, QA.

6 Conclusion and Future Work

In conclusion, we would like to mention that defining various classes of partial textual entailment is the main contribution of this task. Research works on partial entailment is an untouched paradigm so far. Moreover, with best of our knowledge this is the first paper discussing about the entailment problem for the Bengali.

This is an ongoing task. We are collecting more data and experimenting various automatic processes for the partial entailment detection. We are also applying same setup on social media text i.e. tweets.

References

1. http://www.isical.ac.in/~clia/resources.html.
2. A. Senapati, U. Garain: GuiTAR-based Pronominal Anaphora Resolution in Bengali, ACL, pp. 126–130 (2013)
3. http://www.isical.ac.in/~utpal/resources.php
4. Dolamic, L., Savoy, J.: Comparative Study of Indexing and Search Strategies for the Hindi, Marathi and Bengali Languages. ACM Transactions on Asian Language Information Processing, 9(3) (2010)
5. Winkler, E.G.: Understanding Language. Continuum. pp. 84–85 (2007)
6. Cohen J.: A coefficient of agreement for nominal scales. Educational and Psychological Measurement, 20(1):37–46 (1960)

7. Rodney D Nielsen, Wayne Ward, James H Martin. Recognizing entailment in intelligent tutoring systems. Natural Language Engineering, 15(4):479–501 (2009)
8. Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre: SemEval-2012 Task 6: A pilot on semantic textual similarity. In: Proceedings of the 6th International Workshop on Semantic Evaluation, in conjunction with the 1st Joint Conference on Lexical and Computational Semantics, pages 385–393, Montreal, Canada (2012)
9. Omer Levy, TorstenZesch, Ido Dagan, Iryna Gurevych: Recognizing Partial Textual Entailment. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL, Volume 2: Short Papers), pages 451–455, Sofia, Bulgaria (2013)
10. Partha Pakray, Alexander Gelbukh, Sivaji Bandyopadhyay: Textual Entailment using Lexical and Syntactic Similarity. International Journal of Artificial Intelligence and Applications, Vol. 2, No. 1, DOI 10.5121/ijaia.2011.2104 , pp. 43–58 (2011)

Acoustic Vowel Analysis in a Mexican Spanish HMM-based Speech Synthesis

Marvin Coto-Jiménez[1,2], Fabiola M. Martínez-Licona[2], and John Goddard-Close[2]

[1]Universidad de Costa Rica, Electrical Engineering School,
San José, Costa Rica

[2]Universidad Autónoma Metropolitana, Electrical Engineering Department,
Mexico City, México

marvin.coto@ucr.ac.cr{fmml,jgc}@xanum.uam.mx

Abstract. The synthetic voice produced from an HMM-based system is often reported as sounding muffled when it is compared to natural speech. There are several reasons for this effect: some precise and fine characteristics of the natural speech are removed, minimized or hidden in the modeling phase of the HMM system; the resulting speech-parameter trajectories become oversmoothed versions of the speech waveforms. In order to obtain more natural synthetic voices, different training conditions must be tried in the construction of the HMMs. One of the most important issues related to the obtained synthetic voice is that of quality assessment. There are several ways to address this, from subjective to objective approaches, applied to different parameters. This paper presents a comparative analysis of certain acoustic features derived from synthesized speech which has been obtained using different training configurations. Pitch, jitter and shimmer were extracted from the synthesized versions of three training sets of vowels of a Mexican Spanish speech database: the normal training set and sets with alterations in the context and fundamental frequency F0. The results show that these objective features can be part of an adequate quality assessment of synthetic speech.

Keywords: HMM, speech synthesis, jitter, shimmer, pitch.

1 Introduction

The speech production process may be described using the source filter theory of voice production [5], as shown in Fig. 1. This model is called the source-filter model.

Speech synthesis can be realized using this model, e.g. the Klatt synthesizer: voiced and unvoiced speech sounds are produced by applying a source, defined by a pulse train or white noise, to an LTI filter. The LTI filter serves as the vocal tract.

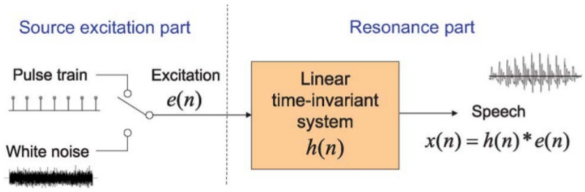


Fig. 1: Source-filter model of speech production [15].

Other methods have been used for speech synthesis, perhaps the most successful to date being that of concatenative synthesis. Recently, HMM-based synthesis, also known as Statistical Parametric Synthesis, has been introduced.

In HMM-based speech synthesis, HMMs are used to generate the speech waveform by modeling pitch, duration, and spectral information, usually in the form of fundamental frequency, f0, parameters, as well as mel frequency cepstal coefficients (MFCC).

HMMs are trained using data obtained from real speakers, and a maximum likelihood criterion is employed to generate these speech parameters [16].

Synthesized speech produced using this technique has been reported as sounding muffled when compared to natural speech, because the generated speech-parameter trajectories are often oversmoothed [15] [1]. This means that detailed characteristics of the speech parameters are removed in the modeling stage, and the model's output is unable to reproduce them.

In order to identify which synthetic voices have better quality, there are several possible approaches; these range from applying either subjective or objective measures. The most popular method for evaluating the quality of synthesized speech is that of a subjective listening test [8].

Several proposals have addressed the issue of relating objective measures to subjective measures such as [3,18,9,14,13,12,17]. In this paper we present a comparative analysis of three acoustic measures: Jitter, Shimmer and Pitch. These acoustic parameters have been used to analyze stress in human speech [11], as well as various pathologies [4] [19], including vowel analysis [2] to mention a few.

We propose extending these studies of pitch, jitter and shimmer, to that of comparing natural and synthesized speech with the objective of using these parameters to aid in assessing the quality of the synthesized voice. More precisely, we use a Spanish speech database consisting of two professional speakers, one male and one female, and under different training conditions employ HTS, an HMM-based speech synthesis system, to build the corresponding synthetic voices. Information on the three parameters is extracted for each voice and statistical tests are conducted and compared to an independent subjective evaluation to assess the possible correlations between them. These results are presented in the paper.

The rest of this paper is organized as follows. Section II describes the Spanish speech database and introduces the methods of analysis of pitch, jitter and shimmer in vowels. Section III describes the results, which are discussed in Section IV. Concluding remarks and future work are presented in Section V.

2 Methods

Two Mexican speakers, a professional actress and actor, recorded three sets of 184 Spanish speech utterances each. The 184 utterances included isolated words as well as sentences which could be in affirmative or interrogative forms. The distribution is shown in Table 1.

Table 1: Spanish Corpus Contents.

Identifier	Corpus contents
1-100	Affirmative
101-134	Interrogative
135-150	Paragraphs
151-160	Digits
161-184	Isolated words

The selection of the words, sentences and paragraphs were the same as that of [10], an emotional speech database originally developed by the Center for Language and Speech Technologies and Applications of the Polytechnic University of Catalonia for the purpose of emotional speech research. The Mexican Spanish recordings were carried out in a professional studio where the recording conditions were completely controlled.

Acoustic features were extracted from the speech signals using Praat [6]; the features selected were pitch, jitter and shimmer, which we shall briefly describe in the following subsections.

2.1 Pitch

Each utterance was segmented into the corresponding phonemes, and the maximum pitch of each of the five Spanish vowels found was extracted using the autocorrelation method. The results were separated according to the vowel and the training conditions.

2.2 Jitter

Jitter is a measure of period-to-period fluctuations in the fundamental frequency. In general it is calculated between consecutive periods of voiced speech as follows:

$$J_t = \frac{|T_i - T_{i+1}|}{\frac{1}{N}\sum_{i=1}^{N} T_i} \tag{1}$$

where T_i, T_{i+1} are the present and posterior periods of speech and N the total number of intervals. The jitter reported is the local jitter, which is used as a voice quality feature, and is defined as the rate between the computed jitter and the mean value of the periods of voiced signal.

2.3 Shimmer

Shimmer is a measure of the period-to-period variability of the amplitude value and is defined as follows:

$$Shm = \frac{|A_i - A_{i+1}|}{\frac{1}{N}\sum_{i=1}^{N} A_i} \tag{2}$$

where A_i, A_{i+1} are the present and posterior periods' amplitude of speech, and N the total number of voiced periods. The shimmer reported is the local shimmer, which is defined as the average absolute difference between the amplitudes of consecutive periods divided by the average amplitude.

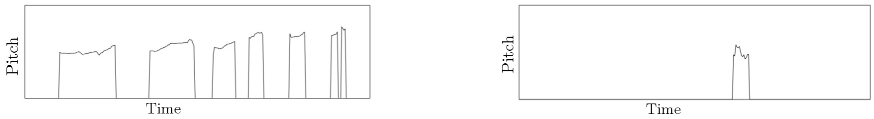
2.4 Training conditions

In order to measure the effects of different training conditions on pitch, jitter and shimmer of a synthetic voice, we used the procedures described below, for both speakers, to produce different voices constructed with the HTS system [20].

Normal training The male and female voices were trained using the full speech corpus with complete context information (24 features) adapted from HTS-CMU-US-ARCTIC demo in [7]. An analysis of the pitch range was made with Praat, so that a correct $f0$ range could be defined in both cases, and the 24 contextual factors were used, considering prosodic features.

Context information reduction In HMM-based speech synthesis, contextual factors are used to capture both segmental and prosodic features [**?**]. In this case, the prosodic contextual factores were removed, and only the phoneme definitions remained. The $f0$ range was the same as in normal training above.

Distorted $f0$ From our experiences with HMM synthesis, we found that the definition of the $f0$ range in training has a decisive influence on the results. A poorly defined range produces often intelligible but very unnatural voices. The reason for this can be seen in Figure 2, where an $f0$ contour is compared for a phrase pronouncing the hour, for both normal training and also with a poor $f0$ range. The lack of some pitch regions in the latter case has a considerable effect on its naturalness.



(a) Normal training

(b) Bad $f0$ range in training

Fig. 2: Pitch contours of the utterance "Son las 8:45" (It's 8:45).

3 Results

In order to compare the parameters of pitch, jitter and shimmer of the speakers and the HTS synthesized voices, an analysis of these parameters on all the database utterances was performed.

100 utterances were also produced using each of the synthesized voices, and 5 were randomly chosen for each voice. A mean opionion score (MOS) test was applied to these 5 utterances using 20 volunteers. We use these subjective evaluations as a reference to compare the possibly significant differences between the three acoustic parameters of the original voice with the synthesized ones.

Both the male and female voices obtained using the normal training conditions received the best subjective evaluation, while the voices obtained with a reduced $f0$ range scored the lowest.

Figure 3 shows the pitch value boxplots for each vowel obtained from the voices constructed using the different training conditions and the original natural male voice, while Figure 4 presents those of the female voice.

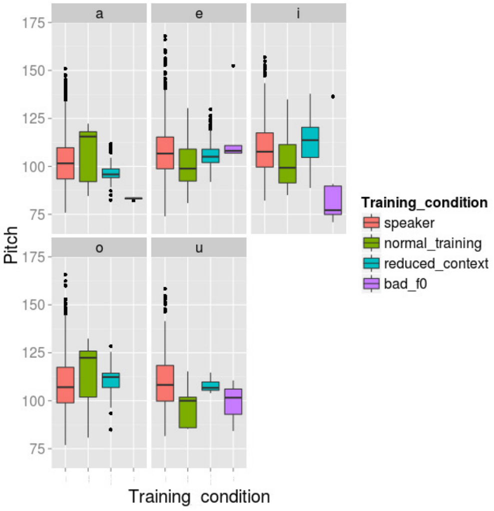


Fig. 3: Boxplots for the pitch values of each vowel using male voice.

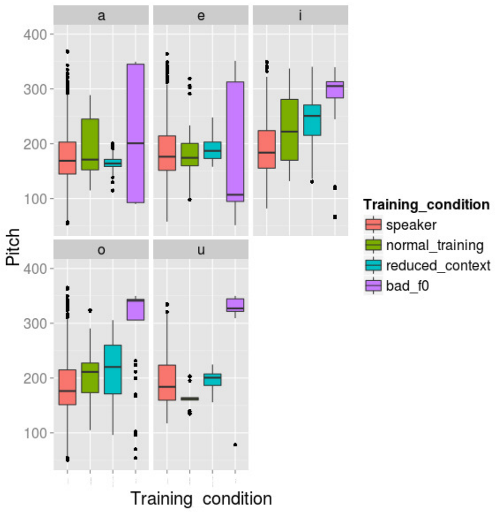


Fig. 4: Boxplots for the pitch values of each vowel using female voice.

Figure 5 shows the jitter value boxplots for the original database and the different training conditions, for the male speaker, and Figure 6 the corresponding shimmer value boxplots for the female speaker.

In order to decide which of the observed differences in the three parameters were statistically significant from the original speaker, we conducted a Friedman test, with a significance level of $\alpha = 0.05$.

Statistically significant differences were detected for several groups of variables, compared to the original speaker, as shown in Table 2. Friedman's test was carried out for all the training conditions, and a Post-hoc test was used to decide which groups are significantly different from each other, with special interest in differences related to the original speaker and with the training conditions.

Table 2: Friedman test Post-Hoc statistically significant differences with original speaker. M: Male voice, F:Female voice

Training conditions	Statistical significant difference		
	Pitch	Jitter	Shimmer
Normal training (M)		✓	
Normal training (F)	✓		✓
Distorted $f0$ (M)	✓	✓	✓
Distorted $f0$ (F)	✓	✓	✓
Reduced context (M)		✓	✓
Reduced context (F)	✓	✓	✓

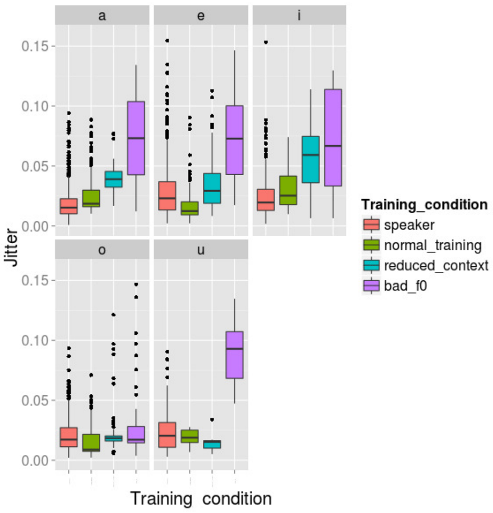


Fig. 5: Boxplots for the jitter values of each vowel using male voice.

Table 3 shows the result of the MOS test, considering the naturalness and intelligibility in pronouncing the time. The scale was from 1 (completely unnatural or completely unintelligible) to 5 (completely natural or completely intelligible).

Table 3: MOS of synthesized voices. M: Male voice, F:Female voice.

Training conditions	MOS Test	
	Naturalness	Intelligibility
Normal training (M)	2.96	3.61
Normal training (F)	1.96	2.52
Distorted $f0$ (M)	2.43	3.18
Distorted $f0$ (F)	1.82	2.79
Reduced context (M)	2.74	3.32
Reduced context (F)	2.60	3.13

4 Discussion

We find that the synthesized voices with a lower subjective value of naturalness and intelligibility have significant differences compared to the original speaker in the case of pitch and shimmer.

Table 2 shows that the voices with less natural values (i.e.those obtained with a poor definition of the f0 range in training), have statistically significant

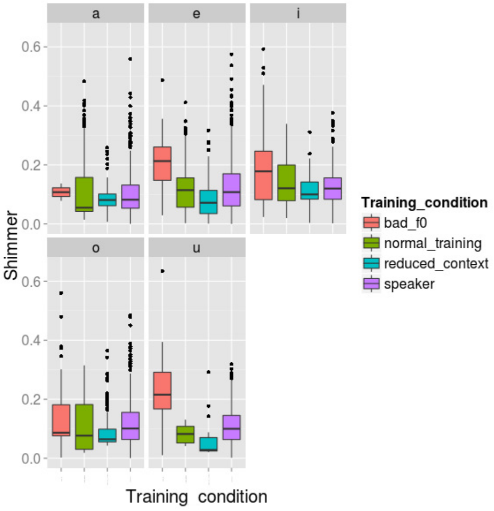


Fig. 6: Boxplots for the shimmer values of each vowel using female voice.

differences with the original speaker in the three acoustic parameters. Similar characteristics are found with the female voice trained with a reduced context, which gets lower subjective values than the voice with normal training. That voice, specially in the case of the male speaker, receives the best subjective evaluation in naturalness and intelligibility.

A combination of the three acoustic parameters can be related to the quality of synthesized voices. For example, Figure 7 shows the scatterplot of the six synthetic voices, where we see that the voices with significant difference compared with the original speaker are differentiated. The voices that have been evaluated with lower scores in the MOS test have these differences with the database.

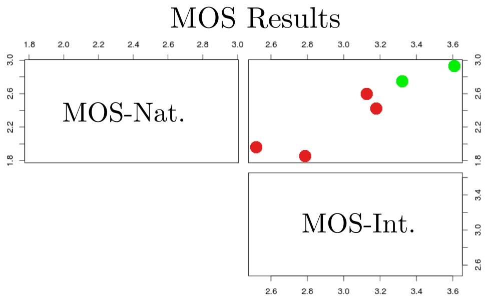


Fig. 7: Scatterplot for MOS test of the six synthesized voices.

5 Conclusions

An analysis of the acoustic parameters of pitch, jitter and shimmer for two speakers, and the voices synthesized from them using HMM-based synthesis was presented. Six synthetic voices were produced using different training conditions, and a MOS test applied to obtain different subjective values of naturalness and intelligibility for them.

The aim of the analysis was to establish a relationship between these acoustic parameters and the MOS results.

The results shows a relationship between the detection of statistically significant differences from a Friedman test and with the lowest quality of the synthetic voices.

These results may lead to establishing the statistical analysis of pitch, jitter and shimmer as a useful reference to determining synthetic speech quality, as related to subjective evaluations.

It is necessary to extend the experiments to other voices, ideally with larger speech databases, to allow a more extensive analysis based on training experiments of individual parameters, their pitch, jitter and shimmer and the corresponding subjective evaluations.

Acknowledgements. This work was supported by the SEP and CONACyT under the Program SEP-CONACyT, CB-2012-01, No.182432, in Mexico, as well as the University of Costa Rica in Costa Rica. We also want to thank ELRA for supplying the original Emotional speech synthesis database.

References

1. Black, A.W., Zen, H., Tokuda, K.: Statistical parametric speech synthesis. In: Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on. vol. 4, pp. IV–1229. IEEE (2007)
2. Brockmann, M., Drinnan, M.J., Storck, C., Carding, P.N.: Reliable jitter and shimmer measurements in voice clinics: the relevance of vowel, gender, vocal intensity, and fundamental frequency effects in a typical clinical task. Journal of Voice 25(1), 44–53 (2011)
3. Chu, M., Peng, H.: An objective measure for estimating mos of synthesized speech. In: INTERSPEECH. pp. 2087–2090 (2001)
4. Falcone, M., Yadav, N., Poellabauer, C., Flynn, P.: Using isolated vowel sounds for classification of mild traumatic brain injury. In: Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. pp. 7577–7581. IEEE (2013)
5. Fant, G.: Acoustic theory of speech production: with calculations based on X-ray studies of Russian articulations, vol. 2. Walter de Gruyter (1971)
6. Goldman, J.P.: Easyalign: An automatic phonetic alignment tool under praat. In: INTERSPEECH. pp. 3233–3236 (2011)
7. HTS-Group: Hts voice demos. `http://hts.sp.nitech.ac.jp/?VoiceDemos` (September 2014)

8. Huang, D.Y.: Prediction of perceived sound quality of synthetic speech. Proc. APSIPA (2011)
9. Huang, D.Y.: Prediction of perceived sound quality of synthetic speech. Proc. APSIPA (2011)
10. Maegaard, B., Choukri, K., Calzolari, N., Odijk, J.: Elra–european language resources association-background, recent developments and future perspectives. Language Resources and Evaluation 39(1), 9–23 (2005)
11. Martínez-Licona, F., Goddard, J., Martínez-Licona, A., Coto-Jiménez, M.: Assessing stress in mexican spanish from emotion speech signals. In: Proc. 8th International Workshop on Models and Analysis of Vocal Emissions for Biomedical Applications, MAVEBA. pp. 239–242 (2013)
12. Möller, S., Hinterleitner, F., Falk, T.H., Polzehl, T.: Comparison of approaches for instrumentally predicting the quality of text-to-speech systems. In: INTERSPEECH. pp. 1325–1328 (2010)
13. Norrenbrock, C.R., Hinterleitner, F., Heute, U., Moller, S.: Instrumental assessment of prosodic quality for text-to-speech signals. Signal Processing Letters, IEEE 19(5), 255–258 (2012)
14. Remes, U., Karhila, R., Kurimo, M.: Objective evaluation measures for speaker-adaptive hmm-tts systems. Proc. SSW, to appear (2013)
15. Tokuda, K., Nankaku, Y., Toda, T., Zen, H., Yamagishi, J., Oura, K.: Speech synthesis based on hidden markov models. Proceedings of the IEEE 101(5), 1234–1252 (2013)
16. Tokuda, K., Yoshimura, T., Masuko, T., Kobayashi, T., Kitamura, T.: Speech parameter generation algorithms for hmm-based speech synthesis. In: Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on. vol. 3, pp. 1315–1318. IEEE (2000)
17. Valentini-Botinhao, C., Yamagishi, J., King, S.: Can objective measures predict the intelligibility of modified hmm-based synthetic speech in noise? In: INTERSPEECH. pp. 1837–1840 (2011)
18. Valentini-Botinhao, C., Yamagishi, J., King, S.: Evaluation of objective measures for intelligibility prediction of hmm-based synthetic speech in noise. In: Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on. pp. 5112–5115. IEEE (2011)
19. Wertzner, H.F., Schreiber, S., Amaro, L.: Analysis of fundamental frequency, jitter, shimmer and vocal intensity in children with phonological disorders. Revista Brasileira de Otorrinolaringologia 71(5), 582–588 (2005)
20. Zen, H., Nose, T., Yamagishi, J., Sako, S., Masuko, T., Black, A.W., Tokuda, K.: The hmm-based speech synthesis system (hts) version 2.0. In: SSW. pp. 294–299. Citeseer (2007)

Bringing Networks Together to Improve Advertising Performance

Luis Miralles Pechuán[1,2] and José M. García Carrasco[1]

[1] Departamento de Ingeniería y Tecnología de Computadores, University of Murcia,
Murcia, Spain

[2] Facultad de Ingeniería, Universidad Panamericana,
Mexico

lmiralles@up.edu.mx, jmgarcia@ditec.um.es

Abstract. We believe that small networks working together can create a more competitive solution against bigger networks, not only regarding ad performance but also fraud detection. Moreover, we have designed algorithms to uniformly distribute visits over several networks, and we have used the average deviation as a parameter to compare results.

Keywords: Internet marketing, click fraud, ad performance, networks working together, fraud detecting algorithm.

1 Introduction

Internet is one of the most revolutionary inventions in the history of humanity. It evolved from a US Department of Defence project known as Arpanet, which was developed back in 1969. Since then it has allowed us to share photos on Facebook, send email with Gmail, make video calls via Skype, Blog on Wordpress, stream videos on YouTube, sell on eBay or pay online using PayPal. We consider them commonplace in today's world, but barely a few decades ago were they absolutely unthinkable.

It offers endless opportunities to those who use it, such as being able to work from anywhere at any time of day, instantly send information or access resources about anything. Logically, the number of people wishing to enjoy the benefits is constantly growing. There has also been a widespread proliferation of companies offering of huge variety of services and the best way for these companies to blossom is by using online ad campaigns.

The first ad banner ever seen on a webpage was for AT&T in 1994, and from then on its use has continued to grow immensely. In the third quarter of 2013, investments in online advertising reached $10.69 billion dollars [1]. Online advertising offers huge advantages to advertisers as it allows them to modify campaigns at any time. Whilst most channels contract closed packets, online advertising allows campaigns to be cancelled should they not show good results, or areas of high sales can be focused on.

Using the web we can check campaign quality in real time, with parameters like number of products sold or the average time users spend on our page. This kind of publicity allows us to select a segmented public with usable-to-program parameters such as age, gender, geographic zone, likes and much more, thanks to networks holding a huge amount of user data.

The cost of such campaigns can be adapted to any budget, as we can select the number of ads to be shown. Finally, we can establish a bidirectional user channel giving us immediate feedback. Such users act by forwarding messages, bookmarking the page or recommending it to a friend.

2 The Problems with Being Small

The main objectives of any advertising platform are to show users the most relevant ads and reduce the number of faults in fraudulent clicks to zero. The largest ad platforms are at an advantage in respect to the smaller networks given that they have more secure fraud detection systems. This allows them to get more advertisers and publishers, in turn creating higher revenues creating a vicious circle making the small networks even smaller and themselves even bigger.

2.1 Ad Performance

As advertisers make more and more specific campaigns, the number of pages they can be on reduces, but at the same time they are more effective given that advertisers are paying a higher price. This is known as targeting [3]. In order to develop a good targeting campaign, we must filter out a series of parameters such access keywords, age, gender, income level, location and likes from user profiles. Another series of attributes, although not as influential must still be taken into account. They include browser, search engine, operating system or device being used.

When a page is visited by a user fitting the desired characteristics an ad is shown and if it results in a click, the advertiser is charged accordingly. A large platform with a lot of publishers can easily find any page related to and be accepted by an advertisers requirements. On the other hand, if the ad network has very few publishers and works independently, ads will receive much less coverage [1,] or rather they will not be shown as much and will hence be seen by less users. To solve this issue, generic campaigns are created with the disadvantage of lower performance.

2.2 Fraud Detection

First, we have to emphasize that fraud really is a threat. According to experts, 15% of all clicks are fraudulent and out of that 20% go undetected [4]. This means that we

[1] Coverage has a value of 0 & 100 and represents the number of times an ad is shown to a user. Having 50% coverage means half the visits have not created revenue, due to them not meeting any advertisers' requirements.

can discount that 3% (0.15 x 0.2) from what is paid by advertisers. As stated by Tuzhilin, it is statistically possible for some advertisers to be unsatisfied, but if the rest are happy the platform will be successful [5].

The problem arises when the number of fraudulent clicks increases, or the ability to detect competition clicks decreases. Publishers prefer working with the best performing platform for their ad space, and advertisers look to improve campaign results [6].

Large ad networks make millionaire investments and tend to use specialized equipment to continuously improve their fraud detection systems. Their false click detection system is also much more superior to those used by small networks. Google for example knows the CTR[2] of every class of webpage, so should a page have different statistics to the rest, it can be easily detected.

Following Kirchhoff's principal [3], the major platforms should publish the techniques used by scammers and the methods used to detect them, enabling systems to be more secure. Furthermore, there has been research that talks of the convenience of networks working together to improve fraud detection [7]. However, large networks are trusted precisely for their click detecting capability so if all platforms were equally secure, the competitive advantage would disappear.

3 Small Networks Working Together

Some authors affirm that the exchange of ads represent the future of online advertising and the solution for small ad platforms however for such exchanges to be successful firstly the issue of click fraud and the legal questions regarding user privacy need resolving; and an exchange model, that generates benefits for all parties involved, needs developing.

3.1 Working Together to Improve Performance

Advertising exchanges consist of platforms exchanging visits not meeting the requirements of any of their advertisers, or they are simply looking for an advertiser willing to pay more. In this model, advertisers pay for space only if certain requirements are met, and editors leaves a space on their page to be filled by the most profitable ad. Let's imagine there are two small ad platforms, SpainOnline97 and BrazilMarket43. Most of SpainOnline97's advertisers would be Spanish speakers and most of BrazilMarket43's would be Portuguese speakers. If these two networks were

[2] CTR is the number of clicks received by an ad divided by the number of times it has been seen, e.g. if 15 clicks have been received and it has been seen 1000 times, the CTR will be 1.5%.

[3] The success of a cryptographic algorithm should not remain a secret. Any algorithm employed using cryptography is published, and should the system become susceptible to an efficient attack, it automatically improves or stops. This policy has allowed systems to be ever more secure, and is now almost invulnerable.

to exchange ads, a user from Spain visiting a BrazilMarket43 page could be shown ads from Spain and are much more likely to be interested in buying the product and vice versa.

Most platforms follow IAB standards, making exchanges easier. This can seem simple when there are only two networks working together, but when hundreds of networks with thousands of advertisers there are certain factors to taken into account, such as: volume, revenues, fraud committed by advertisers or adequately distributed visits.

In order to make such exchanges possible, we need to develop an algorithm, taking into account the fact the ad is to be shown in split seconds and at the same time be really effective, so it is recommended using parallelization.

3.2 Working Together against Fraud

A cost reducing solution could be for ad platforms to outsource click detection to specialists, although the problem is that these specialists could be tempted to create their own threats to ensure work or ally with cheating publishers to increase revenue.

Advertising platforms face many threats such as Click-bots, illegal traffic or users with bad intentions nonetheless [7], these are no different to each other where a platform is concerned, to when a platform detects a malicious IP and warns the rest, the threat is taken care of [8].

Information sharing is an advantage so that all platforms can offer a better service to advertisers, as well as reducing the number of undetected fraudulent clicks. Such advantages include:

- Awareness of page CTRs from other platforms, so should a user have a page with similar characteristics but a distinct CTR, it will be suspicious.
- Sharing of suspect IPs.
- Updating proxy list [4] to invalidate clicks originating from them.
- Share new click-bot [9] detecting methods.
- Comparing ratios from a specific editor with those from editors of other platforms. This tells us if it differs from the average.
- Calculating percentage of fraudulent clicks in order to apply discounts to advertisers.

3.3 Privacy

Advertising platforms recollect user information when services are used. Both Google Analytics and Webmaster tools allow Google to access many different statistics including how long a user remains on a certain page, number of average pages visited

[4] A proxy is a program or device that connects to the internet from another computer. It is used to maintain anonymity, or better security. In the case of click fraud, it allows clicks to be made without the IP being detected.

and much more. The more user information they have, if used effectively, the more personalised ads shown can be, but privacy must always be strictly respected.

In order to protect themselves from being reported regarding privacy, platforms oblige users to accept these services in their terms and conditions. To guarantee the right to privacy [10] and at the same time segment publicity, platforms create profiles where such information is saved. Theoretically, the profiles are not associated with any particular person, but instead work anonymously. It has been known for these profiles to be tracked by security organisations in the detection of possible terrorist threats and paedophiles.

4 Algorithms to Improve Advertising Management Performance

4.1 Networks Working Together to Increase Ad Coverage

Here we aim to show how to improve coverage; this means the higher the percentage of satisfied visits, the higher the number of collaborating networks. To make this test possible we obtained a total of 104,151 real visits from the site history of buscadoreseninternet.net from 01/06/12 to 01/01/13. Each visit uses a series of fields as seen in Table 1.Visits were exported into an excel table from Google Analytics so they can be seen clearly as shown in Table 2.

To show how to improve we have compared each visit on the table with ad campaign requirements given by X participating networks (where X = 1, 2, 3, 5, 10, 25, 50, 100). Each network has 10 campaigns, giving a total of 1000 campaigns simultaneously, giving X a value of 100.

The values selecting each advertiser for each parameters are randomly established based on probability of occurrence, meaning that should 90% of operating systems in visit history be running Windows, the probability of the advertiser choosing the Windows value as the OS parameter will be 90%. As shown in the table, advertisers select where to show their ads from a series of fields, these could include country, city or page category with the ad platform adding ads to relevant webpages using an algorithm.

For each of the options shown in Table 5 different parameters are configured. The number of parameters to be configured depends on the size of the biggest option, making campaigns more specific as well as more difficult to cover.

In Table 4 the Y axis represents the number of options, and the X axis represents configurable parameters, which were explained earlier in Table 1.

In Table 5 the X axis shows coverage related to the number of networks working together and the Y axis show options to be selected by advertisers upon making campaigns. Parameters to be configured can be seen above in Table 4.

As can be seen in the table, when more networks work together coverage is greatly improved.

4.2 Distributing Visits

Apart from trying to improve coverage by contacting other ad platforms, ads must be distributed as fairly as possible. To do this, three algorithms have been developed and for quality assurance a method of average deviation was developed, the lower the average deviation the better the algorithm to be used will be.

Table 1. User visit parameters.

1	Time	This refers to the time of day of the visit. It ranges from 0 to 24 E.g. 2,4,6 etc
2	Browser	This refers to the user's browser e.g. Internet Explorer, Google Chrome, Mozilla Firefox
3	Browser version	Indicating the browser version being used e.g. in Internet Explorer you can see version 9.0, 8.0 & 7.0 etc.
4	Operating System	This refers to the OS of the computer the webpage is being accessed from. The most common being Windows, but Mac OS X and Linux are also used.
5	OS Version	This refers to OS version e.g. Windows 7, 8 or Mac OS X Lion etc.
6	Flash version	Many browsers have flash preinstalled in order to open certain pages, of which there are many versions e.g. 11.3 r31, la 10.0 r32 or 10.2 r153.
7	Has flash?	This parameter indicates if the browser has flash. The value is YES or NO.
8	Screen bitrate	This refers to number of bits required to show a pixel. The most common is 24-bit or 32-bit.
9	Screen resolution	This is the number of pixels the monitor has, it is usually (shown in width x length) around 1280x1024 or 1024x768.
10	Country	Using the IP we can determine the country of the visit.
11	City	As well as country we can also determine the city the visit is coming from.
12	Language	Language being used on user's system e.g. es-419, es, es-mx.
13	Network address	This refers to the ISP url the user is visiting from e.g. megared.net.mx, cablevision.net.mx, prod-infinitum.com.mx,cableonline.com.mx, maxcom.net.mx.
14	Network name	This refers to the name of the network being used by the user e.g. uninetredirection management, uninet s.a. de c.v…
15	Access page	This is the page where our visit originates from, usually a search engine such asyahoo.com, but it could also be being accessed directly or through a link.
16	Visit type	Visit type could be organic if a search engine is used, or referral is a reference is used.

Table 2. Storing user visits.

	Field 1	Field 2	Field 3	Field …	Field N-1	Field N
Visit 1	Firefox	16.0	Windows	7	11.4 r402	24-bit
Visit 2	Chrome	22.0.1229.92	Windows	XP	11.4 r31	32-bit
Visit …	I. Explorer	8.0	Windows	7	(not set)	32-bit
Visit N-1	I. Explorer	8.0	Windows	7	11.1 r102	32-bit
Visit N	Chrome	21.0.1180.89	Windows	XP	11.3 r31	32-bit

Table 3. Storing user visits.

	Field 1	Field 2	Field 3	Field …	Field N-1	Field N
Advertiser 1	Chrome, Firefox	16.0	Windows	XP,7	11.4 r402	24-bit, 32-bit
Advertiser 2	Chrome	22.0.1229.92	Windows	XP	11.4 r31	32-bit
Advertiser N	Internet Explorer	8.0	Windows	XP,7	(not set)	32-bit

Table 4. Parameters selected for each option.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Option 1										X						
Option 2		X								X						
Option 3		X		X						X						
Option 4		X		X						X		X			X	
Option 5		X		X						X	X	X			X	
Option 6		X		X				X		X	X	X			X	
Option 7		X		X				X	X	X	X	X			X	
Option 8	X	X		X				X	X	X	X	X			X	
Option 9	X	X		X	X			X	X	X	X	X			X	
Option 10	X	X	X	X	X			X	X	X	X	X			X	
Option 11	X	X	X	X	X			X	X	X	X	X			X	X
Option 12	X	X	X	X	X			X	X	X	X	X	X	X	X	X
Option 13	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table 5. Advertising coverage in relation to number of networks and options.

Ad Coverage	1	2	3	5	10	25	50	100
Option 1	0,65901	0,79542	0,85574	0,9092	0,95288	0,98307	0,99184	0,9943
Option 2	0,35315	0,51562	0,63524	0,73938	0,8384	0,92112	0,95214	0,97189
Option 3	0,34136	0,5101	0,60923	0,71234	0,82205	0,8987	0,93521	0,95982
Option 4	0,15334	0,25803	0,33614	0,43064	0,53901	0,65418	0,73267	0,79566
Option 5	0,02476	0,04473	0,06103	0,09263	0,15475	0,25647	0,3468	0,44492
Option 6	0,01351	0,02455	0,03379	0,05396	0,09425	0,17517	0,25696	0,34725
Option 7	0,00169	0,00344	0,00549	0,00882	0,01675	0,03739	0,06599	0,10764
Option 8	9,4E-05	0,00019	0,0003	0,0005	0,001	0,00249	0,00487	0,00955
Option 9	6,9E-05	0,00015	0,00023	0,00036	0,00071	0,00176	0,00343	0,00663
Option 10	1,9E-05	3,5E-05	5,1E-05	8,3E-05	0,00015	0,00036	0,00073	0,00146
Option 11	1,2E-05	2,5E-05	4,3E-05	7,4E-05	0,00014	0,00038	0,00074	0,00147
Option 12	3E-06	7E-06	9E-06	1,4E-05	3,1E-05	7,6E-05	0,00016	0,0003
Option 13	1E-06	2E-06	2E-06	4E-06	0,00001	2,6E-05	5,1E-05	0,0001

Table 6. Average deviation from the simple algorithm.

Simple	2	3	4	5	10	25	50	100
Option 1	25317,5	25074,4	22688,6	20387,7	13193,3	6422,08	3512,67	1862,84
Option 2	8876,84	10684,2	10747,8	10379,5	8419,45	4901,6	2913,74	1634,75
Option 3	10042,8	10521,1	10715,2	10377,8	8224,26	4796,97	2852,07	1603,25
Option 4	4939,88	5217,37	5425,4	5236,4	4307,7	2754,35	1763,37	1064,59
Option 5	620,12	776,33	818,67	819,45	816,09	679,08	525,36	374,93
Option 6	396,52	475,69	499,97	511,62	497,21	428,71	350,36	268,62
Option 7	55,81	67,19	85,37	90,59	92,42	90,14	86,49	74,27
Option 8	5,14	5,72	6,04	5,95	6,63	6,81	6,76	6,92
Option 9	3,44	3,68	4,15	4,02	4,77	5,09	5,18	5,15
Option 10	1,26	1,4	1,41	1,4	1,45	1,52	1,56	1,6
Option 11	1,11	1,34	1,44	1,52	1,54	1,59	1,59	1,55
Option 12	0,24	0,28	0,33	0,34	0,42	0,48	0,47	0,47
Option 13	0,09	0,14	0,17	0,19	0,21	0,2	0,2	0,19
							Total	276,473

The average deviation is the average of the absolute values of the deviations from the mean and is shown as Dm.

$$Dm = \frac{1}{n}\sum_{i=1}^{n}|X_i - X| \quad (1)$$

The Simple algorithm first contacts the number one network and in the case it cannot satisfy the request, number 2 will be contacted and so forth until the last network is reached. Table 6 shows the results obtained.

The Round Robin algorithm first contacts the number 1 network in the first cycle, but the second time it moves on to contacting network number 2. Whenever a visit distributed it starts contacting the following network from the last time it was run. The results are shown in Table 7.

Table 7. Average deviation of Round Robin algorithm.

Round	**2**	**3**	**4**	**5**	**10**	**25**	**50**	**100**
Option 1	4747,32	4666,87	3856,47	3486,34	1925,38	798,02	409,9	207,55
Option 2	2981,15	3936,11	3509,33	3172,09	1916,85	890,94	461,24	233,56
Option 3	3951,06	3976,8	3721,39	3456,14	2009,65	900,78	451,18	230,77
Option 4	2646,23	2643,42	2668,82	2349,33	1599,45	774,59	425,82	224,63
Option 5	634,12	692,18	659,68	729,41	659,47	445,21	277,43	160,23
Option 6	390,75	433,59	453,86	456,33	454,81	316,1	221,94	137,84
Option 7	70,19	78,44	87,07	92,67	95,09	90,39	77,25	61,37
Option 8	4,37	5,32	5,51	5,54	6,44	7,01	7,05	6,82
Option 9	3,19	3,71	4,25	4,8	4,85	5,15	5,21	5,21
Option 10	0,99	1,21	1,28	1,28	1,42	1,55	1,59	1,59
Option 11	1,24	1,23	1,37	1,41	1,46	1,48	1,5	1,49
Option 12	0,31	0,39	0,36	0,37	0,4	0,45	0,44	0,45
Option 13	0,08	0,14	0,16	0,19	0,23	0,2	0,2	0,2
							Total	77,116

The Minimum algorithm always contacts the network with the least satisfied visits. To do this it requires the help of a table showing the number of visits distributed per network. The results are shown in Table 8.

Table 8. Average deviation of Minimum algorithm.

Minimum	**2**	**3**	**4**	**5**	**10**	**25**	**50**	**100**
Option 1	255,76	56,35	22,82	15,74	4,24	0,78	0,43	0,37
Option 2	1290,72	539,18	287,88	76,91	16,09	1,75	0,61	0,41
Option 3	679,6	466,18	178,16	96,93	12,12	2,69	0,88	0,43
Option 4	1127,77	815,04	630,44	387,4	153,21	16,58	2,85	0,95
Option 5	634,4	692,12	647,37	625,64	493,62	208,01	74,87	15,33

Minimum	2	3	4	5	10	25	50	100
Option 6	376,96	442,27	432,25	425,59	344,21	202,55	97,03	32,26
Option 7	62,33	77,08	81,23	86,55	91,51	81,2	66,91	47,92
Option 8	5,13	5,68	5,8	5,96	6,42	6,82	6,75	6,65
Option 9	3,15	4	4,7	4,51	5,1	5,15	5,19	4,93
Option 10	1,09	1,54	1,51	1,51	1,47	1,55	1,52	1,51
Option 11	1,17	1,44	1,56	1,53	1,54	1,58	1,61	1,62
Option 12	0,37	0,4	0,44	0,41	0,43	0,49	0,47	0,47
Option 13	0,1	0,14	0,16	0,17	0,2	0,2	0,2	0,2
							Total	**13,595**

To compare results we have summed up all the tests from each algorithm. Using the lowest sum from each one. The best results were gained by the Minimum (13595.04), followed by Round Robin (77115.58) and finally the Simple (276473.42).

4.3 A Fraud Detecting Algorithm

To test the improvements to fraud detection in a collaborative environment, the captcha technique [11] was used along with irrelevant ads [12]. This helps us detect fraudulent IPs. The captcha technique requires asking users to solve a "captcha", when access to ad content is desired. If captchas were to be put on all ads, users would become frustrated so they are only applied to about 20% of ads. The irrelevant ad technique shows a determined user ads unrelated to their profile, meaning that clicks do not come from user interest but rather by malicious means. The user is not expected to click such ads, so there is a high probability that any clicks being made are from botnets or a group of poorly trained, fraudulent users.

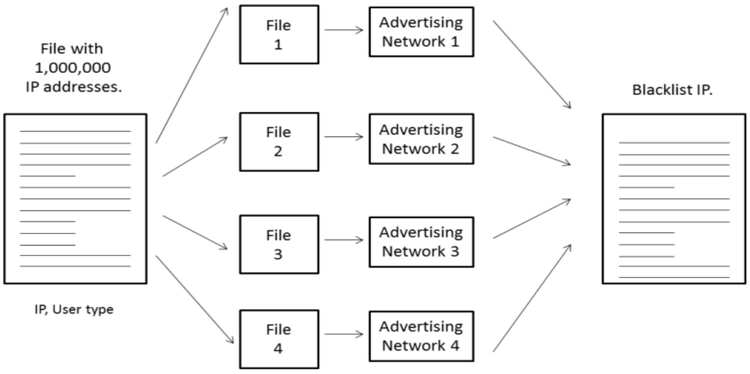


Fig. 1. Network collaboration model for fraud reduction.

If we abuse these, the fraudulent user will feel watched, and will realize that something is not right, provoking them to change techniques. To show the improvement to fraud detection techniques in proportion to collaborating networks, a model where networks exchanged high risk IPs was designed (see Figure 1).

The experiment consisted of creating a catalogue of 100,000 IPs with 10% of them coming from irrelevant ads; of this 10%, 75% come from botnets, 10% fraudulent users and 5% valid users. Of the remaining 90% of visits 80% come from valid users, 15% from botnets and 5% from fraudulent users. 1000 networks took part in the experiment, from which each received 2000 random visits from the original catalogue. To measure fraud detection performance, a check is made using captchas 20% of the time, except in the case of irrelevant ads where checks are always made using captchas. Botnets are unable to able to solve captchas so we will add the IP to a list of suspicious IPs. The detection percentage consists of dividing the number of detected botnets by the numbers of total botnets. Just one network was involved in the first experiment so that there is an empty suspicious IP list. As more networks participated, the number of suspicious IPs on the list increased, so that the 500th network has the fraudulent IPs detected by the previous 499. This explains that as you increase the number of networks their ability to detect fraudulent IPs is higher.

Figure 2 shows the improvements to fraud detection methods against the number of networks working together and number of visits where captchas were applied. The X axis shows the number of networks working together in fraud detection and the Y axis shows the percentage of fraudulent IPs detected.

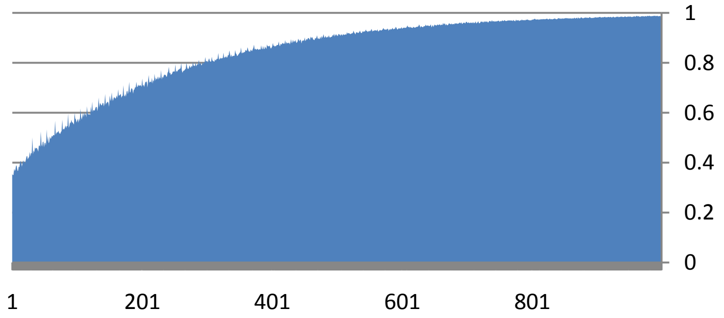


Fig. 2. Percentage of fraudulent IPs detected against number of networks working together in the fight against fraud.

5 Conclusions

The number of internet users has been constantly increasing since the creation of Arpanet back in 1969, the total number of users reached about 2.9 billion in 2014. In developing an online business it is essential to attract visitors, and to do that the most practical way is through online advertising campaigns. The most frequently used payment method is CPC (Cost per Click), where editors pay per every click made by users.

The amount of money circulating online has caught the attention of fraudsters for varying reasons. Such fraud arises from the fact it is relatively easy to commit given that victim and attacker are usually in different countries, and evidence is easy to manipulate.

Some examples of infractions committed in online advertising include click inflation, competition clicks, farmed clicks or the famous click bots. The smaller platforms have two difficult problems to solve, causing them to be less competitive when facing the large platforms, and hence are in danger of disappearing. These problems are fighting fraud and improving advertising performance.

On top of that small platforms lack the financial resources to develop technology to distinguish legitimate clicks from false clicks, by either botnets or humans. On the other hand, advertisers are increasingly focusing on micro-targeting which consists of small groups with similar interests. As publishers have few small networks they are quite often unable to meet the requirements of advertisers using such segmented campaigns.

In this article we have described a collaborative model designed to improve small network performance results as well as increasing their ability to detect fraud is designed. It has been demonstrated that the greater the number of networks cooperating the higher the number of adverts that can be covered.

To ensure adverts are shared fairly amongst networks, so everyone gets an equal gain three algorithms have been used: Simple, Round Robin and Minimum visits. Proving the minimum visits algorithm is the best of the three.

To improve fraud detection we have designed a collaborative environment in which each of the networks informs the rest whenever the IP of a determined click-bot or malicious user is detected, showing that detection is significantly improved when networks work together using captcha and irrelevant advertising techniques.

An interesting line of research and one which could be looked upon further is the optimization of campaign performance. The fact of optimizing campaigns to inform advertisers about parameters allow higher revenues to be gained facilitates advertising campaigns, making it unnecessary to hire an expert to review and analyze results.

Since millions of advertisers can participate in advertising exchanges, it is vital to design an algorithm to find the most relevant advert for every single visitor. Such an algorithm should run in a few tenths of a second, so multiple threads running in parallel will have to be used.

References

1. Goldberg, L., Silber, S.: IAB internet advertising revenue report. Retrieved from http://www.iab.net/about_the_iab/recent_press_releases/press_release_archive/press_release/pr-122313 (2012)
2. Chen, M.: The Effect of Fraud Investigation Cost on Pay-Per-Click Advertising. College of New Jersey, Galloway (2011)
3. Moe, W. W.: Targeting Display Advertising. London, UK: Advanced Database Marketing: Innovative Methodologies & Applications for Managing Customer Relationships (2013)

4. Mills, E.: Click fraud could threaten pay-per-click model. Retrieved from http://www.news.com/Study-Click-fraud-could-threaten-pay-per-clickmodel/2100-1024_3-6090939.html (2006)
5. Tuzhilin, A.: The Lane’s Gifts v. Google Report. Mimeo, New York University. Retrieved from http://googleblog.blogspot.com/pdf/Tuzhilin_Report.pdf (2005)
6. Hui, W.: Estimating the Number of Genuine and Fraudulent Clicks in the Pay-Per-Click (PPC) Model. University of Nottingham Ningbo, China (2010)
7. Vidyasagar, N.: India’s secret army of online ad clickers. The Times of India. Retrieved from http://timesofindia.indiatimes.com/business/india-business/Indias-secret-army-of-online-ad-clicers/articleshow/654822.cms (2004)
8. Mungamuru, B. W.-M.: Should Ad Networks Bother Fighting Click Fraud? (Yes, They Should). (2008)
9. Goodin, D.: Botnet caught red handed stealing from Google. Retrieved from http://www.theregister.co.uk/2009/10/09/bahama_botnet_steals_from_google (2009)
10. Wikipedia: Right to privacy. Retrieved from http://en.wikipedia.org/wiki/Right_to_privacy (2014)
11. Chow, R. E.: Making CAPTCHAs Clickable (2008)
12. Haddadi, H.: Fighting Online Click-Fraud Using Bluff Ads. ACM SIGCOMM Computer Communication Review, v. 40, no. 2 (2010)

Hybrid Subjective Decision Support System based on Computational Semiotics and Computational Intelligence Techniques

Denis Martins and Fernando Buarque de Lima Neto

Computing Engineering Program, School of Engineering,
University of Pernambuco, Recife,
Brazil

{dmlm, fbln}@ecomp.poli.br

Abstract. Subjective decision problems involve personal feelings and opinions, adding substantial complexity to evaluate different candidate options. In order to deal with this kind of problem, in which individual experience is considered and impacts directly in decision making process, many computational methods have been applied. However, the traditional approaches are often not flexible to consider uncertainties, imprecise situation contexts and idiosyncrasies. In this sense, we present a Hybrid Subjective Decision Support System based on Computational Semiotics and Computational Intelligence techniques. Our approach relies on Case-based Reasoning as the problem solving main methodology and Self-organizing Maps, which acts as pattern recognition tool, in order to organize more appropriately retrieval of similar past cases. Furthermore, a semiotic model handles a prior knowledge (i.e. knowledge acquired from a specialist) and domain specific restrictions to guide the search process towards an appropriate problem solution. In the paper we illustrate how the proposed approach can deal graciously with subjective concepts providing a more intuitive and evident decision making.

Keywords: Subjective decision, case-based reasoning, self-organizing maps, semiotics, decision support system.

1 Introduction

In dynamic environments, decision-making is usually an arduous task, especially when subjective decisions are necessary. In such situations, individual experience or personal opinions have to be considered for selecting an appropriate alternative from a set of possible options (*e.g.* decide which car to buy or make a leisure travel plan). This could take a large amount of time, not only because the concept of appropriateness change frequently, but also because environmental changes (than may occur rapidly).

In this context, Computational intelligence (CI) aims to provide adaptive mechanisms inspired by nature, possibilitating the construction of computer systems with intelligent behavior for tackling with such complex and dynamic kind of problems. The use of intelligent techniques such as Artificial Neural Networks and Case-Based Reasoning affords, for example, to these systems the ability to learn and adapt to new situations[1][2]. Thus, we argue that adaptive methods combined with Semiotic concepts of sign deconstruction and consequent individualized reasoning may assist well in subjective decision-making process, providing the needed individualization of decisions while improving efficiency and precision along time (due to the adaptive nature of CI mechanisms).

2 Semiotics

Semiotics is a field of human science which studies signs [3]. It involves the investigation of cognitive process such as communication and interpretation, as well as the study of how meaning are made and represented [4]. Charles Sanders Peirce (1857-1913), defined sign as something that, in some way, means something to someone in some respect or capacity [5]. Peirce model of sign can be seen as a triadic structure, as shown in Fig. 1. The object consists in something that is referred by the sign, the *interpretant* is the effect caused by a sign in someone’s mind (*i.e.* meaning of the sign) and, finally, the sign or *representamen* is related to the form that sign takes.

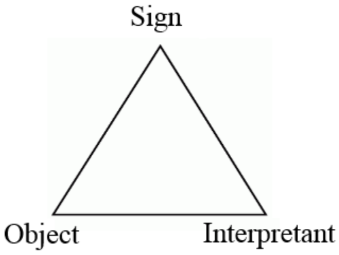


Fig. 1. Triadic structure of a sign as defined by Peirce.

Computational Semiotics is a relatively recent research field that aims to combine a computational approach with some important concepts grounded by Semiotics in order to develop a methodology to artificially create intelligent systems [3]. Because a sign can be deconstructed continuously and its constituents parts became grounded in the mind of the subject, a semiotic-based system is suitable for modeling dynamic processes in which, for example, decision parameters and options change frequently in relation to a particular individual [6].

In this context Computational Semiotics approaches can also apply CI methods, such as Artificial Neural Networks, Case-Based Reasoning, Evolutionary or Swarm computation, to provide the ability to learn and adapt themselves, this time, to problem contextual changes, new situations and, consequently, to idiosyncrasies as well [1]. Indeed, those methods are already been used in function optimization problems, time series prediction, adaptive control, classification tasks and clustering data [7].

Furthermore, a semiotic model handles a prior knowledge (i.e. knowledge acquired from a specialist) and domain specific restrictions to guide the search process towards an appropriate problem solution. As this type of information is, in general, subjective and imprecise, this system may provide a useful computational modeling framework which can encompass some problem-domain concepts that are not clear or contain some uncertainty [8].

The application of Semiotics concepts in intelligent systems can also bring advantages in problems that involve situation or context awareness [9]. As semiotic analysis intends primarily to understand situational dependent concepts, we argue that these analytical processes are also able to model objects or contextual information [3]. In this work, we argue that Semiotics can therefore provide a framework to recognize situation patterns, identifying relevant not obvious information and be helpful in the decision-making process.

3 Case-Based Reasoning

Differently from other Artificial Intelligence methods, Case-based Reasoning (CBR) is a problem solving technique which allows a system to reason and to learn from previous experienced (contextual) situations [10]. These experiences are represented as cases, which contain both problem description and its solution. Appling CBR, whenever a new problem appears, the system searches into a Case Base (CB) in order to retrieve the most similar past cases based on a similarity measure. Then, the retrieved cases are combined and adapted for building a solution that can be used for the new problem.

The Case Base is a collection of past cases used CBR cycle. In general, CB can contains both successful and non-successful cases (*i.e.* cases with a solution that did not respond well to a given problem situation). The first ones guide the search process towards an appropriate problem solution while failed experiences allow system to avoid some states that are unwanted, like occur in human mind reasoning process [2].

In CBR, the reasoning process is incremental, similar to natural human decision making, and can be typically summarized in four steps [11] below, also shown in Fig. 2:

1. Retrieve: given a new problem situation, recall the most similar cases;
2. Reuse: use or combine retrieved cases in order to solve the problem;
3. Revise: evaluate the proposed solution, by an expert or in a real world application, for example. If the solution is not appropriated, some adaptations are made in order to better fit the current problem or to satisfy a specific constraint;
4. Retain: store the proposed solution in Case Base as a new solved case to be utilized in the next CBR cycle.

In this work, CBR approach is used as a problem solving methodology to support intelligent decision-making process which learns from the user interaction. As presented in following sections, we applied a Self-organizing Maps technique, to improve the case-based process, especially to make it more flexible – as a semiotic system would require being.

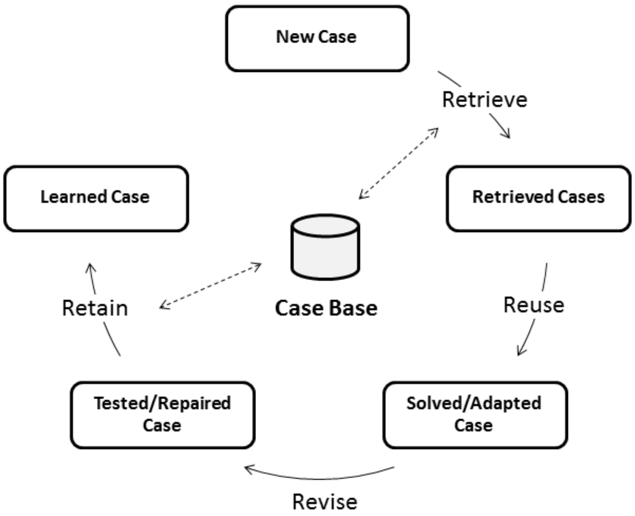


Fig. 2. CBR Cycle. Reasoning process organized in four steps: Retrieve, Reuse, Revise and Retain.

4 Self-Organizing Maps

The Self-organizing Map (SOM) introduced by Teuvo Kohonen in the early 80’s [12] is an unsupervised Neural Network algorithm. Inspired by the human brain cortex, SOM creates a topologically map in which similar data are grouped together by specialized neurons. Hence, this technique is a non-parametric model usually applied as a clustering or a pattern recognition algorithm [13].

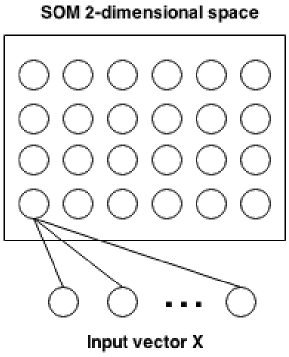


Fig. 3. Basic representation of a two-dimensional SOM.

A common representation of a SOM is a two dimensional grid of neurons, as shown in Fig. 3. Each output neuron is fully connected with an input neuron by a weight link.

The mapping process realized by a SOM arranges the input data preserving topological information [14]. Thus, the arrangement reflects the underling neighborhood relationships between data samples creating groups called clusters. Moreover, this arrangement often gives a quite relevant overview about what is not known, obvious or about the expected patterns; incidentally, information already present in data set.

During the training process, a given input data vector is presented to SOM and the most similar (closest) output neuron is select as a winner neuron, or best match unit (BMU). In Kohonen's work, similarity criterion used is the Euclidean distance [12] and the winner neuron is the one which has the minimum distance value. This process is unsupervised because no labeled data is required.

SOM employs a self-organizing training that aims to minimize the distance adjusting the BMU weights and its neighborhood toward the input vector using the following rule:

$$w_{ij}(t+1) = w_{ij}(t) + \ \alpha(t) h_{ci}(t) \left[x(t) - w_{ij}(t)\right], \tag{1}$$

where, w_{ij} is the weight j of neuron i, $\alpha(t)$ is the learning rate, and $h_{ci}(t)$ is the neighborhood radius function centered on winner neuron c with respect to neuron i. Typically, both leaning rate and neighborhood radius are functions which decrease with time, in order to facilitate system convergence.

Moreover, a trained SOM can be viewed as a data recognition system in which a set of specialized neurons respond to certain stimuli (i.e. a specific type or class of inputs), similar to the process that occurs in brain in which different neurons (sensory areas) respond to a different kind of sensorial stimulus [15].

Other SOM applications involve dimensionality reduction, knowledge discovery in databases (KDD) and intelligent decision support [16]. In our proposed model, a SOM is used as data visualization tool which allows users to identify similar cases during the Case-based Reasoning process.

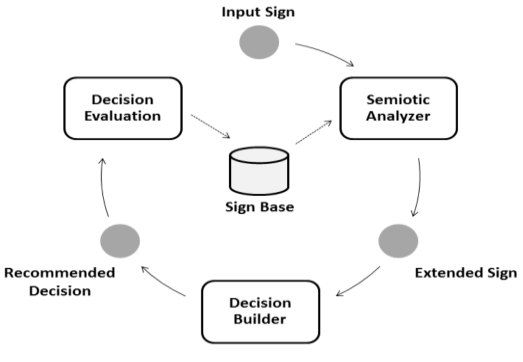


Fig. 4. Proposed subjective decision support process, inspired on CBR cycle.

5 Overview of the Proposed System

We aims to elaborate an intelligent system based on Semiotics and Computational intelligence techniques in order to support subjective decision-making. The conceptual construction followed a blend of CI and Semiotics ethos, derived hybridization of CI techniques, and inspiration on the CBR cycle. The Hybrid Subjective Decision Support Cycle is illustrated in Fig 4.

We use the concept of sign, proposed by Peirce, as the knowledge representation model. Similarly in CBR, a sign is a special type of case that represents a specific situation perceived by the system. In other words, signs can model user interest on a product (*e.g.* on a product for purchasing support or e-commerce applications or patient data in clinical domains, for example). A sign consists of three basic parts:

- Problem description: a feature vector describing both contextual information and situational description, corresponding to *representamen* in Peirce definition;
- Solution: contains subjective information about how the sign was interpreted by someone (*e.g.* a domain specialist or system users), as *interpretant* concept;
- Historical data: describes association among the sign and other signs previously perceived by the system.

Whenever a new problem situation appears, represented by an input sign, the Semiotic Analyzer module (SA) retrieves similar signs from the Sign Base. SA promotes a semiotic deconstruction of retrieved signs extracting contextual data in order to extend input sign. Thus, the extended sign (containing historical information) are adapted by the Decision Builder, combining data from retrieved signs, outputting a recommended decision. Hence, a user can test the decision and store it in the Sign Base for future application.

5.1 Retrieval of Similar Signs and Semiotic Deconstruction

To realize the retrieve phase, SA employs a SOM to search for similar signs. Each sign stored in Sign Base is indexed by a BMU in the SOM neuron grid during the system training phase, as proposed by Wang et al. [17]. Thus, similarity assessment is realized by calculating Euclidean Distance between the index of the input sign and all the indexes of previous signs. The top *n* indexes with smallest distance are selected and retrieved as similar signs.

Hence, SA employs a semiotic deconstruction on retrieved signs comparing their data with the input sign. This operation is realized analyzing each feature value in sign problem description part. Whenever the input sign presents an unexpected value, then it is marked with a warning tag. Deconstruction is described as follows:

- Consider S a sign feature vector: $S = \{s_1,\ s_2,\ s_3, \dots, s_n\}$
- For each feature i in retrieved signs feature set, easy calculations of average μ_i and standard deviation σ_i are carried out.
- Mark feature s_i with corresponding tag using the following rule:

$$tag(s_i) = \begin{cases} normal, & \mu_i - \sigma_i \leq s_i \leq \mu_i + \sigma_i \\ warning, & otherwise \end{cases}. \tag{2}$$

5.2 Recommended Decision

After extend the input sign with tags based on historical and similarity data, the Decision Builder module adapts the extended sign attaching a meaning to it, *i.e.* combine solution information from retrieved signs and associate it to input sign. This is simple, yet very relevant information for the aimed individualization process that tackles subjectivity. In this work, we employ most frequent adaptation to create the recommended decision. This adaptation process evaluates the solution part of all retrieved signs and uses the one with more occurrences to determine final solution decision configuration.

6 Experiments and Results

In order to evaluate our approach we apply our model to subjective decision making problems, where context of evaluation is central.

We select two benchmarking datasets provided by UCI Machine Learning Repository [18]: Pittsburgh Bridges and Auto Imports Database to try out our proposal: The former, in Semiotic Analyzer Module, and the latter in the Decision Builder Module. All data sets have been normalized using Eq. (3), nominal values converted into numeric values and entries with missing values were disregarded.

$$Normalized(x) = \frac{x - x_{MIN}}{x_{MAX} - x_{MIN}}, \tag{3}$$

where x is the current value, x_{MIN} and x_{MAX} are minimum and maximum values of x in dataset, respectively.

6.1 Selection of Parameters

Selection of parameters of computational models often impacts the obtained results quality. Especially in neural-based systems, the selection of appropriate parameters and training configurations can improve performance and avoid problems such as overfitting [1]. As this topics is widely discussed in the literature and out of the scope of the article, we assumed that parameters were selected using common heuristics for neural-system, *e.g.*, Kasaboy [22].

Related to SOM, some important parameters are neuron grid size, initial learning rate, neighborhood radius, and its updating methods. Although there are many approaches to set these parameters, some standards have been established by many authors [19][20]. In our experiments, we have used a 8×6 rectangular neuron map with initial learning rate as 0.1 (decreasing at each iteration to a minimum of 0.01) and performed 10 runs of 1000 iterations.

6.2 Evaluating Semiotic Analyzer Module

Pittsburgh Bridges dataset was used here as it contains information about several distinct bridges. Seven specification properties (problem and context information) and five design description properties were utilized. The evaluation task consists in enter an input bridge specification to the system for prediction of design properties. Results are summarized in Fig. 5 and Table 1.

To evaluate the quality of sign mapping provided by SA, based on SOM algorithm, we employ two measures: Quantization Error and Topological Error [12] [13]. Quantization Error (QE) evaluates the quality of SOM learning process computing how well the output neuron weights fits to data. It is calculated taking the average difference between N inputs and theirs correspondent BMU weights, as follows:

$$QE = \frac{1}{N} \sum_{i=1}^{N} |x_i - m_i| \tag{4}$$

where, x_i is the input vector and m_i is the BMU weight vector.

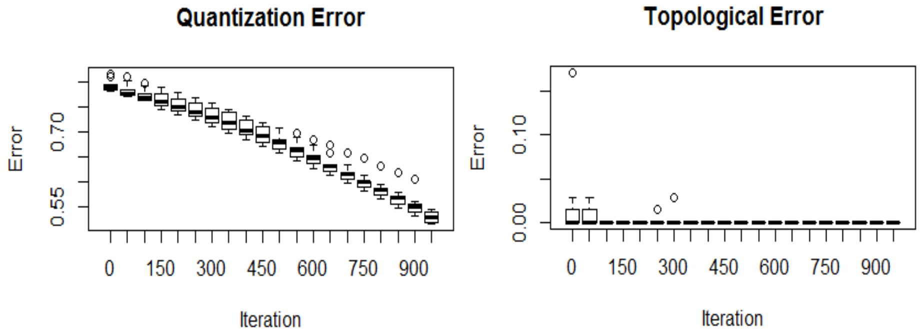


Fig. 5. Quantization and Topological errors for Pittsburgh Bridges dataset.

Table 1. Semiotic Analyzer evaluation results.

Properties	Input bridge	Bridge 1	Bridge 2	Bridge 3	SA tag
River	M	M	A	A	Warning
Location	3	6	28	25	Warning
Period	Crafts	Crafts	Crafts	Crafts	Normal
Purpose	Highway	Highway	Aqueduct	Highway	Normal
Length	Medium	Medium	Medium	Medium	Normal
Lanes	2	2	1	2	Normal
Clear-G	N	N	N	N	Normal
T-or-D	-	Through	Through	Through	-
Material	-	Wood	Iron	Wood	-
Span	-	Medium	Short	Short	-
Span length	-	S	S	S	-
Type	-	Wood	Wood	Wood	-

Topological Error (TE) represents the quality of map topologic preservation. It indicates the proportion of data items for which the first and second-best match units are not close to each other [13]. Whenever those units are not next in map error value is 1, otherwise 0.

As shown in Table 1, SA analyzed bridge specification properties about period, purpose, length, number of lanes and clear-G and retrieve similar bridges, only using contextual information. For a given input say, river and location, SA alerts existence of some divergent items in retrieved data, this, importantly, directs the user attention towards those properties in order to evaluate their values more carefully, helping to reduce cognitive overload on the decision-making process at the same time it continues to produce sound recommendations.

By analyzing QT and TE measures we illustrate that SOM behaves as an effective mechanism of, in this example, bridge information assessment, indexing bridge signs in Sign Base and grouping them in a similarity map; precisely, as hypothesized.

6.3 Purchasing Decision

Purchasing decision is another common subjective decision example. When people look for a product or service to purchase, they consider a large number of set of interest variables: price, brand, presence of a virtual shop, quality and so on. However, those variables are often impacted by personal experience or opinion, increasing decision making complexity, particularly for decision support systems.

In this example, our proposed system helps users to find an appropriate car based on theirs interests. We use information about 197 cars from Auto Imports Database. This dataset describes a car model based on properties such as number of door, body-style, horsepower, fuel type and price.

User inputs a query containing his interest about a car and, based on this information, system shows a set of car options. To retrieve appropriate car models, we defined the concept of relevance. A relevant product is the one that is in accordance with user-defined query at a minimum of 80%, related to each feature of the desired item. Thus, we use an adapted Heterogeneous Euclidean-Overlap Metric (HEOM) as relevance function. This function defines a distance between two items, here the query and each car model in dataset, as shown in Eq (5) and (6):

$$Relevance = \sqrt{\sum_{i=1}^{m} d(x_i, y_i)^2} \tag{5}$$

$$d(x_i, y_i) = \begin{cases} 1, \; if \; x_i \; or \; y_i \; are \; unknown \\ \frac{|x_i - y_i|}{max_i - min_i}, \; otherwise \end{cases} \tag{6}$$

Next, we apply Recall and Precision measures [21], widely used to evaluate information retrieval systems, in order to assess cars relevance. Precision is defined as the proportion of retrieved documents that are relevant, while Recall is the proportion of relevant documents that are retrieved by the system. Tables 2 and 3 summarize the results.

Table 2. Results of the Recommender of cars.

Car properties	User query	Car option 1	Car option 2	Car option 2
Num. of doors	2	4	4	4
Fuel type	Gas	Gas	Gas	Gas
Body style	Sedan	Sedan	Sedan	Hatch
Fuel consumption	-	29	29	29,5
Price	$10.000	$9.960	$9.995	$9.988
Relevance	-	100%	100%	99%

The proposed system proved to be able to perform the recommendation of products based on user defined interests, even if it is provided with incomplete information. Thus, there is no need to fill all the characteristics of the product by the user: a small amount of information is enough to guide the process of searching for relevant items.

Table 3. Information retreival measures.

Measure	Average	Standard deviation
Precision	89%	0,17
Recall	75,2%	0,82

The obtained results show an interesting behavior of the system. Although user has preferred a sedan car, system returned a hatch car. This behavior is important, considering real purchase situations where a human seller could offer a product of different types, but which suits customer's interests. Ultimately, this reveals the system ability to produce results that evidences hidden information, necessary for the subject decision.

7 Conclusion

In this work, we proposed a hybrid decision support system based on Semiotics and Computational Intelligence (*e.g.* Neural Networks and Case-Based Reasoning) readily applicable to subjective decision problems. Semiotics concepts, such as signs, sign deconstruction and contextual information processing allowed the development of a meaning deconstruction process which, applied together with Computational Intelligence methods, provide a degree of individualization of decisions, while improving efficiency and precision.

Initial experimental results illustrate that our approach is easy to be used, yet effective to help users along the selection of decision alternatives, reducing their cognitive overload. Additionally the proposal may also support a more intuitive subjective and evident decision making.

More experiments have to be cared out, especially within larger decision scenarios, *i.e.* more cases and conflicting features. Other distance measures could also tried out, as well as the assessment of appropriateness.

Acknowledgements. The authors would like to thanks CAPES and University of Pernambuco for the financial support of this research.

References

1. A. Engelbrecht: Computational intelligence: an introduction, 2nd ed. Chichester: Wiley (2007)
2. M. M. Richter and R. O. Weber: Case-Based Reasoning. Berlin, Heidelberg: Springer Berlin Heidelberg (2013)
3. R. Gudwin: From semiotics to computational semiotics. In Proceedings of the 9th International Congress of the German Society for Semiotic Studies/ 7th International Congress of the International Association for Semiotic Studies (IASS/AIS) (1999)
4. D. Chandler: Semiotics: the basics. Second ed. New York: Routledge (2007)
5. C. S. Peirce: Collected Papers of Charles Sanders Peirce. Harvard University Press (1958)
6. R. Gudwin and J. Queiroz: Semiotics and Intelligent Systems Development. Hershey: IGI Global (2006)
7. D. F. Pacheco: Hybrid Intelligent Suite for Decision Support. Vol. 55 (2007)
8. M. Kwiatkowska and K. Kielan: Fuzzy logic and semiotic methods in modeling of medical concepts. Fuzzy Sets Systems, Vol. 214 (2013)
9. J. Hugo: The semiotics of control room situation awareness. In Fourth International Cyberspace Conference. Ergon (2005)
10. A. Aamodt and E. Plaza: Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Commuity, Vol. 7 (1994)
11. J. L. Kolodner: An introduction to case-based reasoning. Artif. Intell. Rev., Vol. 6 (1992)
12. T. Kohonen: The self-organizing map. In Proc. IEEE (1990)
13. B. Hammer, A. Gisbrecht, and A. Schulz: Advances in Self-Organizing Maps. In 9th International Workshop, WSOM 2012 (2012)
14. S. Sklorz: Concept Formation by Combining Neural Networks and Machine Learning. (1996)
15. H. Ritter and T. Kohonen: Self-organizing semantic maps, Biol. Cybern., Vol. 254 (1989)
16. Z. Y. Zhuang, L. Churilov, F. Burstein, and K. Sikaris: Combining data mining and case-based reasoning for intelligent decision support for pathology ordering by general practitioners. Eur. J. Oper. Res., Vol. 195, no. 3, pp. 662–675 (2009)
17. H. Wang, J. Gao, and Z. Jiang: Rotating machinery fault case-based reasoning based on vector space model and SOM neural network. J. Comput. Inf. Syst., Vol. 19 (2012)
18. K. Lichman and M. Bache: UCI Machine Learning Repository. University of California, Irvine.
19. S. Haykin: Neural networks and learning machines. 3rd ed. New Jersey: Pearson Education, (2009)
20. T. Kohonen: Self-Organazing Maps. 3rd ed. Berlin: Springer (2001)
21. A. Drigas and J. Vrettaros: Using the Self-Organizing Map (SOM) Algorithm as a Prototype E-Content Retrieval Tool. Sci. Its Appl. (2008)
22. N. Kasbov: Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering (Computational Intelligence), MIT PRESS (1996)

Invoking Behavioral Patterns on Autonomous Virtual Agents by Fuzzy Cognitive Maps and Video Games Techniques Using Modern Tools

J. Carlos Conde R., Abraham Sánchez L., and Abraham Sánchez F.

Benemérita Universidad Autónoma de Puebla,
Computer Science Department, Puebla,
Mexico

{juanc.conde, asanchez}@cs.buap.mx, axy72@hotmail.com

Abstract. In order to increase the game characters realism and its artificial environments, recent research has focused on improving the autonomous adaptation in real-time of characters to particular situations. This work resume an experimental process for creating Autonomous Virtual Agents (AVAs). Our work shows the main phases where an appropriate cognitive modeling leads to suitable responses from external stimuli and internal desires. As first proof of this, an undersea world is built and used to test the perception capabilities of artificial fish and sharks. This behavioral system is based on MoCAMG cognitive architecture following a methodology that combines common video-game techniques with non classical methods of AI. The proposal highlights how simple Fuzzy Cognitive Maps allows to get high-level control and generate simple behavioral patterns equivalent to habits through realistic perception capabilities.

Keywords: Fuzzy cognitive maps, autonomous virtual agents, artificial life.

1 Introduction

According to Computer Graphics Modeling hierarchy considered in [8] by John Funge, Xiaoyuan Tu and Demetri Terzopoulos, to obtain realistic behaviors on Autonomous Virtual Agents (AVAs) the new apex is the Cognitive Modeling. Studies on living beings provide psychological, anatomical and ethological information, so many works argue that an appropriate cognitive model could provide enough information to generate realistic simulations of artificial life and validate these by theoretical models. Actually, video-games and auto-animations techniques facilitate realistic behaviors on characters.

This work focuses on the most important aspects for modeling AVAs in a synthesized way. It proposes to use a own architecture for cognitive modeling which combines Fuzzy Cognitive Maps (not augmented and not nested) and video-games techniques for blending animation and sensing. As result it obtains easily several behavioral patterns, without having to define every single detail of behaviors. In particular, it reproduces common habits of fish in an undersea world. Unlike other approaches, in this way the programmer do not need to know each and every one of possible world states using a complex data structure. Therefore, above is a proof of the high-level control obtained in this work.

The content of this paper is as follows. *Section 2* describes what is the most important behavioral components in living beings. A key piece is knowledge representation which is addressed in *Section 3*. Use of Fuzzy Cognitive Maps for this approach is discussed in *Section 4*. Related work is presented in *Section 5*. As part of this proposal, *Section 6* describes MoCAMG architecture for cognitive modeling and an overview of methodology of implementation. As proof MoCAMG, methodology and FCMs are used for implementing a simulation of artificial life using a set of fish and shark as predator in *Section 7*. Furthermore, this section shows the behavioral patterns obtained. Conclusions and future work are presented in Section *8*.

2 Behavioral Elements

Most important objectives in living beings are survive or reproduce, these are goals that can be decomposed into other more immediate. Animals have developed acute perception modalities to detect opportunities and threats in their habitat. They focus their attention in two ways: (1) *using their specialized sense organs* and (2) through *cognitive attention*. Therefore, the factors that must be considered by a behavioral system are [18]: the **environment**, **external stimuli** and **desires**, the **action selection** after obtaining sensory information and the **behavioral animation** which introduce motion control algorithms.

This situation lead to a design problem that can be solved by identifying the principles for which living beings select their actions, that is, the priorities for the different behaviors. An action selection mechanism based on reasoning involves the use of AI techniques and is called *task-level motion planning*. Meanwhile the behavior that enables artificial beings become autonomous and "survive" in dynamic environments is a primitive behavior: reactive or adaptive behavior [13,17,14]. According to Jafreezal Jaafar in [10], it should be considered specific elements for modeling an Intelligent Virtual Agent which is the basis of an AVA.

3 Knowledge Representation

Since the aim is to facilitate the inference of decisions, representation of information together with an interpretation theory give meaning to phrases of logic. In this case, an AVA should execute and action of "perception" or sensing to update its information about the world and then enable a *re-planning* process of actions. So is critical to select an appropriate architecture to implement effective cognitive models using a viable approach to represent uncertainty.

In most of related literature, *precondition axioms* define statements to specify what is the state of the world before performing an *action*. The outcomes of an action are represented by *effect axioms*. *Actions*, *effect axioms* and *precondition axioms* can be expressed as a tree data structure where *nodes* are *all* possible situations, *effect axioms* describe the characteristics of each situation (root corresponds to initial situation s_0) and *precondition axioms* allow to know which are the sequence of permitted actions, as seen in Figure 1. Each pathway on the tree represents one possible action sequence. Whether some situations (nodes) are defined as targets then is possible to use programming based in conventional algorithms for performing searches. The aim is to obtain one actions sequence which leads to character or AVA towards its objective [8].

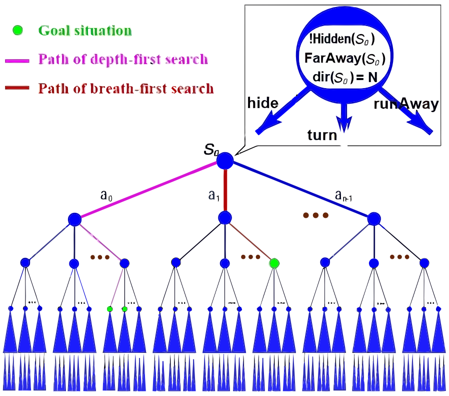


Fig. 1. Knowledge representation using "situation tree" presented in [8].

4 Fuzzy Cognitive Maps as a Viable Option

Mathematical models such as Navier-Stokes equations, Inverse Kinematics equations or even coupled differential equations are models that help to represent changes in virtual worlds. The main disadvantage is that these are hard to find, hard to solve and hard to run in real-time. Usually situation trees “facilitate” action selection process updating the stored knowledge. However, each inference uses only a small part of stored knowledge, and even the programer have to anticipate exactly what action could be selected in all possible conditions.

In contrast, Fuzzy Cognitive Maps (FCMs) [11] are a viable way to represent feedback because these are causal networks represented by fuzzy digraphs (see Figure 2), where nodes stand for concepts, actions or desires, and causal edges state the fuzzy rules between nodes. Furthermore, each input fires all the rules to some degree for modeling the “circular causality” of realistic virtual worlds.

FCMs by themselves act like a nonlinear dynamic system (such as an ANN) wherein all inputs are mapped to equilibrium states as output. Thus, a pathway in a simple FCM ends at a fixed point or in a finite loop.

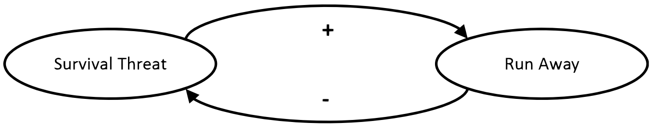


Fig. 2. Changes in a virtual world are causal and at the same time this causality feedback others [6].

5 Related Work

All the discussions point to the position that computational cognitive models can be true theories of cognition. Peter C. Pantelis and his team says in [15] that behavior of AVAs are modulated by a small number of distinct "mental states": exploring, gathering food, attacking, and fleeing. In two experiments, they studied subjects' ability to detect and classify the agents' continually changing mental states on the basis of their motions and interactions. The data provide intriguing evidence about the factors that influence over mental state, a key step towards a true "psychophysical intention".

Moreover, to get a believable behavior in characters the challenges are laid on generating on the fly flexible motion and complex behaviors inside of environments using realistic perception capabilities. N. Magnenat-Thalmann and D. Thalmann present in [12] research results which highlights interactivity and group behaviors are important parameters to test a cognitive model since realistic perception of environments and internal estate of the "actor" can be reflected affecting its behavior. Other interesting work more nearby to our approach is proposed by Toni Conde in [4] who presents two novel methodologies: the first technique integrates persistency to obtain a cognitive map of the perceived, the second technique integrates a perception approach by including the faculty of prediction, e. g., the orientation of the AVA attention.

Nevertheless, there are few works about Cognitive Architectures to improve AVAs in 3D environments using modern tools. A close architecture to this area is the RASCALS cognitive architecture presented in [2] which has an high expressive power to building advanced synthetic characters. But this is focused only on human cognition and it use an approach of Natural Language Generation (NLG) for the communication *user-character*.

In contrast, this work focus on cognitive modeling of autonomous agents or characters (not only humans) through endowing them with a certain amount of directives and letting them to generate the details by themselves. This approach considers the importance of modeling primitive actions properly and integrate them into complex actions combining an architecture, FCM and video-games techniques for blending animations and sensing. The component which is responsible of evoking "habits" considers the use of simple Fuzzy Cognitive Maps for modeling motivations in AVAs, but this could change using another approach of IA.

6 MoCAMG Architecture for Cognitive Modeling

The Movis' Cognitive Architecture for Modeling Game-characters (MoCAMG, Fig. 3) is a hybrid architecture based on the well known cognitive architectures ACT-R and Soar. This provides a conceptual frame work to create models of how performing tasks that involve movement. As ACT-R and Soar, MoCAMG has two types of memory; the *declarative memory* is associated with removable goals, so is regarded as a *short-term memory*. *Procedural memory* adds, updates but no removed "rules" in run-time as in a *long-term memory*.

A key component is the *Intentions Generator* because this allows to obtain complex behaviors considered as habits. This plays a role of behavior arbitrator that considers the internal and external state of a virtual agent.

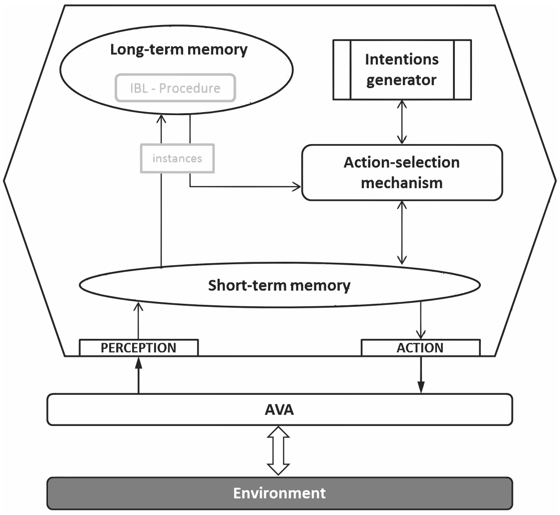


Fig. 3. MoCAMG architecture for cognitive modeling of Autonomous Virtual Agents.

Finally, we are currently working in the *IBL-Procedure* which serves to select an appropriate response (action) from a set of *instances* defined by situations (internal and external state) stored in the long-term memory. The purpose of this component is to obtain more weighted responses. However, the scope of this work focuses on the basis of reactive behavior to achieve integration and functionality in a simple way; using simple FCMs. Therefore, *Intentions Generator* can evoke "habits" or behavioral patterns through a high-level control (without defining every detail of movement). This is an important step towards more realistic behaviors.

6.1 Component-Based Behavioral System

As overview of the methodology used and to synthesize the cognitive modeling process in AVAs, a diagram of workflow implemented is shown in Fig. 5. This is based on the hierarchy of behavioral animation modeling since it has made impressive strides toward autonomy resulting in virtual characters auto-animated endowed of more realistic behaviors [9,8,7,5].

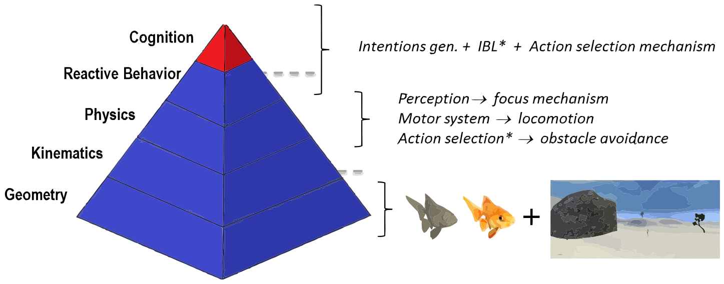

Cognition
Reactive Behavior
Physics
Kinematics
Geometry
Intentions gen. + IBL* + Action selection mechanism
Perception → focus mechanism
Motor system → locomotion
Action selection* → obstacle avoidance
+

Fig. 4. Computer Graphics Modeling (CGM) hierarchy.

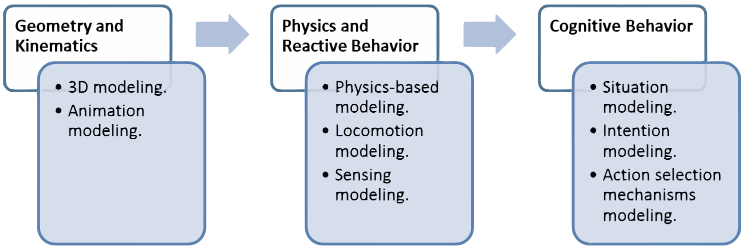


Fig. 5. Workflow for cognitive modeling which considers three levels of modeling: realistic appearance, realistic smooth and flexible motion and realistic high-level behaviors.

At this point is important to specify that from now on it will consider a *desire* is a potential influence of *intention*. An agent may have multiple desires, but only those which are the strongest desires will be considered collectively as intention. Additional information should be represented as *qualitative recommendations* of actions (referred as *motor preferences*) then is feasible to implement an Intention Generator using non-classical techniques of AI.

7 Implementation of Autonomous Virtual Fish Using MoCAMG Architecture

Fish and undersea world were implemented to test complex movements in full 3D space and to simulate the paradigm of *artificial life*. This consist of several goldfish, a shark, plants and different obstacles. This simulation was developed programming in LUA and using ShiVa3D as game engine.

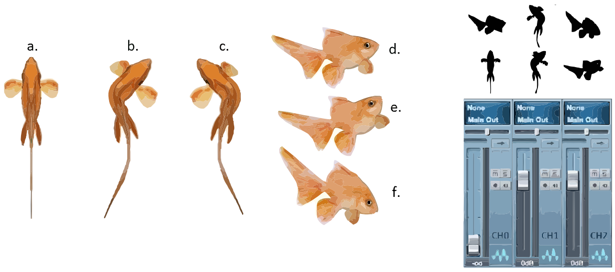


Fig. 6. Left: atomic movements. Right: manager of animations scheme.

The anatomy and movement form are set according to the related ethology [16]. This was modeled as shown in Figure 6 (left), where the combination of atomic actions generates necessary movement sequences. This represents the main idea of Barthel, Dannenmann and Hagen [1] where behaviors can be either low-level or high-level. Two or more simple movements can be executed simultaneously, as long as these are non-exclusive movements, through a component named manager of animations (scheme on the right side in Figure 6).

The implemented perception system considers four sensing models: (1) Avoidance of collision with terrain, (2) Avoidance of collision with fixed obstacles, (3) Avoidance of collision with dynamic obstacles and (4) Physical touch.

Capability to *avoid collision with terrain* was designed to be activated only when the agent is displacing using a ray casting method, it draws a distinction between go forward and go backward. *Avoidance of collision with fixed obstacles* is less simple; real fish use landmarks to generate mental maps of geometric relationships [3], so the calculation of attractive potential field is viable way to define a target in the space while calculation of repulsive potential field serves to avoid "known" obstacles.

However, fish possess highly developed sense organs, like chemoreceptors which detect gentle currents and vibrations (e. g. lateral line system). Therefore, to model similar sensory capabilities this implementation models the extended perceptual range as spherical sensor that detects other similar sensors. A visual perception model and the extended perception model together can detecting moving objects or even other cognitive entities (see Figure 7). Subsequently, is necessary to simulate physical contact to define more complex behaviors. So a *ragdoll* is used to simulate catching of food and physical touching in group behaviors.

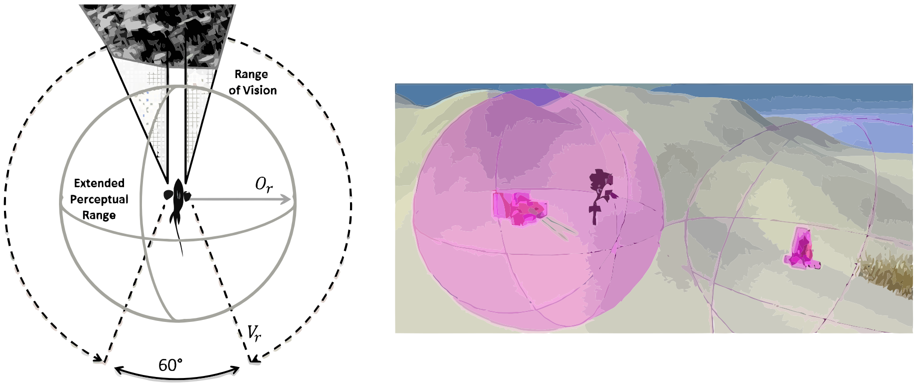


Fig. 7. Sensing modeling which attempts to simulate special skills of perception in fish (range of vision V_r and extended perceptual range O_r).

Updating process of cognitive agent's world model is also a primitive action. In turn, this primitive action requires sensing actions where the behavioral system selects which to execute to discard unnecessary sensory information. The reactive system is responsible for executing motor controls efficiently, but this works with the action selection mechanisms to manage the suitable perception model and the most useful motor controller. Intentions generator controls these subsystems in a higher level to satisfy imperative desires at a given moment.

A hierarchy of intentions considered in [18], establishes theoretical basis to define the following scheme for this implementation.

Intention to *avoid collision*
⇓
Intention to *run away* from predator
⇓
Intention to *rest* / Intention to *eat*
⇓
Intention to *school*
⇓
Intention to *wander* / Intention to *leave*

Intention to avoid collision is directly related with the data obtained from the sensing system, i. e., this is a reactive behavior. Intentions to wander and leave are the result of absence of other. However, for the implementation there are variables considered as desires; *hunger*, *fatigue* or *survival threat*, which governing in a sub-cognitive level all above intentions.

The FCM for a fish was taken from Julie A. Dickerson and Bart Kosko's work in [6] and this is shown in Figure 8. This is represented through a square matrix and each component takes values between -1 and 1. Therefore the matrix establishes the causality flow between concepts of a virtual fish's FCM.

	F_1	F_2	F_3	F_4	F_5	F_6	F_7
F_1	1	0	0	-1	1	0	0
F_2	1	0	1	0	-1	0	0
F_3	1	-1	0	1	-1	0	0
F_4	1	0	0	0	0	-1	0
F_5	-1	0	0	0	0	0	0
F_6	0	0	0	1	-1	0	1
F_7	1	1	0	-1	-1	-1	0

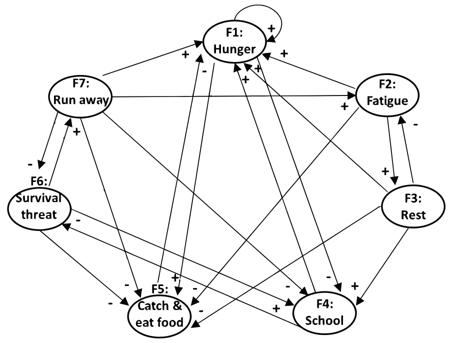
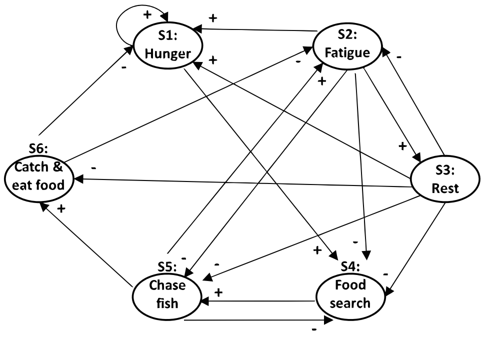


Fig. 8. Trivalent FCM implemented by Intentions Generator of virtual fish.

The FCM implemented for modeling an Intention Generator of a shark is based also in [6]. However, according to theory underlying, it was modified to obtain effective attracting points for the action selection mechanism. In this case, the action selection mechanism works as activation function for an ANN. Figure 9 shows the proposed FCM and its square matrix.



	S_1	S_2	S_3	S_4	S_5	S_6
S_1	1	0	0	1	0	0
S_2	1	0	1	-1	-1	0
S_3	1	-1	0	-1	-1	-1
S_4	0	0	0	0	1	0
S_5	0	1	0	-1	0	1
S_6	-1	-1	0	0	0	0

Fig. 9. Trivalent FCM implemented by Intentions Generator of virtual shark.

The approach presented in [6] proposes to use an *augmented FCM* to implement a virtual undersea world in 2D, where causal links are all listed in a single square matrix. However, the approach of this work is more ambitious because it attempts to test all cognitive capabilities defined by the proposed MoCAMG architecture inside of a 3D environment. Even the approach of Julie A. Dickerson and Bart Kosko considers an explicit "mental" connection between characters, which is impossible in real world, at least so far.

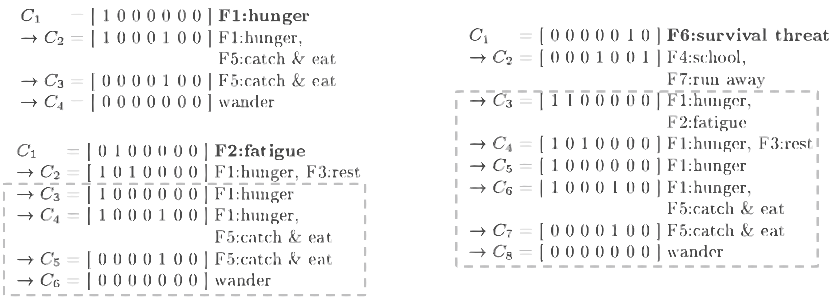
Considering all above, the contribution is resumed in the Algorithm 1. This represents the integration of several approaches in this area. Proposed algorithm could be generalized to other characters, specially for those with similar cognitive capabilities. This proves the usability of the architecture and FCMs.

Algorithm 1 Artificial behavior.

```
Initialize global variables
Load animations
Load FCM
Load repulsive potential field
Initialize sensors
Enable critical sensors
Get bounding-box data from 3D model
Get current rotation data (Y axis)
Initialize variable $Fatigue$ randomly
Initialize variable $Momentum$ (frame/s)
for Each frame do
    if locomotion is enable then
        Enable sensing for extended perception
        Update sensory information
        Avoid collision with obstacles
        Avoid collision with terrain
        Update movement
        Moving towards goal
    else
        Enable float
        Disable sensing for extended perception
        Decrement variable $Fatigue$
    end if
    Generate intentions
end for
```

7.1 Resulting Behavioral Patterns Using FCMs

Resulting information from the sensing system and internal state enables *Intentions Generator* to fire the current state into the FCM. This is represented by a vector of seven components: $C_i(t) = [F_1\ F_2\ F_3\ F_4\ F_5\ F_6\ F_7]$. Bounded loops form hidden behavioral patterns through nested loops observed in Figure 10.



$C_1 = [\,1\ 0\ 0\ 0\ 0\ 0\ 0\,]$ **F1:hunger**
$\rightarrow C_2 = [\,1\ 0\ 0\ 0\ 1\ 0\ 0\,]$ F1:hunger, F5:catch & eat
$\rightarrow C_3 = [\,0\ 0\ 0\ 0\ 1\ 0\ 0\,]$ F5:catch & eat
$\rightarrow C_4 = [\,0\ 0\ 0\ 0\ 0\ 0\ 0\,]$ wander

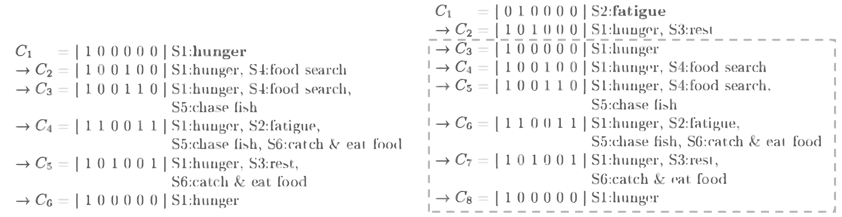
$C_1 = [\,0\ 1\ 0\ 0\ 0\ 0\ 0\,]$ **F2:fatigue**
$\rightarrow C_2 = [\,1\ 0\ 1\ 0\ 0\ 0\ 0\,]$ F1:hunger, F3:rest
$\rightarrow C_3 = [\,1\ 0\ 0\ 0\ 0\ 0\ 0\,]$ F1:hunger
$\rightarrow C_4 = [\,1\ 0\ 0\ 0\ 1\ 0\ 0\,]$ F1:hunger, F5:catch & eat
$\rightarrow C_5 = [\,0\ 0\ 0\ 0\ 1\ 0\ 0\,]$ F5:catch & eat
$\rightarrow C_6 = [\,0\ 0\ 0\ 0\ 0\ 0\ 0\,]$ wander

$C_1 = [\,0\ 0\ 0\ 0\ 0\ 1\ 0\,]$ **F6:survival threat**
$\rightarrow C_2 = [\,0\ 0\ 0\ 1\ 0\ 0\ 1\,]$ F4:school, F7:run away
$\rightarrow C_3 = [\,1\ 1\ 0\ 0\ 0\ 0\ 0\,]$ F1:hunger, F2:fatigue
$\rightarrow C_4 = [\,1\ 0\ 1\ 0\ 0\ 0\ 0\,]$ F1:hunger, F3:rest
$\rightarrow C_5 = [\,1\ 0\ 0\ 0\ 0\ 0\ 0\,]$ F1:hunger
$\rightarrow C_6 = [\,1\ 0\ 0\ 0\ 1\ 0\ 0\,]$ F1:hunger, F5:catch & eat
$\rightarrow C_7 = [\,0\ 0\ 0\ 0\ 1\ 0\ 0\,]$ F5:catch & eat
$\rightarrow C_8 = [\,0\ 0\ 0\ 0\ 0\ 0\ 0\,]$ wander

Fig. 10. Behavioral patterns for virtual fish determined by bounded loops.

Since behavioral patterns can be considered as habits, this are described in Figure 10. Notice that in several cases there are more than one component set in one, so note the importance of implement a hierarchy of intentions into the *Action Selection Mechanism*.

As proof of the simple way to cognitive modeling using MoCAMG, FCMs and a game engine, a predator character was added using basically the same algorithm and changing the data structure of its FCMs. According to ethology, the artificial predator is a *black-tips shark*. In this moment the character has an acceptable reactive behavior.



$C_1 = |\ 1\ 0\ 0\ 0\ 0\ 0\ |$ S1:**hunger**
$\rightarrow C_2 = |\ 1\ 0\ 0\ 1\ 0\ 0\ |$ S1:hunger, S4:food search
$\rightarrow C_3 = |\ 1\ 0\ 0\ 1\ 1\ 0\ |$ S1:hunger, S4:food search, S5:chase fish
$\rightarrow C_4 = |\ 1\ 1\ 0\ 0\ 1\ 1\ |$ S1:hunger, S2:fatigue, S5:chase fish, S6:catch & eat food
$\rightarrow C_5 = |\ 1\ 0\ 1\ 0\ 0\ 1\ |$ S1:hunger, S3:rest, S6:catch & eat food
$\rightarrow C_6 = |\ 1\ 0\ 0\ 0\ 0\ 0\ |$ S1:hunger

$C_1 = |\ 0\ 1\ 0\ 0\ 0\ 0\ |$ S2:**fatigue**
$\rightarrow C_2 = |\ 1\ 0\ 1\ 0\ 0\ 0\ |$ S1:hunger, S3:rest
$\rightarrow C_3 = |\ 1\ 0\ 0\ 0\ 0\ 0\ |$ S1:hunger
$\rightarrow C_4 = |\ 1\ 0\ 0\ 1\ 0\ 0\ |$ S1:hunger, S4:food search
$\rightarrow C_5 = |\ 1\ 0\ 0\ 1\ 1\ 0\ |$ S1:hunger, S4:food search. S5:chase fish
$\rightarrow C_6 = |\ 1\ 1\ 0\ 0\ 1\ 1\ |$ S1:hunger, S2:fatigue, S5:chase fish, S6:catch & eat food
$\rightarrow C_7 = |\ 1\ 0\ 1\ 0\ 0\ 1\ |$ S1:hunger, S3:rest, S6:catch & eat food
$\rightarrow C_8 = |\ 1\ 0\ 0\ 0\ 0\ 0\ |$ S1:hunger

Fig. 11. Behavioral patterns for virtual shark determined by permanent desires of *hunger* or *fatigue*.

8 Conclusions and Future Work

This work represents a very important step towards the generation of cognitive characters with realistic behaviors, in a synthesized way. This is an important approach to obtain autonomous characters or AVAs combining video-games techniques. The proposal highlights how behavior organized in hierarchical way leads to obtain a high-level control which compete for reflexes rather than reflexes compete for control of "muscles" or other details. Even, this first implementation serves as an useful tool to test and improving other cognitive procedures.

There is a lot of future work, which is good because this attempts to explore more cognitive phenomena related with long-term memory. One of these is the Instance Based Learning based on the Instance Based Theory. And regarding the tests, it will be necessary to introduce more predator characters and cause antagonism defining another role, specifically introducing a dolphin character. But prior this work should must evolve to improve cognitive capabilities to evoke more complex behaviors, thus generating artificial learning and reasoning.

References

1. Barthel, H., Dannenmann, P., Hagen, H.: Towards a General Framework for Animating Cognitive Characters. In: Proceedings of the 3rd IASTED International

Conference on Visualization, Imaging, and Image Processing. VIIP'03, Benalmádena, Spain (2003)

2. Bringsjord, S., Khemlani, S., Arkoudas, K., Mcevoy, C., Destefano, M., Daigle, M.: Advanced synthetic characters, evil. In: and E," in Game-On 2005, 6th International Conference on Intelligent Games and Simulation (M. Al-Akaidi and. pp. 31–39. European Simulation Society (2005)
3. Chung, S.: Appropriate maze methodology to study learning in fish. University of Toronto Journal of Undergraduate Life Sciences 2(1) (2009)
4. Conde, T., Thalmann, D.: An integrated perception for autonomous virtual agents: active and predictive perception. Journal of Visualization and Computer Animation 17(3-4), 457–468 (2006)
5. Dannenmann, P., Barthel, H., Hagen, H.: Multi level Control of Cognitive Characters in Virtual Environments. In: Proceedings of the 14th IEEE Visualization 2003 (VIS'03). pp. 92–. VIS '03, IEEE Computer Society, Washington, DC, USA (2003)
6. Dickerson, J.A., Kosko, B.: Virtual Worlds as Fuzzy Cognitive Maps. Presence 3(2), 173–189 (1994)
7. Funge, J.: Cognitive modeling for games and animation. Commun. ACM 43(7), 40–48 (Jul 2000)
8. Funge, J., Tu, X., Terzopoulos, D.: Cognitive modeling: knowledge, reasoning and planning for intelligent characters. In: Proceedings of the 26th annual conference on Computer graphics and interactive techniques. pp. 29–38. SIGGRAPH '99, ACM Press/Addison-Wesley Publishing Co., New York, USA (1999)
9. Funge, J.D.: Making them behave: cognitive models for computer animation. Ph.D. thesis, University of Toronto, Toronto, Ont., Canada (1998)
10. Jaafar, J., McKenzie, E.: A reactive architecture for autonomous agent navigation using fuzzy logic. The University of Edinburgh, Edinburgh, UK (2009)
11. Kosko, B.: Fuzzy cognitive maps. International Journal of Man-Machine Studies (24), 65–75 (1986)
12. Magnenat-Thalmann, N., Thalmann, D.: Virtual humans: thirty years of research, what next? The Visual Computer 21(12), 997–1015 (2005)
13. Manning, A., Dawkins, M.: An Introduction to Animal Behaviour. Cambridge University Press (1998)
14. McFarland, D.: Animal behaviour: psychobiology, ethology, and evolution. Longman Scientific and Technical (1993)
15. Pantelis, P.C., Cholewiak, S., Ringstad, P., Sanik, K., Weinstein, A., Wu, C.C., Feldman, J.: Perception of intentions and mental states in autonomous virtual agents. Cognitive Sciences Society 33rd Annual Meeting pp. 1990–1995 (2011)
16. Stephens, K., Pham, B., Wardhani, A.: Modelling fish behaviour. In: Proceedings of the 1st International Conference on Computer Graphics and Interactive Techniques. pp. 71–78. GRAPHITE '03, ACM, New York, USA (2003)
17. Tinbergen, N.: The study of instinct. Clarendon Press (1951)
18. Tu, X.: Artificial animals for computer animation: biomechanics, locomotion, perception, and behavior. Springer-Verlag, Berlin, Heidelberg (1999)

Making Decisions with Reactive Multi-agent Systems: A Possible Alternative to Regular Decision Processes for Platoon Control Issue

Baudouin Dafflon and Franck Gechter

IRTES-SET, UTBM, Belfort,
France

franck.gechter@utbm.fr,baudouin.dafflon@utbm.fr

Abstract. Decision processing is a key element in computer science, in automatic control and in robotics. The literature presents a lot of various approaches for decision processing. These approaches generally depend on both the adopted conceptual point of view and of the application field. Some of the classical methods suffers from several problems such as a limited adaptivity, or a high computational cost. In this context, reactive multi-agent systems can be good candidates to overcome some of these drawbacks. The goal of this paper is to present how to use reactive multi-agent systems to make decision, what are their advantages and their drawbacks as compared to classical methods. It also illustrates the proposal with an example of application dealing with the platoon control issue.

Keywords: Reactive multi-agent systems, platoon control, vehicle control

1 Introduction

Decision processing is a key element in computer science, in automatic control and in robotics. It can be defined as the process of choosing one action or a sequence of actions aimed at reaching one goal while respecting constraints (energy consumption, kinematic and dynamical limits, etc.). Thus, a good decision process must take into account both the problem constraints and the best way to reach system's goal. In some cases the decision process can be composed of several steps [21]: (1) Definition of the problem; (2) Definition of the goal(s); (3) Problem solving process; (4) Evaluation of the obtained solutions; (5) Selection of one solution and application of the related actions. In terms of mobile robotics the goal can have different definitions depending on the level of abstraction that is used. One can find high level definitions such as gathering specific information, collecting a specified amount of stones in mars rover mission or low level definitions such as reaching one position, performing one specific action, etc.

In literature, one can find a lot of various approaches for decision process. Among the most widespread, one can cite artificial neural networks [16] the fuzzy logic [19], or the stochastic approaches such as Markov decision processes [26] or those based on Bayesian networks. Of course, the literature contains also multi-agent based problem solving methods and decision processes such as [25].

The classical methods suffers from several problems which, even if they are acceptable in most cases, can be dissuasive for some applications. For instance, neural networks have limited adaptation skills due to the way the neural weights are computed/learned, Markov decision processes require a large amount of knowledge to be usable (transition function, set of states, set of actions, reward function, etc.) and suffers from a high computational cost (generally in $o(nbr^2_{states})$) making them hard to use for multi-robot applications in dynamical environment, etc.

Since a couple of years multi-agent systems, have became more and more wide-spread for a wide range of applications (simulation, problem solving, complex system modelling, etc.) in computer science and especially in artificial intelligence. Their properties (adaptability, robustness, etc.) can make them good candidates for developing new kinds of decision processes.

A multi-agent system can be defined as a set of entities, named agents, in interaction with each-other and with their evolution area, generally named environment, and aimed at producing collectively a spatio-temporal organization. The main difficulty while developing such systems is prediction and the verification of the organizational global properties while having knowledge on only local agents behaviours and interaction mechanisms. In literature, two principal trends can be encountered: the cognitive and the reactive approaches. The cognitive trend relies on the interactions between entities with high level reasoning, behavioural and/or perceptual abilities that can be qualified to be intelligent. By contrast, the reactive approach focuses on agents interactions instead of their intrinsic capacities. The main interest of this kind of approaches is this ability to adapt to a wide range of domain while keeping a conceptual and functional simplicity of the programmed entities. These models are well adapted to simulation/modelling of complex systems and to dynamic problem solving methods. They are used in various application fields such as artificial life simulation [12], multi-vehicle navigation [15], etc. As for the interaction models, several inspiration sources can be found in literature. Among the most widespread, one can cite the biological inspired approaches and the models based on Physics. The first ones can be based on the use of artificial pheromons or on the adaptation of biological behaviours on specific problems [3]. The Physics inspired approaches are based whether on regular forces [6] or force fields [2].

Using a reactive multi-agent system for building a decision process is, in some cases, natural especially when the initial problem to solve is already distributed on the functional or topological points of view or when some elementary coupling between problem entities are already known. Moreover, literature contains also several conception methodologies/principles among which one can cite [4], [22], etc. Even if some of them are hard to be adapted to a wide range of problems, it is possible to extract from them guidelines for conceiving such systems. As for the reactive systems, the task is hard to tackle with due to their complexity (linked to the huge number of entities in interaction) and to the emergent and non-predictible aspect of the global properties. Thus, conceiving a reactive multi-agent system for decision purpose implies to focus on interactions and on the way they are put into practice (directly or indirectly through environment) instead on focusing on agent themselves. More over, one must also define how the link between the problem and the agency is made and how to convert emergent properties evaluation into a suitable set of actions. The general structure of a reactive multi-agent decision process can be represented by Figure 1.

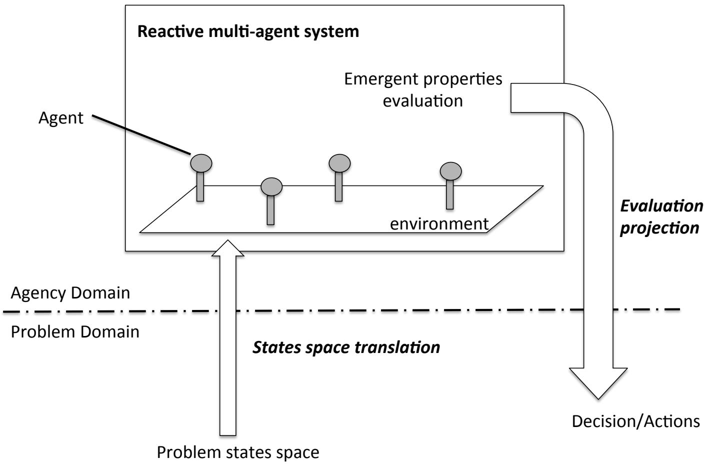


Fig. 1. General description of a reactive multi-agent decision process.

The goal of this paper is to explain how reactive multi-agent systems can be used to make decision and what kind of advantages (and drawbacks) they can provide as compared to regular methods. To that way, we will give a description of the key elements (problem interpretation, problem solving and action(s) selection) that have to be taken into consideration to tackle this kind of issue. These descriptions will be illustrated by an example dealing with the platoon control issue.

The paper is structured as follows: Section 2 gives a global presentation of the key elements required for reactive multi-agent system decision processes. This section details, in particular, the role of the environment (Section 2.1) and how the problem and its characteristics/constraints can be translated into this structure common to all agents, the role of the interactions and how they can be chosen (Section 2.2) and how to convert the agency emergent properties into suitable actions for the system (Section 2.3). This section is concluded by a presentation of the most widespread application fields and of the limitations of the reactive approach. Section 3 shows an instanciation of the approach to the platoon control issue. Finally, Section 4 concludes this paper giving the main limitations encountered with these new kind of approaches and giving some clues for further works.

2 Key Elements for Reactive Multi-Agent Systems for Decision Process

2.1 Agent Environment: An Active Element of the Decision Process

The role of the environment is particularly important in multi-agent systems designed for decision processing. Generally, this one is built so as to "translate" the problem into the agents world. This translation relies on a projection of space state of the problem into a space state which can be travelled by the agents.

Since several years, works on multi-agent systems give a more and more active role to the environment especially when agents are situated [23]. In such systems, the environment is an active entity with its own behaviours. Thus, it can handle various role such as shared memory, interaction catalyst or coordination entity as in pheromon based approach or stigmergic approaches.

Several models for environment can be used. [20] proposes an environment which defines perception/action relations with agents. Environments can be whether centralized or decentralized. Finally, in some cases, the environment can be defined as a standalone multi-agent system [12].

In the decision making context, the environment has got the central role. Since the agents behaviours are as simple as possible, the system can compute only thanks and in the environment. As compared to classical decision making approaches, the main interest of this active environment is to be able to link together the problem world and the solving process world. Besides, since the environment has got its own behaviours, it can adapt, while maintaining a temporal coherence of the information on the problem, to the dynamical evolution of it. This dynamical adaptation cannot be made in such a wide extend with Markov decision processes or in artificial neural networks.

2.2 Agents Interactions: A Conceptual Choice

An interaction is defined as a dynamical relation between agents through a set of mutual actions. These actions can have an influence on other agents perception and then on their further actions. The main difference with classical systems is the importance of this interaction that prevails on the definition of the agents themselves. These inter-agent links can be direct or indirect using the environment as in pheromon based approaches. Two main inspiration sources are generally used for defining interactions: Biology or Physics.

Biological approaches relies on the study of the behavioural mechanisms of social insects. Among the most widespread, one can cite ant based algorithms which relies on artificial pheromons dropped into environment and aimed at sharing information at global level [11]. Other models, inspired by wolves, fireflies or social spiders have been also be used in several context. The main interest of bio-inspired approaches is the fact that the global emergent organisation is known in advance. Thus, the engineering strategy relies on an adaptation of existing and already observed behaviours to a specific decision process. The decision is obtained when an equilibrium between an amplification mechanism and a regulation mechanism is reached. The decision relies then on the interpretation of the information persistence into environment such as the geometrical repartition of agents.

Physics inspired approaches are also widespread. As opposed to biological inspiration, the engineering methodology is not based on the global expected result but on the definition of the behaviour one want to provide to agents. These methods generally relies on attraction/repulsion mechanisms. As in biological paradigm, the goal is to obtain a stable or meta-stable state of equilibrium which can take the form of a specific spatial organization of agents.

To summarize, the goal of the interaction mechanisms, whether they are Bio or Physics inspired, is to obtain a spatio-temporal organization that corresponds to the result of the problem space state exploration by the agents. When a perturbation occurs, the system evolves into another organization which corresponds to the new available data on the problem.

2.3 Evaluating Agency Properties: the Key for a Good Decision

Agents, Complex Systems, Emergence and Self-Organization The evaluation of the agency properties is the most difficult part to handle due the the complex character of reactive multi-agent systems. A complex system is considered as a set of components in interaction, the global behaviour of which, and its evolution, can neither be computed nor predicted by an external observer. Thus a system is said to be complex if the global final result can be predicted only by experiments or simulations even if all components and internal relations are perfectly known. Reactive multi-agent systems can be considered in some ways as complex systems. Even if the internal local mechanisms are perfectly known, the application of them by each agent, depending on their perception and behaviours, lead to a dynamic un-predictible variation of the agency properties. These considerations allow to differentiate standard particle systems from particle argent systems (i.e. systems in which agents acts as particles, the behaviours of which change in run-time relatively to their perception).

The problem of the global evaluation has to be put in parallel with emergence and self-organizing notions that lead to numerous publications on both practical and theoretical levels. Self-organization is generally linked to an increase in the order of the system without any possible external control of the process that lead to this result. In [9], self-organizing property is defined as a "dynamical and adaptive process allowing the system to acquire or maintain a structure without any external control". If the self-organization ability of one system is defined with a consensus in the scientific community, the definition of emergence is far from being as clear. The common sense is linked to the presence of an external observer able to detect, measure and analyse global phenomena produced by a process. Emergence leads to many works in computer science [10], complex systems studies [17], in sociology, etc. As the study of the complex system cannot be performed in a reductionist frame, the emergence leads to the same kinds of issues. Thus, it can not be considered as a system property but as a property of the point of view one can have on the system. In [9], a system is defined to have emergent properties when phenomena appear dynamically at a macroscopic level thanks to the microscopic actions that link the system components together.

Due to the complexity of such systems and to the difficulties encountered to define the emergence, this part is sometimes hard to tackle. The evaluation of the emergent properties is generally split into two steps: (1) Developing a pertinent measure of the agency properties; (2) Projecting this measure into the possible actions space.

Evaluating Emergent Properties The evaluation of emergent properties allows to have a pertinent analysis of the solution given by the multi-agent system. In many examples, this relies on a simple observation of the global result. In this case, the engineering process is similar to a reinforcement learning such as the one that can be found in artificial neural networks. Once the system is built, a tuning phase of the local parameters, generally made empirically taking into account system goal, has to be done. Some systems are also based on more precise evaluation functions that are not necessary tied directly to the problem domain. These functions take into account both local and global properties. The literature proposes solutions inspired by Biology (fitness function), Sociology (utility function, satisfaction, altruism, etc.) or Physics (entropy, energy, etc.)

Once the evaluation is made, the projection function can be defined relying on mathematical considerations and problem knowledge.

2.4 Application Fields, Properties and Difficulties

The global structure and the different steps of a multi-agent decision process have been sketched into the previous sections. Until now, multi-agent decision processes applied to robotics can be found scarcely in literature. They have been used for multi-configuration platoon control [15], [18], obstacle avoidance [24], driving assistance [13] or automatic parking device [7].

The properties are mostly tied to the multi-agent aspect of these systems such as adaptability or robustness. For instance, multi-agent based obstacle avoidance algorithms allow to deal with one fixed obstacle but also with several moving obstacles without changing any elements of the system. By the same, platoon control algorithms are able to adapt the platoon structure according to vehicles/robots perceptions. Moreover, the number of agents can be adapted to the available computational resources providing a decision process with a controllable complexity independently to the dynamical variations of the problem.

If we assume that the local interactions are aimed at maintaining an equilibrium between problem variations (translated into agents environment) on one side and the goal of the system on the other, the two main difficulties are in the translation of the problem space state into a suitable space environment for agents and the interpretation of the emergent properties/structure as a decision. The first difficulty is mainly linked to the dimension of the problem states space and the desired dimension for the multi-agent system environment. The second problem is more delicate since the projection of the agency properties relies mainly on a definition of a "good" estimation function. This function has to takes into account agents interactions, emerging structure, problem domain, etc. Unfortunately, there is no method to design such a function for the moment.

3 Application to Platoon Control

The goal of this section is to present how to instantiate the preceding proposals within a concrete example. After a state of the art of the platoon control issue, we will explain how to apply the previous statements to a platoon of vehicles.

3.1 State of the Art

A platoon is generally considered as a set vehicles moving synchronously following a predefined geometrical configuration (column, line, echelon, etc.) without any material coupling. Platoon control methods are generally divided into two categories: global and local. In global approaches, vehicles are controlled by a system which relies on information shared by vehicles on the convoy level. Global approaches mostly requires a vehicle to vehicle (V2V) and/or a vehicle to infrastructure (V2I) communication system. For instance, [1] proposes a system where vehicles are represented by a bicycle cinematic model. Lateral and longitudinal control are separated. Lateral control is made using a shared reference trajectory. Longitudinal control is made using the curvilinear abscissa of the leader vehicle over the reference trajectory. In [14] an algebraic graph theory is used. The vehicles exchange information according to a pre-specifies communication graph. The feedback control is based only on relative information about vehicle states shared via the communication links. By contrast, local approaches are based on reactive capability of vehicles. Vehicle actions are based on its own perceptions. In [8], for instance, an automatic control mechanism based on the PID controller has been proposed. Longitudinal control consists in maintaining a time $T = \frac{D}{V}$ constant between two vehicles, where D is the distance between two vehicles and v is the vehicle speed. By contrast, lateral control consists in maintaining the direction angle between the vehicle and its predecessor close to zero. [5] presents local approaches based on a virtual mechanical link inspired by physics. Each vehicle is considered to be able to perceive and reacts while maintaining a constant distance with its predecessor using an impedance control model composed of springs and dampers.

In this paper, a local approach for platoon system is presented using the reactive multi-agent system decision process exposed in the previous section. In this example, each vehicle has got its own multi-agent process for decision making.

3.2 Applying Reactive Agent Decision Process to Platoon Control

Platoon system could be considered as a trajectory following problem. By contrast to classical methods, the trajectory to follow is defined by the leader vehicle instead of being known in advance.

Building the Environment The agents' environment is built using data from the problem. In the platoon control context, the data is composed of the position of the leader, relatively to the considered follower vehicle, and its path. The agents' environment is then considered as a 2D space where the leader trajectory is translated into an attractive area and where the other region of the vehicle space are set to be repulsive (see Figure 2).

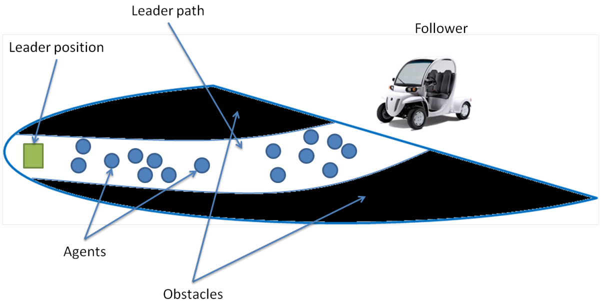


Fig. 2. Translation of the problem data into the agents' environment.

Interaction model Interactions in agents' environment can be summarized as shown in Figure 3.

- **Agent-Leader representative interaction (Attraction)** : The attraction force generated by the leader representative is computed as a linear force defined by:
$$\boldsymbol{F} = \beta_g m \boldsymbol{A_i T} \tag{1}$$
- **Agent-Obstacles interaction (Repulsion)** : The repulsion is made by classical newtonian force in $\frac{1}{r^2}$.

$$\begin{cases} Fo_i^X = \sum_o \left(\Delta_o \cdot m \cdot m_o \frac{(x_i - x_o)}{((y_i - y_o)^2 + (x_i - x_o)^2)^{\frac{3}{2}}} \right) \\ Fo_i^Y = \sum_o \left(\Delta_o \cdot m \cdot m_o \frac{(y_i - y_o)}{((y_i - y_o)^2 + (x_i - x_o)^2)^{\frac{3}{2}}} \right) \end{cases} \tag{2}$$

- **Agent-Agent interaction (Repulsion)** : This interaction shares the same formulation as agent-obstacle repulsion. The repulsion between agents is generally introduced to ensure a homogeneous exploration of the environment avoiding false agents grouping and local minima locking. This repulsion If A_i and A_j are two agents located in P_i and P_j, the repulsion force is given by:

$$\boldsymbol{Fr_{ij}} = \alpha m_i m_j \frac{\boldsymbol{P_i.P_j}}{\|\boldsymbol{P_i.P_j}\|^3} \tag{3}$$

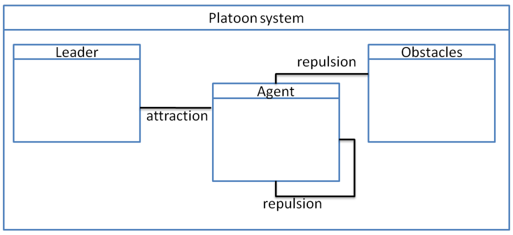


Fig. 3. Interaction diagram.

Making the Decision As previously explained, the decision making process is based on agent's population observation. So as to select the tuple (steering angle, speed) which correspond to a valid order for the vehicle low level controller, one makes an observation of the agency based on topological indicators. Two vectors are computed. The first, called P_{dir} is the vector which starts from the representative of the follower vehicle (generally taken as the origin of the agents' environment) and finishes to the mean position of the agents. The second denoted P_{speed} is composed of the mean speed of all agents. A combination of these two vectors leads to a set point to sent to the vehicle and denoted P_{order} (see Figure 4).

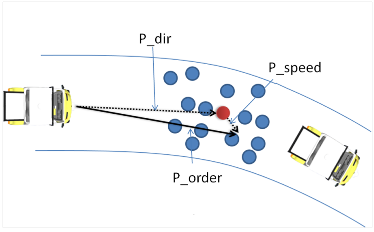


Fig. 4. Decision making.

3.3 Results

So as to test the approach, simulations have been performed considering lateral and longitudinal distances.

Lateral Distance Lateral distance measures the distance between leader and follower trajectories. This distance should always be nearest as possible the setpoint. Figure 5 (top) shows the distance error. Taking account time stabilization, the maximal error is around 0.15 meter.

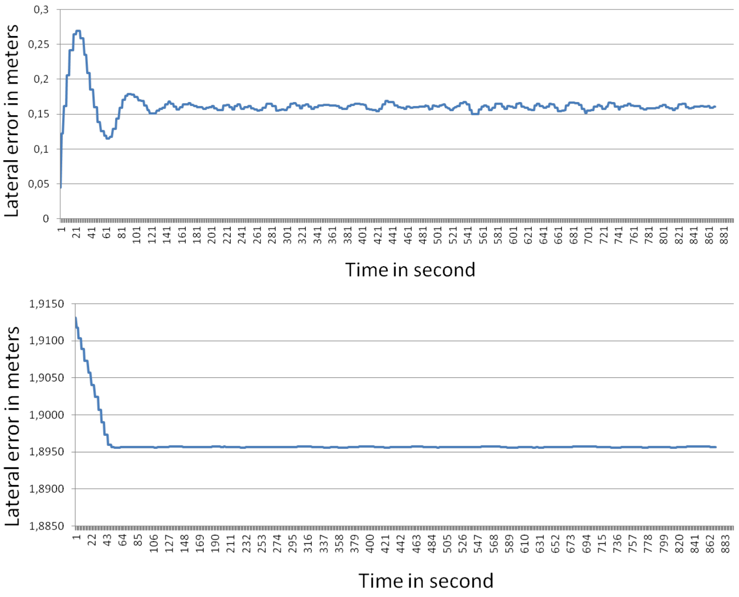


Fig. 5. Lateral distance error (top) and longitudinal distance (bottom).

Longitudinal Distance Longitudinal distance consists of mesuring distance between tow vehicles. This distance should always be above a security distance and nearest as possible the setpoint. Security distance is set to 1 meter. Figure 5 (bottom) shows this distance during simulation. Longitudinal distance is always over security distance and oscillations are mastered.

4 Conclusion and Perspectives

Using multi-agent system as a decision process is an approach which can be considered as a good alternative to classical decision process methods. Indeed, it is able to provide interesting properties such as adaptability and robustness and can deal by nature with dynamical problems. By contrast, due to its novelty, this approach is harder to use. The principal difficulties in conception are the translation of the system states space into a suitable environment that can be travelled by agents and the projection function which evaluates agency emergent properties and translates them into decisions.

Its application to the platoon issue contrasts with the classical agent methods which consider each vehicle as an agent. In this example, each vehicle uses a system of agents which allows to make a suitable decision taking into account problem constraints into an abstract agent's environment and chossing one action using agency properties evaluation. The other first examples found in literature give also some good results. For the moment, they are limited to situated problems where problem dimension fit to the dimension of the decision space. It seems now interesting to perform further research in this field so as to develop rigorous conception methodologies and to test this new kind of approaches to other application fields.

References

1. Avanzini, P., Thuilot, B., Martinet, P.: A control strategy taking advantage of inter-vehicle communication for platooning navigation in urban environment. In: Workshop "Perception and Navigation for Autonomous Vehicles in Human Environment" at IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). San Francisco (USA) (2011)
2. Bautin, A., Simonin, O., Charpillet, F.: Stratégie d'exploration multirobot fondée sur les champs de potentiels artificiels. Revue d'Intelligence Artificielle 26(5), 523–542 (2012)
3. Bourjot, C., Chevrier, V., Thomas, V.: How social spiders inspired an approach to region detection. AAMAS pp. 426–433 (2002)
4. Burrafato, P., Cossentino, M.: Designing a multi-agent solution for a bookstore with the passi methodology. In: AOIS@CAiSE (2002)
5. Contet, J.M., Gechter, F., Gruer, P., Koukam, A.: Reactive multi-agent approach to local platoon control: stability analysis and experimentations. International Journal of Intelligent Systems Technologies And Application. (2011)
6. Contet, J.m., Gechter, F., Lefoulon, L.: Autonomous emergent painting triggered by music and motion. International Journal of Arts and Technology (2013)
7. Dafflon, B., Contet, J.m., Gechter, F., Gruer, P.: Toward a reactive agent based parking assistance system. In: International Conference on Tools with Artificial Intelligence (2012)
8. Daviet, P., Parent, M.: Longitudinal and lateral servoing of vehicles in a platoon. IEEE Intelligent Vehicles Symposium, Proceedings pp. 41 – 46 (1996)
9. De Wolf, T., Holvoet, T.: Emergence versus self-organisation: different concepts but promising when combined. Engineering Self Organising Systems: Methodologies and Applications, Lecture Notes in Computer Science 3464, 1–15 (2005)
10. Dessalles, J.L., Ferber, J., Phan, D.: Emergence in Agent based Computational Social Science: Conceptual, Formal and Diagrammatic Analysis. In: Intelligent Complex Adaptative Systems, p. 24 (2008)

11. DiMarzo-Serugendo, G., Karageorgos, A., Rana, O., Zambonelli, F.: Engineering Self-Organising Systems: Nature-Inspired Approaches to Software Engineering. Lecture notes in Atificial intelligence, number 2977 ISBN-3540212019 (2004)
12. Galland, S., Gaud, N., Demange, J., Koukam, A.: Environment model for multiagent-based simulation of 3D urban systems. In: the 7th European Workshop on Multiagent Systems (EUMAS09). Ayia Napa, Cyprus (2009)
13. Gechter, F., Contet, J.m., Gruer, P., Koukam, A.: Car-driving assistance using organization measurement of reactive multi-agent system. Procedia Computer Science CS pp. 317–325 (2010)
14. Lafferriere, G., Caughman, J., Williams, A.: Graph theoretic methods in the stability of vehicle formations. In: American Control Conference. vol. 4, pp. 3729–3734 (2004)
15. Lozenguez, G., Adouane, L., Beynier, A., Mouaddib, A.I., Martinet, P.: Calcul distribué de politiques d'exploration pour une flotte de robots mobiles. In: JFSMA. pp. 117–126 (2011)
16. Matsuda, S.: A neural network model for the decision-making process based on anp. In: Neural Networks, 2006. IJCNN '06. International Joint Conference on. pp. 1421–1426 (2006)
17. Moncion, T., Amar, P., Hutzler, G.: Automatic characterization of emergent phenomena in complex systems. Journal of Biological Physics and Chemistry 10, 16–23 (2010)
18. Mouad, M., Adouane, L., Schmitt, P., Khadraoui, D., Martinet, P.: Mas2car architecture - multi-agent system to control and coordinate teamworking robots. In: ICINCO (2). pp. 451–456 (2011)
19. Naim, S., Hagras, H.: A hybrid approach for multi-criteria group decision making based on interval type-2 fuzzy logic and intuitionistic fuzzy evaluation. In: Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference on. pp. 1–8 (2012)
20. Russell, S.J., Norvig, P.: Artificial Intelligence - A Modern Approach (3. internat. ed.). Pearson Education (2010)
21. Schneider, D.K.: Modlisation de la dmarche du dcideur politique dans la perspective de l'intelligence artificielle. Thse de l'Unit Technologie de Formation et Apprentissage - Facult de Psychologie et des Sciences de l'Education - Universit de Genve (1994)
22. Simonin, O., Gechter, F.: An environment-based methodology to design reactive multi-agent systems for problem solving. In: (Lecture Notes in Artificial Intelligence) LNAI Vol. 3830 : Environment for Multiagent Systems II (revised and selected papers of Environment for Multi Agent Systems (E4MAS) 2005). pp. 32–49. Springer-Verlag Berlin Heidelberg (2006)
23. Weyns, D., Parunak, H.V.D., Michel, F., Holvoet, T., Ferber, J.: Environments for multiagent systems state-of-the-art and research challenges. In: E4MAS. pp. 1–47 (2004)
24. Yang, S., Gechter, F., Koukam, A.: Application of reactive multi-agent system to vehicle collision avoidance. In: The 20th IEEE International Conference on Tools with Artificial Intelligence ICTAI, Dayton, Ohio, USA. pp. 197–204. IEEE Computer Society (2008)
25. Zhao, P., Suryanarayanan, S., Simoes, M.: An energy management system for building structures using a multi-agent decision-making control methodology. In: Industry Applications Society Annual Meeting (IAS), 2010 IEEE. pp. 1–8 (2010)
26. Zhou, Y., Ma, L., Mathew, J., Sun, Y., Wolff, R.: Maintenance decision-making using a continuous-state partially observable semi-markov decision process. In: Prognostics and Health Management Conference, 2010. PHM '10. pp. 1–8 (2010)

Mobile ACORoute: Route Recommendation Based on Communication by Pheromones

Carla S. G. Pires, Marilton S. de Aguiar, and Paulo R. Ferreira

Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas (UFPEL), Brazil

Abstract. Urban mobility problems affects the vast majority of cities nowadays. Thus, systems that provide real time information to assist in planning routes and choosing the most appropriate paths are essential to make transport more effective. As an alternative solution to problems related to mobility in cities, there are the so-called Intelligent Transportation Systems (ITS) which include the Route Recommendation Systems (RRS) and methodologies for congestion prediction that combine Information and Communication Technology (ICT) with Artificial Intelligence (AI) technonology to improve the quality of transport systems. In this context, this work proposes the use of pheromone-based communication for building an ITS that offers information about real time traffic flow, taking into account the mobility of vehicles and passengers and the traffic dynamics. The general goal is to provide an Android solution able to suggest users routes calculated by the hybrid algorithm between A* and pheromone mechanism. The idea is to avoid areas of heavy traffic congestion.

Keywords: Route recommendation systems, intelligent transportation systems, pheromone based communication.

1 Introduction

In the past decades, the traffic in medium and large cities, as well as the incovenience caused directly or indirectly by it, cause increasing mobility problems. The Intelligent Transportation Systems (ITS) are shown as an alternative to improve mobility within cities through the application of Information and Communication Technology (ICT) to support the existing traffic infrastructure and improve the quality of transport systems [17].

A wide variety of ITS tools have played important roles in the effectiveness of transport. These systems provide information related to traffic, influencing in various aspects of transport in relation to urban mobility. Most of these ITS tools uses static information aided by the traffic infrastructure integrated technologies [5]. This article proposes to use information from mobile devices to dynamically determine the best path for the driver seeking to avoid traffic congestion.

To this end, this paper proposes an approach for the calculation of the trajectory based on the use of pheromone dynamics. Other approaches to congestion prediction using Swarm Intelligence have been propose [10,11,12]. Previous works differ in how pheromones influence the calculation process and do not consider some aspects of their practical application.

The paper is arranged as follows: in Section 2 the theoretical background are briefly introduced; Section 3 discusses the conceptual and technological foundations of the work; Section 4 discusses about the development of ACORoute and shows how the proposed approach was implemented; 5 presents the results obtained from the application of the proposed approach in a simulation environment and, finally, Section 6 presents the conclusions of the work.

2 Related Works

2.1 Intelligent Transportation Systems (ITS)

The term ITS emerged in USA in the late 1980s with a movement aiming to make transport safer, more effective and reliable. The ideia of the ITS is to support the existing traffic infrastructure without the need to change it [1]. Thus, a number of tools have been developed in order to assist drivers to stay informed about the traffic conditions and the relevant aspects of the dynamics of its flow. Areas of Computing as Artificial Intelligence (AI), Computer Vision, Pattern Recognition, Machine Learning, Data Mining, and Intelligent Control have been intensively explored in this process [14].

The tools developed in ITS enable users to get several kinds of information related to traffic. This information range from current location, alternative routes, road conditions, even weather forecasts along the route, providing greater power of decision on the actions and choices of individuals.

In a simplified way one can say that the ITS aim at the optimization of existing transport systems by making use of a wide range of tools that combine technologies and improvements in information systems, communications, sensors and advanced mathematical methods. The objective is to obtain roads, vehicles and more "intelligent" users, with an attempt to facilitate the flow of traffic, and solve a variety of transportation problems of our days, such as congestion, safety and environmental problems [16].

2.2 Swarm Intelligence

An interesting approach is the use of techniques of swarm intelligence, as the organization of bee colonies and allocation of tasks among insect societies. The dynamics of insect society are the result of different actions between the population and the environment. This interaction between agents creates a system of communication that contributes to the formation of "Collective Intelligence" assigned to the insect society. This technique arose from observations and studies of the behavior of living beings usually searching for food [4].

Swarm intelligence or collective intelligence has been successfully applied in dynamic optimization problems in various fields, such as the traveling salesman problem, quadratic assignment problems, load balancing and vehicle routing problems. These are just some examples where swarm intelligence is applied [15].

Examples of applications of this nature can be seen in [2], which introduced systems of bees as a new approach in the area of intelligent swarms applied to problems of transportation engineering. They developed a new heuristic for the traveling salesman problem by defining an artificial environment of bees; swarm intelligence technique inspired by the behavior of ant colonies are also used to optimize the timing of traffic lights [6]; and, the use of the swarm intelligence inspired by the behavior of bees society for task allocation using clustering, grouping agents by skills, considering societies of bees and how they collect the best nectar from the available sources through simple rules of behavior [13].

The use of intelligent insect swarms has already been applied in problems to find shorter routes between cities (TSP), such as [9], who developed a new heuristic for the traveling salesman problem by setting an artificial environment of bees, where each bee is an agent performing activities defined by the model and the communication occurs in the interaction between them [8].

This work has focused on ant colony optimization (ACO) that is inspired by the observation of ant communities and their organization to find food sources [3].

Experiments were conducted to understand this behavior, and showed that ants have the ability to discover the shortest path very quickly. It was observed that only the first ants randomly choose the path, thus those who choose the shorter path arrive faster to the nest. Thus, the probability of choosing the shortest way increases continuously and rapidly all ants start using the shortest path [15].

The ACO is based on this behavior, considering that each ant walking on a trail deposits certain amount of pheromone in it, then the next ants follow the path with a proportional probability to the amount of pheromone present in this path and thus reinforcing the current pheromone [3].

ACO was the first swarm intelligence algorithm to be developed, it is used in the development of this work. It has been effectively applied to solve the traveling salesman problem, as well as several other problems related to Transportation and Traffic Engineering. [7] and [11] used the swarm intelligence technique inspired in the behavior of ants society (ACO). They employed this methodology for predicting traffic congestion in a simulated environment, considering agents (sensors) installed at the intersections of routes that manage information of pheromones. [2] evaluated the application of pheromone communication in real traffic conditions applied to congestion prediction and observed that the method is effective for short-term predictions.

3 Mechanism for Congestion Identification Based on Communication by Pheromones

Technologies for congestion identification are a key element to support Intelligent Transportation Systems. Currently, many methods have been proposed, many of them using collective intelligence, based on the idea that societies of insects perform complex tasks using decentralized communication based on pheromones. In this context, pheromones are considered as a means to provide information [2,7,11,12,18] among others.

in this paper we are proposing congestion identification using the technique of communication based on pheromones (ACORoute), which was also proposed by [2], [7] and [11] in their works. Different from the adopted in our work, they use the infrastructure provided by local transport system and/or devices installed in vehicles to collect, process and store data related to traffic. In addition, when calculating the route, ACORoute uses the pheromone information in heuristic of the best route, while the papers presented use a common navigator to calculate, merely use pheromones for congestion prediction. A comparison between relevant works can be seen in Table 1.

when dealing to congestion prediction, the application WAZE should be mentioned, which recommends routes avoiding places of heavy flow. However, it only offers the collaboration of other users who interact with the application to report occurrences. Thus, apart from Waze (2013), which also makes congestion prediction, other works have results only in simulated environment to date to validate their technical environments. There are no real applications.

4 ACORoute

4.1 Pheromone-Based Model

The use of pheromone dynamics for congestion identification was discussed in the development of ACORoute, and it was proposed a model that predicts congestion based in the pheromones communication mechanism. The model used considers vehicles as insects that deposit pheromones.

Table 1: Characteristics of the Method for Congestion Prediction. Where *Ph* stands for pheromone.

	Self-organising	Pheromone	Traffic-Congestion	Hybrid ACO	WAZE	ACORoute
	Narzt (2010)	Ando (2006)	Kurihara (2013)	Ochiai (2014)	WAZE (2013)	Us
Algorithm	–	Ph	–	Dijkstra	A*	A*/Ph
Context data	No	No	No	No	No	Yes
Mobile	No	No	No	No	Yes	Yes
Sources of information	No	Yes	Yes	Yes	Yes	Yes
Traffic infrastructure	Yes	Yes	Yes	No	No	No
User preferences	No	No	No	Yes	No	Yes
Historical data	No	No	No	No	Yes	Yes
Real Time	No	No	Yes	No	No	Yes
route re-planning	Yes	No	No	No	No	Yes

The operating principle of this strategy occurs with vehicles that mark their path by dropping digital pheromones which are perceived by all the vehicles that travel in the environment. This mechanism is used to allow the calculation of a route that avoids heavy traffic.

The vehicle sends information about geopositioning to the server at every 30 seconds and with this information is created of the pheromone map, that is built as follows: when the georeferencing information is sent by the application it is received by the WebService, the value of 0.8 is incremented at the referenced node (Latitude and Longitude).

When a given node remains without receiving information from pheromones longer than 30 seconds, the evaporation process starts until the amount of pheromones in the node reaches 0. The value of the decrement to pheromone is 0.3. These values were calibrated through simulation, which confirmed to be the most effective.

4.2 The Best Route Algorithm

To determine a route between a source node (latitude and longitude) and a destination node (latitude and longitude), the solution presented here uses a variation of the routing heuristic in graphs A*, considering besides the distance, the amount of pheromones in this route. The proposed algorithim prefers routes with lower level of pheromones

The map stored in the database is represented by a directed graph with weights representing distances. We can say that the graph is a logic representation of the map, containing the streets of the city of Pelotas, each section of the street is considered an edge and the points are nodes. The search algorithm A*, that considers the pheromone information to calculate the route, is used to find the best route between two points. The database used is derived from the open database of the Open Street Maps only considering the points inside the area of Pelotas.

4.3 Android Application

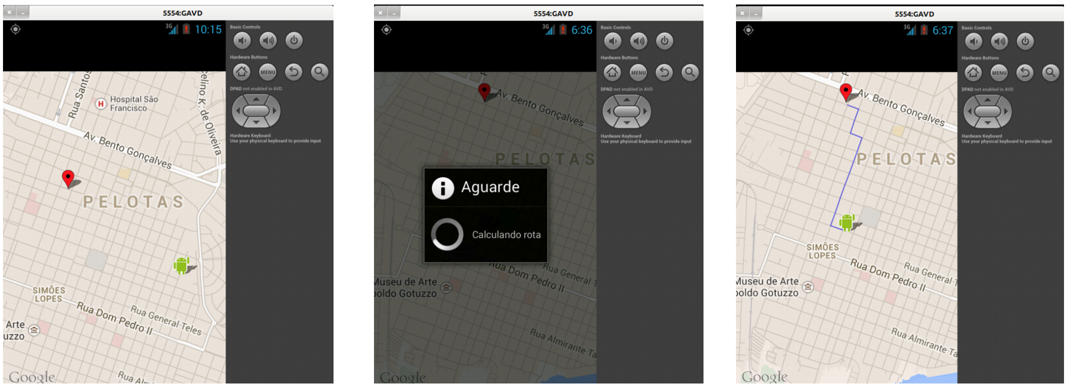
For the development of the application, the Eclipse IDE (Integrated Development Environment) combined with Android SDK (Software Develpoment Kit), that provides an API (Application Programming Interface) required to build and to test applications for Android was used.

Two languages were used for the implementation: XML and Java. XML is used to build the graphical part of the application, where the interaction with the user happens, made through the components of the graphic application, the widgets, such as buttons and textboxes. In the control, the Android-Java language, a standard Java subset, which implements specific API for Android application development was used.

In the Android API Activity and Service concepts that are related to the graphical interface of the application are defined which are responsible for the different screens, operations performed in the background, among them communication with the server.

The implementation project of the application was divided into two stages: i) receipt and transmission of georeferenced data, used to establish the historical base and the construction of the pheromone map, these information are obtained through GPS (Global Positioning System) from the device itself and ii) development of the interface with the user where information such as location and path of the calculated routes will be presented.

A service responsible for connecting the application to the GPS, receiving location information periodically every 30 seconds, which can be set remotely from the WebService was created in the first stage of the work. After receiving the information, Service sends it to the WebService, keeping the historical database and pheromones. The application prototype interface is shown in Figures 1a, 1b and 1c.



Aguarde

Calculando rota

(a) Origin and destination points

(b) Calculating the route

(c) Route view

Fig. 1: Prototype Interface.

The time to update the users location was defined in order to optimize the battery consumption of the device without sacrificing the accuracy of the application. When using smaller and more frequent values, it creates several accesses to the GPS which is not considered a good strategy for the substancial increase of battery consumption of the device. In contrast, using higher values can lead to a possible loss of information because it becomes complex to deduce the route taken by the user.

In the second stage of development, two activities were created, the first one, responsible for controlling the transmission of location information and for opening the map; and the second, responsible for displaying the map and all the relevant information, such as location of the user, menus and routes. The map chosen for the application was Google Maps, because it is easy to use and complete, in addition to having its own API for Android programming, which makes the implementation simpler.

5 Numerical Experiments and Discussions

After the congestion identification method was defined, it was necessary to evaluate it in terms of precision and, consequently, the viability of the method. To that end, simulations comparing the performance of the proposed method with the algorithm A*, standard technique for the route calculation, were made. In a simulation environment developed in Netlogo, 2 (two) cars were inserted, one of them using the methodology that uses pheromones when calculating the route and the other one using pure A* algorithm. In addition, experiments were performed for different amounts of cars of random behaviour for the evaluation in distinct congestion situations (as shown in Table 2). The parameterization was made according to [10] that also used ACO for congestion prediction, where values of 0.7 and 0.8 for evaporation and deposit respectively were pointed as good.

After 6000 simulations of the model, it was possible to observe that the average time (in *ticks* of simulation) of completion of route, when considering the weight of the pheromone in the cost of the Heuristic A* is lower if compared to the route without considering the pheromone for all congestion situations.

It can be observed that the number of cars used in the simulations influence the average times obtained for calculating the best route. As the number of cars increases, the average time for the calculation also increases, thus, the longer times were obtained when 600 cars were used. The average times presented in Table 2 shows that in all simulations the time of the route using pheromones were lower in relation to the simulations not using pheromones. It could be observed in the simulations that situations where the congestion rate is high, to seek for alternative routes may lead to other congestion situations, which can be worse than waiting for the normalization of the flow. Thus, it was verified that the proposed method is effective.

Subsequent experiments for calibration of the parameters were made, such as increment and decrement of pheromones, weight of the pheromones, weight of the distance to cover all the way, evaporation of pheromone, pheromones limit. Also, experiments were performed to verify the minimum percentage of the number of cars depositing pheromones that makes the model more effective.

The methodology of the tests was systematized in order to compose the best set of parameters for the model. The systematization and planning of experiments was necessary to make possible the evaluation of the behavior of the evaluated parameters. In this sense, an analysis of variance was performed using for this, the full factorial design.

Were analyzed to each experiment 2 factors at 3 levels, so we will have a planning where all parameters are studied in three levels thus have a factorial design 2^3 with a planning matrix with 9 experiments.

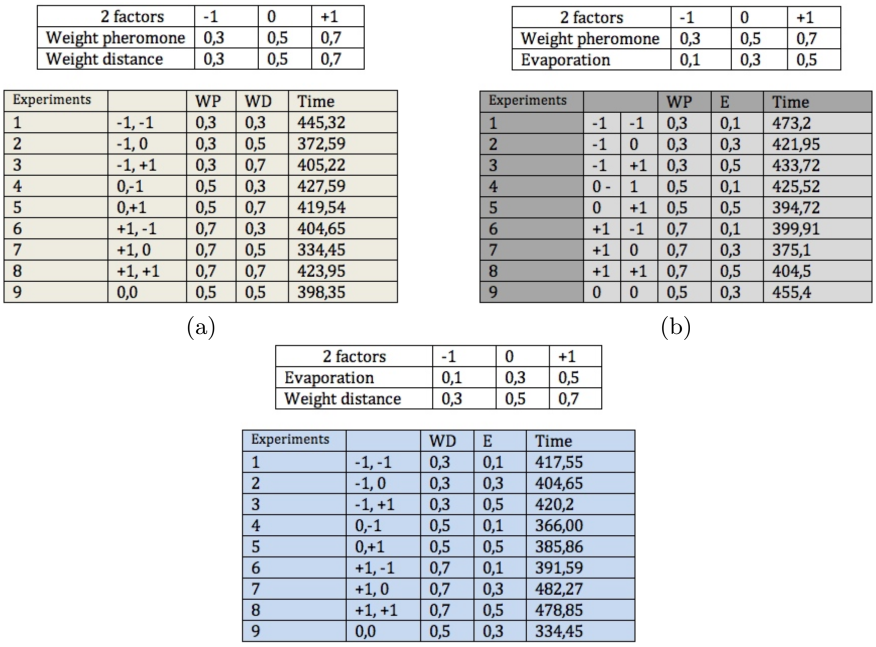
For planning matrices presented in Figure 2 the following response surfaces were obtained.

Considering the data of the correlation between weight of pheromone and weight of distance, was obtained on analysis of variance in Table 3 , where we can observe that the values of p are smaller than 0.05, this means that factors are considered significant, ie, both factors are relevant in the media presented.

Observing the results presented in Table 3, it is possible to identify that the best results were obtained with the parameterization Weight of the Pheromone 0.7, Weight of Distance 0.5 and Evaporation 0.3. Also, it can be observed that the highest average times were obtained when the decrements used for Evaporation were higher. This behavior can be attributed to the fact that the algorithm considers without congestion a stretch that is still crowded, and calculate a route that leads the vehicle to get congested rather than calculate a route that avoids a particular stretch. Thus, this parameterization was used in the other simulations.

Table 2: Efficiency of the pheromone model.

	Number of cars	With pheromones	Without pheromones
	200	**368.37**	426.91
Average time (*ticks*)	400	**488.48**	542.52
	600	**491.14**	572.81



2 factors	-1	0	+1
Weight pheromone	0,3	0,5	0,7
Weight distance	0,3	0,5	0,7

Experiments		WP	WD	Time
1	-1, -1	0,3	0,3	445,32
2	-1, 0	0,3	0,5	372,59
3	-1, +1	0,3	0,7	405,22
4	0,-1	0,5	0,3	427,59
5	0,+1	0,5	0,7	419,54
6	+1, -1	0,7	0,3	404,65
7	+1, 0	0,7	0,5	334,45
8	+1, +1	0,7	0,7	423,95
9	0,0	0,5	0,5	398,35

(a)

2 factors	-1	0	+1
Weight pheromone	0,3	0,5	0,7
Evaporation	0,1	0,3	0,5

Experiments			WP	E	Time
1	-1	-1	0,3	0,1	473,2
2	-1	0	0,3	0,3	421,95
3	-1	+1	0,3	0,5	433,72
4	0 -	1	0,5	0,1	425,52
5	0	+1	0,5	0,5	394,72
6	+1	-1	0,7	0,1	399,91
7	+1	0	0,7	0,3	375,1
8	+1	+1	0,7	0,5	404,5
9	0	0	0,5	0,3	455,4

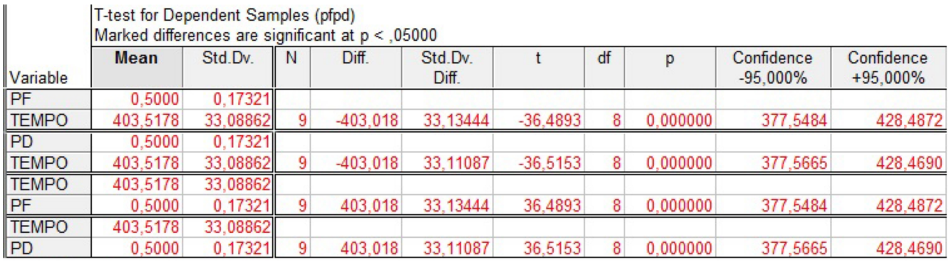
(b)

2 factors	-1	0	+1
Evaporation	0,1	0,3	0,5
Weight distance	0,3	0,5	0,7

Experiments		WD	E	Time
1	-1, -1	0,3	0,1	417,55
2	-1, 0	0,3	0,3	404,65
3	-1, +1	0,3	0,5	420,2
4	0,-1	0,5	0,1	366,00
5	0,+1	0,5	0,5	385,86
6	+1, -1	0,7	0,1	391,59
7	+1, 0	0,7	0,3	482,27
8	+1, +1	0,7	0,5	478,85
9	0,0	0,5	0,3	334,45

(c)

Fig. 2: Planning matrix: (a) Weight pheromone x Weight distance; (b) Weight pheromone x Evaporation; (c) Weight distance x Evaporation.



T-test for Dependent Samples (pfpd)
Marked differences are significant at p < ,05000

Variable	Mean	Std.Dv.	N	Diff.	Std.Dv. Diff.	t	df	p	Confidence -95,000%	Confidence +95,000%
PF	0,5000	0,17321								
TEMPO	403,5178	33,08862	9	-403,018	33,13444	-36,4893	8	0,000000	377,5484	428,4872
PD	0,5000	0,17321								
TEMPO	403,5178	33,08862	9	-403,018	33,11087	-36,5153	8	0,000000	377,5665	428,4690
TEMPO	403,5178	33,08862								
PF	0,5000	0,17321	9	403,018	33,13444	36,4893	8	0,000000	377,5484	428,4872
TEMPO	403,5178	33,08862								
PD	0,5000	0,17321	9	403,018	33,11087	36,5153	8	0,000000	377,5665	428,4690

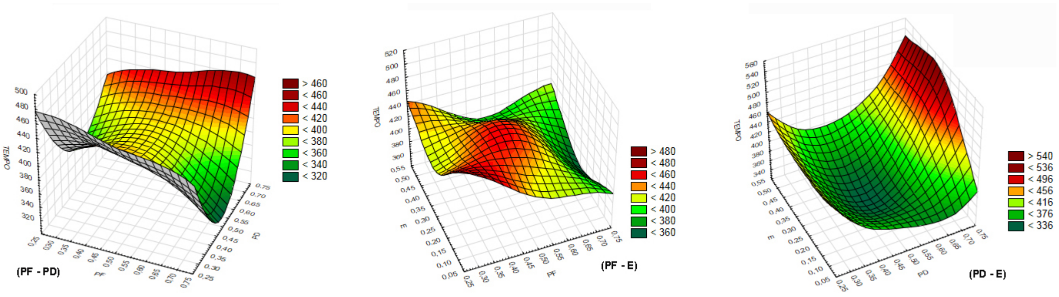
Fig. 3: Student t-test values.

Experiments with the inclusion of route recalculation were made once the parameterization was defined. The recalculation occurs in situations where the vehicle remains motionless for a certain period. These tests aimed to identify the effectiveness of including recalculation and also calibrate the parameter called Recalculation Limit, which determines the time limit for the route recalculation. The results are displayed in Table 4.

Observing Table 4 can be verified that the time limit $1,0$ Ticks to recalculate the route presented better results.

Table 3: Analysis of parameterization with 600 cars on the environment, 100% with pheromones.

Parameterization for 6000 performances			Average time (Ticks)	
Weight pheromone	Weight distance	Evaporation	W/ pheromone	W/o pheromone
0,5	0,5	0,3	**455,4**±14,41	398,35±4,64
0,5	0,5	0,1	**425,52**±8,31	418,09±4,14
0,5	0,5	0,5	**394,72**±4,65	408,86±6,23
0,5	0,3	0,3	**427,59**±3,54	480,36±5,45
0,5	0,7	0,3	**419,54**±4,67	460,09±4,41
0,5	0,5	0,3	**398,35**±9,24	455,4±7,86
0,7	0,5	0,1	**366,00**±2,44	399,91±4,34
0,7	0,5	0,5	**385,86**±3,18	404,05±6,23
0,7	0,7	0,1	**391,59**±7,45	399,27±5,87
0,7	**0,5**	**0,3**	**334,45±2,14**	**375,10±2,87**
0,7	0,3	0,3	**404,65**±4,67	440,35±6,14
0,7	0,3	0,1	**417,55**±2,98	430,3±4,32
0,3	0,3	0,3	**445,32**±3,23	479,18±5,90
0,3	0,5	0,3	**372,59**±2,05	421,95±2,97
0,3	0,7	0,3	**405,22**±1,65	459,82±4,03
0,3	0,5	0,1	**473,2**±4,26	445,25±8,90
0,3	0,5	0,3	**421,95**±3,89	372,59±7,98
0,3	0,5	0,5	**433,72**±6,75	392,72±5,73



(a) (b) (c)

Fig. 4: Response surface graph: (a) Weight pheromone x Weight distance; (b) Weight pheromone x Evaporation; (c) Weight distance x Evaporation.

Table 4: Analysis of decision parameters for the route recalculation.

600 cars in the environment	Average time (*Ticks*)	
Recalculation limit (*Ticks*)	With pheromones	Without pheromones
1.5	**369.87**±3.52	393.15±2.63
1.0	**362.15±4.58**	**399.54±2.41**
0.5	**379.28**±5.96	381.90±2.06

6 Conclusions and Future Work

This work has as main contribution: the development of a methodology to predict congestion using pheromone-based communication. The results obtained in validation tests of the methodology through simulation and implementation of mobile device (ACORoute) show its viability at gaining time with its use.

The results obtained in the simulations confirm the applicability and effectiveness of the proposed method in predicting congestion, obtaining better results in time when compared to the non-use of the technique.

In general, it can be concluded that:

- The average times using pheromones are better in all cases.
- With the increase of the number of cars, the averages increase, yet the results were better using pheromones.
- With the introduction of the recalculation, the average was better.
- The application is in operation and it calculates the route avoiding locations with congestions.
- The application still needs to be optimized in some aspects.

Among the aspects raised to continue the work, we need to improve the following:

- Minimize the battery consumption;
- Testing in a real environment;
- Determine the pattern of behaviour of routes based on historical data, for situations where there is no information of pheromones;
- Implement the use of historical information in route calculation;
- Optimization of route calculation, improving the response time;
- Implement ACORoute for other platforms;
- Make download of the application available to the academic community.

References

1. Adler, J.L., Blue, V.J.: Toward the design of intelligent traveler information systems. Transportation Research Part C: Emerging Technologies 6(3), 157 – 172 (1998)
2. Ando, Y., Fukazawa, Y., Masutani, O., Iwasaki, H., Honiden, S.: Performance of pheromone model for predicting traffic congestion. In: Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems. pp. 73–80. AAMAS '06, ACM, New York, NY, USA (2006)
3. Barbosa, H.J. (ed.): Ant colony optimization - techniques and applications. InTech Chapters published, Croatia (2013)
4. Blum, C., Merkle, D.: Swarm Intelligence: Introduction and Applications. Springer Publishing Company, Incorporated, 1 edn. (2008)

5. Ferris, B., Watkins, K., Borning, A.: Location-aware tools for improving public transit usability. Pervasive Computing, IEEE 9(1), 13 –19 (2010)
6. Hoar, R., Penner, J., Jacob, C.: Evolutionary swarm traffic: if ant roads had traffic lights. In: Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on. vol. 2, pp. 1910–1915 (2002)
7. Kurihara, S.: Traffic-congestion forecasting algorithm based on pheromone communication model. Ant Colony Optimization - Techniques and Applications 104, 167 – 175 (2013)
8. Lucic, P., Teodorovic, D.: Transportation modeling: An artificial life approach. In: Proceedings of the 14th IEEE International Conference on Tools with Artificial Intelligence. pp. 216–. ICTAI '02, IEEE Computer Society, Washington, DC, USA (2002)
9. Lucic, P., Teodorovic, D.: Computing with bees: Attacking complex transportation engineering problems. International Journal on Artificial Intelligence Tools 12(03), 375–394 (2003)
10. Masutani, O., Ando, Y., Sasaki, H., Iwasaki, H., Fukazawa, Y., Honiden, S.: Pheromone model: Application to traffic congestion prediction. In: Brueckner, S., Marzo Serugendo, G., Hales, D., Zambonelli, F. (eds.) Engineering Self-Organising Systems. Lecture Notes in Computer Science, vol. 3910, pp. 182–196. Springer Berlin Heidelberg (2006)
11. Narzt, W., Wilflingseder, U., Pomberger, G., Kolb, D., Hrtner, H.: Self-organising congestion evasion strategies using ant-based pheromones. Iet Intelligent Transport Systems 4 (2010)
12. Ochiai, J., Kanoh, H.: Hybrid ant colony optimization for real-world delivery problems based on real time and predicted traffic in wide area road network. Fourth International conference on Computer Science and Information Technology - CCSIT 2014 4(2) (2014)
13. dos Santos, D.S., Bazzan, A.L.: Distributed clustering for group formation and task allocation in multiagent systems: A swarm intelligence approach. Applied Soft Computing 12(8), 2123 – 2131 (2012)
14. Steinmacher, I., Vieira, V., Salgado, A.C., Tedesco, P., Times, V., Ferraz, C., Huzita, E., Chaves, A.P.: The ubibus project: Using context and ubiquitous computing to build advanced public transportation systems to support bus passengers. VIII Simpsio Brasileiro de Sistemas de Informao (2012)
15. Teodorovic, D.: Swarm intelligence systems for transportation engineering: Principles and applications. Transportation Research Part C: Emerging Technologies 16(6), 651 – 667 (2008)
16. Tito, A., Borgiani, F., dos Santos, R., Tedesco, P., Salgado, A.: Contextual information in user information systems in public transportation: A systematic review. In: Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on. pp. 361 –366 (2012)
17. Wahle, J., Annen, O., Schuster, C., Neubert, L., Schreckenberg, M.: A dynamic route guidance system based on real traffic data. European Journal of Operational Research 131(2), 302 – 308 (2001)
18. Waze: Waze mobile. `https://www.waze.com/wiki/How_Waze_calculates_routes` (2013)

Reviewing Committee

Alexander Gelbukh
Felix Castro Espinoza
Francisco Viveros Jiménez
Grigori Sidorov
Gustavo Arroyo Figueroa
Hugo Terashima Marín
Ildar Batyrshin
Jesús González Bernal
Luis Villaseñor Pineda
Maya Carillo Ruiz
Miguel González Mendoza
Noé Alejandro Castro Sánchez
Obdulia Pichardo Lagunas
Omar Montaño Rivas
Oscar Herrera Alcantara
Rafael Murrieta Cid
Sabino Miranda Jiménez
Sofía Galicia Haro

Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
Noviembre de 2014
Printing 500 / Edición 500 ejemplares