ISSN 2395-8618

Editorial

SINCE this issue we welcome to our editorial team a new associate editor, Prof. Ramón Silva Ortigoza, an expert in control of mechatronic systems, control of mobile robots, control in power electronics, and geometric optics, who has authored a number of books and over 30 research papers in these areas.

This issue of Polibits includes ten papers by authors from ten different countries: Colombia, Cuba, India, Italy, Japan, Mexico, Norway, Romania, UK, and USA. The papers included in this issue are devoted to such topics as sensor networks, service robotics, control of mechatronic systems, business process modeling, cross-language information retrieval, unsupervised word sense disambiguation, generation of assessment tests in education, large-scale text classification, knowledge discovery in datasets, and automatic text summarization.

B. Anjum and **C. L. Sabharwal** from **USA** in their paper "Filtering Compromised Environment Sensors Using Autoregressive Hidden Markov Model" propose a simple and computationally inexpensive method of identifying compromised sensors in a sensor network. In their experiments on artificial and real datasets, their model shows very high accuracy on a large sensor network, but, even more importantly, it provides high accuracy when learning from a small sensor networks.

A. Vanzo et al. from **Italy** in their paper "Robust Spoken Language Understanding for House Service Robots" address one of the most important parts of the interface between the end user and service robot, namely, speech recognition module. They describe a robust method for re-ranking the hypothesis generated by the speech recognition module adapted for use in the context of service robotics and specifically suited for recognition of typical commands. They show that their method outperforms general-purpose automatic speech recognition programs.

M. G. Villarreal-Cervantes et al. from **Mexico** in their paper "PC Based Open Control Architecture for Mechatronic Systems" describe a low-cost, flexible, reconfigurable, and versatile open control architecture for mechatronic systems implemented in an ordinary personal computer. This architecture will be useful both for modeling in design of mechatronic systems and for teaching design of mechatronic systems in classroom.

H. Ordoñez et al. from **Colombia** in their paper "Business Process Models Clustering Based on Multimodal Search, K-means, and Cumulative and No-Continuous N-Grams" present a method for indexing, searching, and grouping business processes models in order to facilitate the use of large process repositories. Their method, based on linguistic and behavioral information, outperforms existing approaches in terms of precision and recall.

R. Prasath and **S. Sarkar** from **Norway** and **India** in their paper "Cross-Language Information Retrieval with Incorrect Query Translations" improve the performance of information retrieval under cross-language setting, that is, when the user does not know English well enough to formulate the query for a search engine and thus formulates it in his or her own language (Tamil in their experiments), but the documents to be retrieved are in English. Moreover, they consider the situation when the query cannot be reliably automatically translated into English, which is a common phenomenon due to colloquial language style typically used by the Internet users when searching for information.

S. Torres-Ramos et al. from **Mexico** in their paper "Unsupervised Word Sense Disambiguation Using Alpha-Beta Associative Memories" extends the classical Lesk algorithm for unsupervised word sense disambiguation with a novel word-similarity measure based on alpha-beta associative memory operators. They show that the new algorithm is especially effective for dealing with inflective and derivational forms of words without the need for stemming procedure. This can be especially useful for languages for which no good stemming procedure has been developed. To the best of my knowledge, this is the first use of alpha-beta associative memories as a similarity measure in natural language processing.

D. Popescu Anastasiu et al. from **Romania** in their paper "A Method Based on Genetic Algorithms for Generating Assessment Tests Used for Learning" present a method for automatically generating an optimal sequence of tests, out of a given repository, for use in educational settings for evaluation of the performance of the students on a given topic. The method can be configured by the teacher via the use of specific keywords that describe the topics of interest for the evaluation procedure.

M. G. Sohrab et al. from **Japan** in their paper "IN-DEDUCTIVE and DAG-Tree Approaches for Large-Scale Extreme Multi-label Hierarchical Text Classification" propose a novel method for hierarchical text classification applicable to very large document collections such as Wikipedia. The method is based on large-scale hierarchical inductive learning and deductive classification. The authors evaluate their method on standard evaluation datasets provided as part of the PASCAL Challenge on Large-Scale Hierarchical Text Classification.

ISSN 2395-8618

Jarvin A. Antón-Vargas et al. from **Cuba** and **Mexico** in their paper "Instance Selection to Improve Gamma Classifier" present an improvement for the Gamma classifier used in the pre-processing stage of noise filtering in knowledge discovery in datasets, thus alleviating the problem of the presence of misclassified or non-representative instances in the training data. The authors introduce a novel similarity function for the Gamma classifier. Experimental results are presented on fifteen different datasets.

N. Sanchan et al. from **UK** in their paper "Understanding Human Preferences for Summary Designs in Online Debates Domain" study user preferences in generating summaries for online debates. While for some other domains such as news or scientific papers summarization it is clear what information should be included in a good summary, in many domains, including online debates, this is not clear and needs a separate research. With the help of sixty independent evaluators, the authors show that the best summary types for this domain are chart summary and side-by-side summary. This finding will guide future development of automatic summarization systems for this domain.

This issue of the journal will be useful to researchers, students, and practitioners working in the corresponding areas, as well as to general public interested in advances in computer science, artificial intelligence, and computer engineering.

Alexander Gelbukh

Instituto Politécnico Nacional,
Mexico City, Mexico
Editor-in-Chief

ISSN 2395-8618

Filtering Compromised Environment Sensors Using Autoregressive Hidden Markov Model

Bushra Anjum and Chaman Lal Sabharwal

***Abstract*—We propose a method based on autoregressive hidden Markov models (AR-HMM) for filtering out compromised nodes from a sensor network. We assume that sensors are healthy, self-healing and corrupted whereas each node submits a number of readings. A different AR-HMM (A, B, π) is used to describe each of the three types of nodes. For each node, we train an AR-HMM based on the sensor's readings, and subsequently the B matrices of the trained AR-HMMs are clustered together into two groups: healthy and compromised (both self-healing and corrupted), which permits us to identify the group of healthy sensors. The existing algorithms are centralized and computation intensive. Our approach is a simple, decentralized model to identify compromised nodes at a low computational cost. Simulations using both synthetic and real datasets show greater than 90% accuracy in identifying healthy nodes with ten nodes datasets and as high as 97% accuracy with 500 or more nodes datasets.**

***Index Terms*—Autoregressive hidden Markov models, environment sensing, filtering corrupted nodes, sensor network, clustering, anomaly detection.**

I. Introduction

SENSOR systems have significant potential for aiding scientific discoveries by instrumenting the real world. For example, the sensor nodes in a wireless sensor network can be used collaboratively to collect data for the purpose of observing, detecting and tracking scientific phenomena. Sensor network deployment is becoming more commonplace in environmental, business and military applications. However, sensor networks are vulnerable to adversaries as they are frequently deployed in open and unattended environments. Anomaly detection is a key challenge in ensuring both the security and usefulness of the collected data. In this paper, we propose a method to filter the compromised nodes, be it self-healing or corrupted, using an autoregressive hidden Markov model (AR-HMM).

The paper is organized as follows. In the next Section II we cover the literature review for this work. In Section III, we give a brief overview of AR-HMMs. In Section IV, we describe the proposed algorithm, and in Section V we give numerical results. The conclusions are given in Section VI.

Manuscript received on September 30, 2016, accepted for publication on October 24, 2016, published on October 30, 2016.

The authors are with Missouri University of Sci & Tech, Rolla, MO 63128, USA (e-mail: bushra.anjum@gmail.com, chaman@mst.edu).

II. Related Work

Hidden Markov models and the Baum–Welch algorithm were first described in a series of articles by Leonard E. Baum and his peers at the Institute for Defense Analysis in the late 1960s [1]. However detecting compromised nodes using AR-HMM is a new area of investigation and we were unable to find any references that researched the same. Hence we provide the literature review in two parts, first, how compromised nodes are currently filtered and second, on the use of AR-HMM for solving diverse problems of identification, filtering, and prediction.

Wang & Bagrodia [2] have designed an intrusion detection system for identifying compromised nodes in wireless sensor networks using common application features (sensor readings, receive power, send rate, and receive rate). Hinds [3] has used Weighted Majority voting algorithm to create a concept of a node which could not be compromised, and to develop detection algorithms which relied on the trustworthiness of these nodes. Li, Song and Alam [4] have defined a data transmission quality function which keeps close to constant or change smoothly for legitimate nodes and decreases for suspicious nodes. The final decision of whether or not a suspicious node is compromised is determined by a group voting procedure. These designs take the en-network detection approach: misbehaved nodes are detected by their neighboring watchdog nodes. However en-network designs are insufficient to defend collaborative attacks when many compromised nodes collude together in the network. Zhang, Yu and Ning [5] present an alert reasoning algorithm for intelligent sensors using cryptographic keys. Zhanga et al. [6] exploit a centralized proven collision-resilient hashing scheme to sign the incoming, outgoing and locally generated/dropped message sets. These algorithms work on pinpointing exactly where the false information is introduced and who is responsible for it, but they are centralized and do so at a high computational cost. We propose a simple, decentralized model based approach to identify compromised nodes at a low computational cost using HMMs.

HMMs have been successfully used to filter unreliable agents, see Chang & Jiliu [7], Anjum et al. [8]. However the approach is unable to model correlation between observations. Autoregressive HMMs alleviate this limitation. Introduced in the 1980's, Juang and Rabiner published a series of papers [9], [10] regarding the application of Gaussian mixture

ISSN 2395-8618

autoregressive HMM to speech recognition. Switching autoregressive processes are well understood and have been applied in many areas. They have been particularly popular in *economics*, see Alexander [11] and Hamilton [12]. A brief summary of this subject and an extensive list of references can be found in [13]. The model is also successfully used by Park, Kwon and Lee [14], they have used AR-HMM by modeling the probabilistic dependency between sequential target appearances, presenting a highly accurate algorithm for robust visual tracking.

III. Proposed Model using Autoregressive Hidden Markov Models

A hidden Markov model is a Markov Chain with N states, which are hidden from the user. When the Markov chain is in a state, it produces an observation with a given state-dependent probability distribution. There are M different observation paths. It is this sequence of observations that the user sees, and from which it is possible to estimate the parameters of the HMM. An HMM is defined by the A matrix which is the one-step transition matrix of the Markov chain, the B matrix (referred to as the event matrix) which contains the probability distribution of the observations likely to occur when the Markov chain is in a given state, and π, the vector of probabilities of the initial state of the Markov chain. The (A, B, π) are together represented by λ. In this paper, we will make use of the following notation:

A: One-step transition Matrix ($N \times N$)
a_{ij}: One-step probability from state i to state j
B: Event Matrix ($N \times M$)
$b_i(k)$: Probability that the kth value will be observed when system shifts to state i
π: Initial Probability vector ($N \times 1$)
N: Number of hidden states
M: Number of observations
O: An observation sequence
T: Length of the observation sequence
Q: A sequence of hidden states traversed by the system
$\lambda = (A, B, \pi)$

A useful extension of HMM is *autoregressive* HMM (AR-HMM), which enhances the HMM architecture by introducing a direct stochastic dependence between observations. In AR(p)-HMM, the observation sequence is not only dependent on the HMM model parameters but also on a subset of p previous observations. Thus the model switches between sets of autoregressive parameters with probabilities determined by a state of transition probability similar to that of a standard HMM:

$$v_t = \sum_{r=1}^{P} a_r(s_t) v_{t-r} + \eta_t \quad with \quad \eta_t \sim N(0, \sigma^2)$$

where $a_r(s_t)$ is the r^{th} autoregressor when in state s ∈ {1 … N} at time t and each η_t is an i.i.d. normally distributed innovation with mean 0 and variance σ^2. Observations in the general AR-HMM can be continuous, but for our purposes we restrict the discussion to the discrete case. For example, a discrete HMM with AR(1) can be depicted by the Directed Acyclic Graph (DAG) as shown in Fig. 1. Three different, but related, problems have been defined for HMMs, briefly described below.

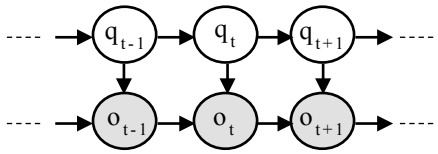

Fig. 1. DAG of the AR(1)-HMM where q_i and o_i represent the state and the observation generated at time slot i respectively.

Problem 1: Forward Probability Computation: Given an observation sequence, compute the probability that it came from a given λ.

Let the system be in state i at time t. The probability of the system shifting to state j at t+1 is given by the one-step transition probability a_{ij}. The probability that of the M possible observed states, the kth one is observed given that the system shifted from state i to j is $a_{ij}b_j(k)$. Since the state i can be any one of the states from 1 to N, the probability of observing the kth value, given that the system shifts to state j at time t+1 is the sum of all probable paths to state j, i.e., $\left[\sum_{i=1}^{N} a_{ij}\right] b_j(k)$.

Now, the only remaining unknown is the joint probability of having observed the sequence from O_1 through O_t and being in state i at time t, given λ. Representing this quantity by $\alpha_t(i)$ and representing the kth observed value at time t+1 by O_{t+1}, we have

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}), t = 1.2, \ldots, T-1, \qquad j \in [1, N]$$

To start off this induction, the initialization step for calculating $\alpha_t(i)$ can be obtained as follows. The probability of the system being in state i at time 1 is given by π_i and the probability of observing O_1 is then given by $b_j(O_1)$. Thus,

$$\alpha_1(i) = \pi_i b_j(O_1), i \in [1, N]$$

$\alpha_T(i)$ obtained from the induction step represents the probability of observing the sequence from O_1 through O_T, and ending up in state i. Thus, the total probability of observing the sequence O_1 through O_T, given λ is

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

ISSN 2395-8618

Problem 2: Backward Probability Computation: Given an observation sequence, compute the probable states the system passed through.

The quantity $\beta_t(i)$ is defined as the probability that the sequence of observations from O_{t+1} through O_T are observed starting at state i at time t for a given λ. It is calculated in the same way as $\alpha_t(i)$, but in the backward direction. We have,

$$\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j) a_{ij}\, b_j(O_{t+1}), \qquad t = T-1, \ldots, 1$$

For $t = T$ we have $\beta_T(i)$=1, i = 1, 2, …, N. To calculate the probability that the system was in state i at time t given O and λ, we observe that $\alpha_t(i)$ accounts for the observation sequence from O_1 through O_t and $\beta_t(i)$ accounts for O_{t+1} through O_T, and both account for the state i at time t. So the required probability is given by $\alpha_t(i)\beta_t(i)$. Introducing the normalizing factor from problem 1, $P(O|\lambda)$, we have

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)}$$

At each time t, the state with the highest γ is the most probable state at time t. A better way to obtain the most probable path of states Q that give rise to an observation sequence O, is to use dynamic programming, as described in [13].

Problem 3: Matrix Estimation: Given an observation sequence O, compute the most probable λ.

The term a_{ij} can be calculated as the ration of the number of transitions made from state i to state j over the total number of transitions made out of state i. We have from problem 2 that $\gamma_t(i)$ is the probability of being in state i at time t. Extending this, the probability of being in state i at time t and in state j at time t+1 can be calculated as follows:

$$\xi_t(i,j) = \frac{\alpha_t(i)\, a_{ij}\, b_j \beta_{t+1}(j)}{P(O|\lambda)}$$

IV. The Algorithm

We consider a set of sensors that consists of a mix of healthy, self-healing and corrupted sensors. We assume that time is slotted, let T be the total number of slots. During each slot, each sensor submits a reading about the environment. For simplicity, let all the nodes submit a reading at each time slot. (This constraint can be easily removed, by appropriately modifying the way the $\alpha_i(t)$ and $\beta_i(t)$ are calculated.)

For each node, we have a sequence of T readings (observations) to train an AR-HMM. Thus, we end up with as many AR-HMMs as the number of nodes. Using statistical clustering techniques, we cluster the B matrices of these AR-HMMs into two groups, one for healthy and the other for compromised nodes (self-healing and corrupted). This permits us to identify and filter out the compromised sensors.

To test the accuracy of the algorithm, we consider three types of sensors, healthy, self-healing and corrupted. To achieve this, we define three λ's, all having the same A and π, but different B. The B matrix of the healthy nodes (B_h) should be such that the readings generated echo the environmental phenomenon it is sensing. The B matrix of self-healing nodes (B_s) introduces spurs of invalid data before moving back to the valid state and the B matrix of the corrupted nodes (B_c) is set up to predominantly generate invalid data. The exact matrices are defined in the next section.

Next, this sequence of T readings is used to estimate the A, B, and π matrices of the sensor. Using the B matrices, we cluster the sensors into two groups, i.e., healthy and compromised nodes. For this, we take the mean squared error (MSE) of each estimated B matrix with the perfectly healthy matrix B_p (see Section IV), where

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\sum_{k}^{M} (b_i(k) - b_{pi}(k))^2$$

and $b_i(k)$ and $b_{pi}(k)$ are the $(i,k)^{\text{th}}$ element of the B and B_p matrices respectively.

The resulting MSE values are then classified into two clusters using the k-means clustering algorithm. As B_p represents the truly healthy matrix, the cluster with a center closer to 0 represents the group of healthy nodes.

V. Experimental Setup and Results

A. Synthetic Datasets – Richardson's Model

For our simulation study, we use the classic Richardson's model for the temperature [14] to define the transition matrix A and the autoregressors v_t. Richardson's model uses two states S_t = *{Dry, Wet}*, hence N = 2, and second order autoregression, AR(2), to define the temperature readings. The model is given as follows:

$$A = \begin{bmatrix} 0.55 & 0.45 \\ 0.33 & 0.77 \end{bmatrix}$$

$$v_t = \begin{cases} 1.22 + 0.90Y_{t-1} - 0.13Y_{t-2} + 2.11\eta_t & (dry) \\ 2.14 + 0.70Y_{t-1} - 0.003Y_{t-2} + 1.87\eta_t & (wet) \end{cases}$$

The results reported in this section were based on 60% healthy nodes, 20% self-healing, and 20% corrupted nodes. In all experiments, we set the Markov chain to start with an equal probability of being in any of the two hidden states, i.e., π = [0.5, 0.5]. We define three B matrices (B_h, B_s and B_c), such that they model the behavior of the three nodes under consideration, healthy, self-healing and corrupted respectively. B is a 2 × 2 matrix where the first column corresponds to the probability of generating a value using autoregressive equations, and second column corresponds to the probability

ISSN 2395-8618

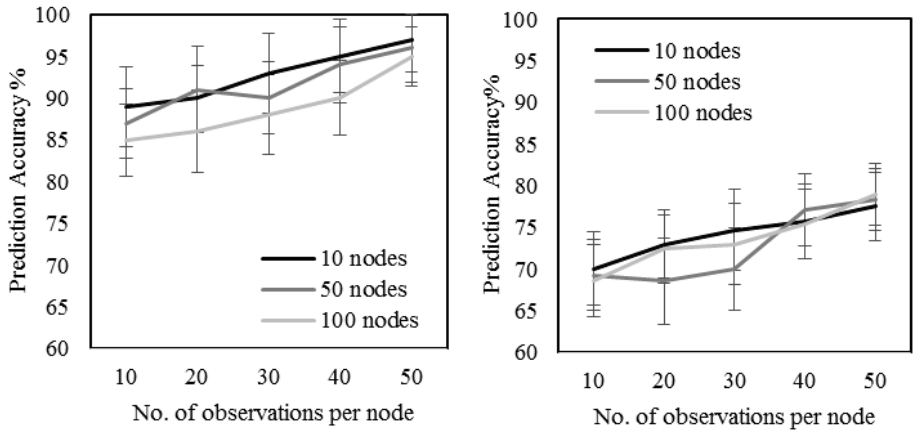

Fig. 2. Prediction accuracy for healthy nodes (left), corrupted nodes (right)

TABLE I
PREDICTION ACCURACY FOR SYNTHETIC SENSOR DATA MODEL

	Percentage of self-healing nodes					
	0%	10%	20%	30%	40%	50%
Healthy nodes	0.98	0.96	0.95	0.97	0.96	0.95
Corrupted nodes	0.95	0.88	0.84	0.78	0.75	0.68

of entering the corrupted state and generating an invalid value. (The selection of N = M = 2 is not significant, and any other values can be readily used).

$$B_h = \begin{bmatrix} 0.95 & 0.05 \\ 0.95 & 0.05 \end{bmatrix}, B_s = \begin{bmatrix} 0.65 & 0.35 \\ 0.65 & 0.35 \end{bmatrix}, \quad B_c = \begin{bmatrix} 0.05 & 0.95 \\ 0.05 & 0.95 \end{bmatrix}$$

The perfectly healthy matrix, B_p used for MSE calculations is defined as

$$B_p = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$$

For the above A matrix, we define three AR(2)-HMMs, namely, $\lambda_h = (A, B_h, \pi)$, $\lambda_s = (A, B_s, \pi)$ and $\lambda_c = (A, B_c, \pi)$.. Subsequently, we generate random samples of temperature readings for all the nodes, and then apply the algorithm described in the previous section to identify the group of healthy nodes. Then, we compare the number of identified healthy nodes to the original set of healthy sensors and report the result as a percentage of correctly identified healthy nodes. We used MS_Regress MATLAB package for Markov Regime Switching Models by Perlin [15] to implement the above algorithm and obtain numerical results.

The results are given in Fig. 2 which gives the percentage of correctly identified healthy (left) and corrupted (right) nodes along with the 95th confidence interval. For the figure, we assume 10, 50 and 100 sensors, and vary the number of observations per sensor from 10 to 50. The confidence intervals are obtained by replicating each result 100 times, and each time using a different seed for the pseudo-random number generator.

Interesting observations (Fig. 2): As the number of readings increases per node, the prediction accuracy the nodes (both healthy and corrupted) increases as well. For healthy nodes, the average prediction accuracy for 10 readings is around 88% and for 50 readings per node, it jumps to around 97%. Overall, the prediction accuracy of healthy nodes is much higher than that of corrupted nodes. This can be explained by the fact that the cluster distribution of self-healing nodes and corrupted nodes is quite similar whereas both differ significantly from the cluster distribution of healthy nodes.

Based on the results, we also observe that it is not necessary to have a large sample of nodes and readings per node. We see that with as little as 10 nodes and 10 readings per node, we obtain results which are similar to those obtained with larger number of nodes and readings per node. This leads us to conclude that our approach can be used in a decentralized manner, i.e., we do not need data from all (or a large number) of the sensors in order to prune out the compromised ones.

Finally, we turn towards sensitivity analysis of our approach, i.e., how does the percentage of self-healing nodes impact the filtering accuracy. For this we varied the percent of self-healing nodes from 0 to 50%. Keeping the number of nodes to 50 and the number of readings per node to also 50, we calculate the prediction accuracy of the healthy and corrupted nodes. The results are presented in Table I.

We make the following observations. As the number of self-healing nodes increases, the prediction accuracy of both healthy nodes and corrupted nodes decreases. However, where

ISSN 2395-8618

the change in healthy node prediction is minor (98% to 95%), corrupted node prediction suffers increasingly as the number of self-healing nodes increases (95% to 68%). This confirms our initial observation that the cluster distribution of self-healing nodes is similar in nature to corrupted nodes. Hence it becomes difficult for the algorithm to pick corrupted nodes exclusively from the set of corrupted and self-healing nodes. The success of our approach, however, lies with the identification of healthy nodes, with accuracy greater than 90%.

B. Real Sensor Datasets

In order to test our models on real world sensor datasets, we use datasets from two sources: Intel Berkley Research Laboratory [16] and Labeled Wireless Sensor Network Data Repository (LWSNDR) projects [17]. Both projects represent the state of the art in sensor systems and collect measurements in very different environments. Hence, these datasets allow us to evaluate the accuracy of AR-HMM classification on representative and diverse sensor system data.

First dataset is collected from 54 sensors deployed in the Intel Berkeley Research lab. The Mica2Dot sensors with weather boards collected time stamped topology information, along with humidity, temperature, light and voltage values once every 31 seconds. This dataset includes a log of about 2.3 million readings collected from the 54 motes (mote is a sensor that is capable of doing some processing in addition to collecting and transmitting data), where data from some motes may be missing or truncated.

Second dataset is collected from a simple single-hop and a multi-hop wireless sensor network deployment using TelosB motes. The data consists of humidity and temperature measurements collected during 6 hour period at intervals of 5 seconds. For this evaluation, we are using the single hop labeled readings which consist of approximately 15,000 entries.

The first dataset is unlabeled. We have tested our anomaly detection algorithms with by visually inspecting the sensor data time series. The second dataset, is a labelled wireless sensor network dataset where label '0' denotes normal data and label '1' denotes an introduced event (anomaly). More details can be obtained from [18].

We assume that the sensor readings collected over a reasonable duration capture the normal patterns in the sensor data series. As a general rule, we have used 20% of the data points to work out the auto regression equations for the system. The auto regressive coefficients are calculated using the least squares method. Also, the same data points are used to calculate $\lambda = (A, B, \pi)$ as defined under Problem 3 in Section 2.

TABLE II
OVERALL PREDICTION ACCURACY FOR REAL SENSOR DATA

	Algorithm	Accuracy
Intel Berkley Research Lab	Naïve Bayes	89.785 %
	ZeroR	85.543 %
	AR-HMM	**97.231 %**
Labelled WSN Data Repository	Naïve Bayes	90.754 %
	ZeroR	86.352 %
	AR-HMM	**98.413 %**

For comparison, we are using two popular classification algorithms ZeroR and Naïve Bayes along with our proposed AR-HMM model. Briefly, ZeroR is a useful predictor for determining a baseline performance, predicting mean for a numeric class and mode for a nominal class, and Naïve Bayes is a conditional probabilistic classifier based on Bayer's theorem. The filtering accuracy to identify healthy nodes is given in Table II.

Table II displays that as ZeroR provides a baseline accuracy, AR-HMM clearly surpasses Naïve Bayesian classification for correctly identifying the healthy sensors. We further describe our method's accuracy using (1) number of false positives (detecting non-exist compromised nodes) and (2) number of false negatives (not being able to detect a compromised node) as our metrics. Specifically, the results in Table III below are presented as follows – the x/y number indicates that x out of y compromised nodes were detected correctly (corresponding to y-x false negatives) plus we also indicate the number of corresponding false positives.

VI. CONCLUSION

In this paper, we propose a method based on autoregressive hidden Markov models (AR-HMMs) for filtering out

TABLE III
PREDICTION ACCURACY FOR HEALTHY AND COMPROMISED NODES INDIVIDUALLY

	Algorithm	Healthy nodes	Compromised nodes
Intel Berkley Research Lab	Naïve Bayes	33/40	9/14
	ZeroR	30/40	8/14
	AR-HMM	**37/40**	**11/14**
Labelled WSN Data Repository	Naïve Bayes	12/16	3/5
	ZeroR	11/16	2/5
	AR-HMM	**14/16**	**4/5**

ISSN 2395-8618

compromised nodes in a sensor network. We confirm through experimentation, based on revised Richardson's temperature model, that our filtering method is quite accurate and identifies the healthy sensors with an accuracy greater than 90%. We further used real sensor data from Intel Labs and WSN repository from UNC to calculate the prediction accuracy of AR-HMM approach as opposed to Naïve Bayes and ended up with encouraging results, with prediction accuracy as high as 97%

REFERENCES

[1] R. Lawrence, "First Hand: The Hidden Markov Model". *IEEE Global History Network*. Retrieved 2 October 2013.

[2] Y.T Wang and R. Bagrodia, "Com-Sen: A Detection System for Identifying Compromised Nodes in Wireless Sensor Networks", *SECURWARE 2012*.

[3] C.V. Hinds, *Efficient detection of compromised nodes in wireless sensor networks*, 2012.

[4] T. Li, M. Song, M. Alam, "Compromised Sensor Nodes Detection: A Quantitative Approach," *ICDCSW 2008*.

[5] Q. Zhang, T. Yu, and P. Ning, "A Framework for Identifying Compromised Nodes in Wireless Sensor Networks," *ACM Transactions on Information and Systems Security*, vol. 11, no. 3, Article 12, 2008.

[6] Y. Zhanga, J. Jun Yangb, W. Lia, L. Wangc, and L. Jind, "An authentication scheme for locating compromised sensor nodes in WSNs," *Journal of Network and Computer Applications*, vol. 33, no. 1, 2010.

[7] L. Chang, and Z. Jiliu. "The reputation evaluation based on optimized hidden Markov model in e-commerce," *Mathematical Problems in Engineering*, vol. 2013, Hindawi, 2013.

[8] B. Anjum, M. Rajangam, H. Perros, and W. Fan, "Filtering Unfair Users: A Hidden Markov Model Approach," *ICISSP*, Loire, France, 2015.

[9] B.H. Juang and L.R. Rabiner, "A Probabilistic Distance Measure for Hidden Markov Models," *AT&T Tech. J.*, vol. 64, no. 2, pp. 391–408, 1985.

[10] B.H. Juang and L.R. Rabiner, "Mixture Autoregressive Hidden Markov Models for Speech Signals," *IEEE Trans.*, vol. ASSP-33, no. 6 , pp. 1404–1413, 1985.

[11] C. Alexander, "Market Risk Analysis: Practical Financial Econometrics," Wiley Books, 2008.

[12] J.D. Hamilton, "Regime Switching Models," *Palgrave Dictionary of Economics*, available at http://dss.ucsd.edu/~jhamilto/palgrav1.pdf, 2005.

[13] Y. Ephraim and N. Merhav, "Hidden Markov processes," *IEEE Trans. Inf. Theory*, vol. 48, no. 6, pp. 1518–1569, 2002.

[14] C.W. Richardson, "Stochastic simulation of daily precipitation, temperature, and solar radiation," *Water Resour. Res.*, vol. 17, no. 1, 182–190, doi:10.1029/WR017i001p00182, 1981.

[15] M. Perlin, "MS Regress. The MATLAB Package for Markov Regime Switching Models." Available at http://ssrn.com/abstract=1714016, 2014.

[16] Intel Lab Data. http://db.csail.mit.edu/labdata/labdata.html

[17] UNC Greensboro, *Machine Learning Models and Algorithms for Big Data Classification*. http://www.uncg.edu/cmp/downloads.

[18] S. Suthaharan, M. Alzahrani, S. Rajasegarar, C. Leckie and M. Palaniswami, "La-belled Data Collection for Anomaly Detection in Wireless Sensor Networks", in *Proceedings of the Sixth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP 2010)*, Brisbane, Australia, 2010.

ISSN 2395-8618

Robust Spoken Language Understanding for House Service Robots

Andrea Vanzo, Danilo Croce, Emanuele Bastianelli, Roberto Basili, and Daniele Nardi

***Abstract*—Service robotics has been growing significantly in the last years, leading to several research results and to a number of consumer products. One of the essential features of these robotic platforms is represented by the ability of interacting with users through natural language. Spoken commands can be processed by a Spoken Language Understanding chain, in order to obtain the desired behavior of the robot. The entry point of such a process is represented by an Automatic Speech Recognition (ASR) module, that provides a list of transcriptions for a given spoken utterance. Although several well-performing ASR engines are available off-the-shelf, they operate in a general purpose setting. Hence, they may be not well suited in the recognition of utterances given to robots in specific domains. In this work, we propose a practical yet robust strategy to re-rank lists of transcriptions. This approach improves the quality of ASR systems in situated scenarios, i.e., the transcription of robotic commands. The proposed method relies upon evidences derived by a semantic grammar with semantic actions, designed to model typical commands expressed in scenarios that are specific to human service robotics. The outcomes obtained through an experimental evaluation show that the approach is able to effectively outperform the ASR baseline, obtained by selecting the first transcription suggested by the ASR.**

***Index Terms*—Spoken language understanding, service robotics, re-ranking of automatic speech recognition systems.**

I. Introduction

DURING the recent years, the interest of the research community in the Robotics field has been rapidly increasing: robotic platforms are spreading in our domestic environments and the research on Service Robotics is becoming a hot topic. A significant aspect in this context is the study of the interaction between humans and robots, especially when this communication involves non-expert users. For this reason, natural language is a key component in human-robot interfaces. Specifically, the task of *Spoken Language Understanding* (SLU) is related to the interpretation of spoken language commands and their mapping into actions that can be executed by a robotic platform in the operational environment. Hence, the input of a typical SLU process is the user's speech, while the output can be either the corresponding action or, more in general, a response. When dealing with this problem, manifold approaches can be adopted. On the one hand, grammar-based approaches allow the design of systems that embed the entire process in a single stage, from the speech recognition up to the semantic interpretation, e.g., [1], [2], [3]. These systems rely on grammars generated by knowledge engineers, that aim at covering the (possibly vast) plethora of linguistic phenomena the user may be interested into. Moreover, these grammars can be provided with semantic attachments [4], that enable for a structured representation of the meaning of the sentence. On the other hand, approaches relying on statistical methods [5] alleviate the need to explicitly encode the information required by the NLU process, but they require training data annotated with the targeted (linguistic) phenomena the final system is expected to capture.

Regarding the Automatic Speech Recognition (ASR) systems, most of the existing off-the-shelf solutions are based on very well-performing statistical methods [6], that enable their adoption in everyday scenarios. Nevertheless, these tools rely on general-purpose language models and false positives might be generated in specific scenarios. For example, they may be optimized to transcribe queries for a Search Engine, that are characterized by different linguistic constructions with respect to a command for a robot. However, it is reasonable to expect that domain-specific scenarios provide knowledge and specific information that can improve the performance of any off-the-shelf ASR. To this regard, several works proposed techniques where a hybrid combination of free-form ASRs and grammar-based ASRs is employed to improve the overall recognition accuracy. In these approaches, the grammar-based ASR is often used to prune the transcriptions hypothesized by the free-form ASR [7], [8] or to generate new training sentences [9], [10]. Nevertheless, the above approaches are subject to several issues. In fact, as often emphasized, e.g., [11], grammar-based approaches may lack of adequate coverage, especially in dealing with the variability of (often ungrammatical) spoken language, causing a high rate of failures in the recognition of the transcription of the ASR system. On the contrary, a highly complex grammar can improve the coverage of the captured linguistic phenomena. However, this complexity may introduce ambiguities. Moreover, the cost of developing and maintaining a complex grammar may be inapplicable in realistic applications.

Manuscript received on February 8, 2016, accepted for publication on June 16, 2016, published on October 30, 2016.

Andrea Vanzo and Daniele Nardi are with the Sapienza University of Rome, Department of Computer, Control and Management Engineering "Antonio Ruberti", Rome, Italy (e-mail: {vanzo, nardi}@diag.uniroma1.it).

Danilo Croce and Roberto Basili are with the University of Roma, Tor Vergata, Department of Enterprise Engineering, Rome, Italy (e-mail: {croce, basili}@info.uniroma2.it).

Emanuele Bastianelli is with the University of Roma, Tor Vergata, Department of Civil Engineering and Computer Science Engineering, Rome, Italy (e-mail: bastianelli@ing.uniroma2.it).

ISSN 2395-8618

In this work, we propose an approach to increase the robustness of an *off-the-shelf* free-form ASR system in the context of Spoken Language Understanding for Human-Robot Interaction (HRI), relying on grammars designed over specific domains. Our target is house service robotics, with the special purpose of understanding spoken commands. We rely here on the semantic grammar proposed in [2]: this is modeled around the task of interpreting commands for robots expressed in natural language by encoding (i) the set of allowed actions that the robot can execute, (ii) the set of entities in the environment that should be considered by the robot and (iii) the set of syntactic and semantic phenomena that arise in the typical sentences of Service Robotics in domestic environment. In [2], this grammar has been used to directly provide a semantic interpretation of spoken utterances. However, this interpretation requires every sentence to be entirely recognized by this grammar: even a single word or syntactic construct missing in the process may potentially cause the failure of the overall process.

We propose here to adopt a grammar to improve the robustness of an ASR system by relying on a *scaling-down* strategy. First, we relax some of the grammar constraints allowing the coverage of shallower linguistic information. Given a grammar, we derive two lexicons designed to recognize (i) the mention to robotic actions (ii) the mention to entities in the environment. For each lexicon, we define a specific *cost* that is inversely proportional to its correctness. The transcriptions initially receive a cost that is inversely proportional to the rank provided by the ASR system and, each time one of them is recognized by the grammar or a lexicon, the corresponding cost decreases. The more promising transcription is the one minimizing the corresponding final cost. The final decision thus depends on the combination of all the costs so that, even when none of the transcriptions is recognized by the complete grammar, their rank still depends on the lexicons. In this way, those transcriptions that do not refer to any known actions and/or entities are accordingly penalized.

The proposed re-ranking strategy has been evaluated on the Human Robot Interaction Corpus (HuRIC, [12]) a collection of utterances semantically annotated and paired with the corresponding audio file. This corpus is related with the adopted semantic grammar as this has been designed by starting from a subset of utterances contained in HuRIC. Experimental results show that the proposed method is effective in re-ranking the list of hypothesis of a state-of-the-art ASR system, especially on the subset of utterances whose transcriptions are not recognized by the grammar, i.e., no pruning strategy is applicable.

In the rest of the paper, Section II provides an overview of the existing approaches to improve the quality of ASR systems. Section III presents the proposed approach and defines individual cost factors. In Section IV an experimental evaluation of the re-ranking strategy is provided and discussed. Finally, Section V derives the conclusions.

II. Related work

The robustness of Automatic Speech Recognition in domain-specific settings has been addressed in several works. In [13], the authors propose a joint model of the speech recognition process and language understanding task. Such a joint model results in a re-ranking framework that aims at modeling aspects of the two tasks at the same time. In particular, re-ranking of n-best list of speech hypotheses generated by one or more ASR engines is performed by taking the NLU interpretation of these hypotheses into account. On the contrary, the approach proposed in [14] aims at demonstrating that perceptual information can be beneficial even to improve the language understanding capabilities of robots. They formalize such information through Semantic Maps, that are supposed to synthesize the perception the robot has of the operational environment.

Regarding the combination of free-form ASR engines and grammar based systems, in [15] two different ASR systems work together sequentially: the first is grammar-based and it is constrained by the rule definitions, while the second is a free-form ASR, that is not subject to any constraint. This approach focuses on the acceptance of the results of the first recognizer. In case of rejection, the second recognizer is activated. In order to improve the accuracy of such a decision, the authors propose an algorithm that augments the grammar of the first recognizer with valid paths through the language model of the second recognizer. In [7], a robust ASR for robotic application is proposed, aiming at exploiting a combination of a Finite State Grammar (FSG) and an n-gram based ASR to reduce false positive detections. In particular, a hypothesis produced by the FSG-based decoder is accepted if it matches some hypotheses within the n-best list of the n-gram based decoder. This approach is similar to the one proposed in [16], where a *multi-pass decoder* is proposed to overcome the limitations of single ASRs. The FSG is used to produce the most likely hypothesis. Then, the n-gram decoder produces an n-best list of transcriptions. Finally, if the best hypothesis of the FSG decoder matches with at least one transcription among the n-best, then the sentence is accepted. A hybrid language model is proposed in [8]. It is defined as a combination of a n-gram model, aiming at capturing local relations between words, and a category-based stochastic context-free grammar, where words are distributed into categories, aiming at representing the long-term relations between these categories. In [9], an interpretation grammar is employed to bootstrap Statistical Language Models (SLMs) for Dialogue Systems. In particular, this approach is used to generate SLMs specific for a dialogue move. The models obtained in this way can then be used in different states of a dialogue, depending on some contextual constraints. In [17], n-grams and FSG are integrated in one decoding process for detecting sentences that can be generated by the FSG. They start from the assumption that sentences of interest are usually surrounded by carrier phrases. The n-gram is aimed at

ISSN 2395-8618

detecting those surrounding phrases and the FSG is activated in the decoding-process whenever start-words of the grammar are found.

All the above approaches can be considered complementary to the one proposed here. However, the advantages of our method are mainly in the simplicity of the proposed solution and the independence of the resulting work-flow from the adopted free-form ASR system: our aim is to define a simple yet applicable methodology that can be usable in every robot.

III. A ROBUST DOMAIN-SPECIFIC APPROACH

In this section, we propose an approach to select the most correct transcription among the results proposed by a Automatic Speech Recognition (ASR) system. Given a spoken command from the user, e.g., *move to the fridge*, such a system produces a rank of possible transcriptions such as

1) *move to the feet*
2) *more to the fridge*
3) *move to the fridge*
4) *move to the fate*
5) *move to the finch*

In this case, the correct transcription is ranked as third. In order to choose this sentence, we apply a cost function to the hypotheses based on (i) the adherence to the robot grammar, as it describes the typical commands for a robot, (ii) the recognition of action(s) applicable/known to the robot (as for *move*) and (iii) the recognition of entities, like nouns referring to objects recognized/known to the robot, e.g., *fridge*. The cost function we propose decreases along with the constraints satisfied by the sentence, e.g., the second sentence satisfies (iii), but not (i) and (ii) (as $more$ is not an action); as a consequence it results into a higher cost with respect to the third transcription. Before discussing the cost function as a ASR ranking methodology, we define the grammatical framework used in this work, in line with [2].

A. Grammar-based SLU for HRI

Robots based on speech recognition grammars usually rely on speech engines whose grammars are extended according to conceptual primitives, generally referring to known lexical theories such as Frame Semantics [18]. Early steps in the HRI chain are based on ASR modules that derive a parse tree encoding both syntactic and semantic information based on such theory. Parse trees are based on grammar rules activated during the recognition, and augmented by an instantiation of the corresponding semantic frame, that corresponds to an action the robot can execute. Compiling the suitable robot command proceeds by visiting the tree and mapping recognized frames into the final command.

The applied recognition grammar jointly models syntactic and semantic phenomena that characterize the typical sentences of HRI applications in the context of Service Robotics. It encodes a set of imperative and descriptive commands in a verb-arguments structure. Each verb is retained as it directly evokes a frame, and each (syntactic) verb argument corresponds to a semantic argument. The lexicon of arguments is semantically characterized, as argument fillers are constrained by one (or more) semantic types. For example, for the semantic argument THEME of the BRINGING frame, only the type TRANSPORTABLE_OBJECTS is allowed. As a consequence, a subset of words referring to things transportable by the robots, e.g., *can*, *mobile phone*, *bottle* is accepted. A subset of the grammar for the BRINGING frame, covering the sentence *Bring the book to the table* is reported hereafter:

`Bringing` → `Target Theme Goal` | ...
`Target` → *bring* | *carry* | ...
`Theme` → *the* `Transportable_objects` | ...
`Transportable_objects` → *can* | *book* | *bottle* | ...
`Goal` → ...

We will distinguish between terminals denoting entities (such as can, $book$, $bottle$ that belong to the lexicon of TRANSPORTABLE_OBJECTS) from the lexicon of possible actions (such as $bring$, $take$ or $carry$ characterizing the actions of the frame BRINGING) as they will give rise to different predicates augmented with grammatical constraints. Moreover, transcribed sentences covered by the grammar, i.e., belonging to the grammar language, are more likely to correspond to the intended command expressed by the user, and should be ranked first in the ASR output.

B. A grammar-based cost model for accurate ASR ranking

A first interesting type of constraint is posed by the ASR system itself. In fact the rank proposed by an ASR system is usually driven by a variety of linguistic knowledge in the ASR device. A basic notion of cost can be thus formulated ignoring the domain of the specific grammar.

Given a spoken utterance v, let $\mathcal{H}(v)$ be the corresponding list of hypotheses produced by the ASR. The size $|\mathcal{H}(v)| = N$ corresponds to the number of hypotheses. Each hypothesis $h \in \mathcal{H}(v)$ is a pair $\langle s, \omega(s) \rangle$, where s is the transcription of v, and $\omega(s)$ is a cost attached to s. Let $p(s)$ be its position in the ASR systems ranking. According to this cost function, the higher is $\omega(s)$, the lower the confidence in h being the correct transcription.

Since many off-the-shelf ASR systems do not provide the confidence score for each transcription, in order to provide a general solution, only the rank is taken into account: let v be a spoken utterance and $\mathcal{H}(v)$ the corresponding list of transcriptions, then, $\forall s \in \mathcal{H}(v)$ the *ranking cost* ω_{rc} is defined as follows:

$$\omega_{rc}(s, \theta) = \frac{p(s) + \theta}{\sum_{s' \in \mathcal{H}(v)} p(s') + \theta N} \tag{1}$$

where $p(s)$ corresponds to the position $(1, \ldots, |H(v)|)$ of s in $\mathcal{H}(v)$. Here θ is a smoothing parameter that enables the tuning of the variability allowed to the final rank with respect to the initial rank proposed by the ASR system.

ISSN 2395-8618

The overall cost assigned to a transcription s depends on the ASR ranking as well as on the grammar. Let $s \in \mathcal{H}(v)$, let ω_i be a parametric cost depending on the grammar $\mathcal{G}$, the overall cost $\omega(s)$ can be defined as:

$$\omega(s) = log\big(\omega_{rc}(s,\theta)\big) + \sum_i log\big(\omega_i(s,\alpha_i)\big) \tag{2}$$

where the different ω_i capture different aspects of the grammar $\mathcal{G}$ with scores derived from the grammatical or lexical criteria. Higher values of ω_i correspond to stronger violations. Moreover, $\omega_{rc}(s,\theta)$ is the ranking cost as in Equation (1), while α_i is the parameter associated to each cost ω_i.

In this paper we investigate three possible cost factors, i.e., $i = 1, 2, 3$, to enforce information derived by different grammatical, i.e., domain-dependent, constraints. As these can be different, we designed three different cost factors:

- $\omega_G(s, \alpha_G)$ is the *complete-grammar cost* that is minimal when the transcription belongs to the language generated by the grammar $\mathcal{G}$, and maximal otherwise;
- $\omega_A(s, \alpha_A)$ is the *action-dependent cost* that is minimal when the transcription explicitly refers to actions the robot is able to perform, and maximal otherwise;
- $\omega_E(s, \alpha_E)$ is the *entity-dependent cost* that takes into account the entities targeted by the commands, and is minimal if they are referred into the transcription s and maximal otherwise.

These cost factors are detailed hereafter.

Complete-grammar cost. When dealing with the Spoken Language Understanding with robots we may want to restrict the user sentences to a set of possible commands. This is often realized by defining a grammar covering the linguistic phenomena we want to catch. Moreover, if the grammar is designed to embed also semantic information as in [2], it can be introduce also higher level semantic constraints. For instance, the BRINGING action can be applied only to TRANSPORTABLE_OBJECTS. As an example, a sentence a transcription such as *bring me the fridge* is discarded by the grammar if the *fridge* is not a TRANSPORTABLE_OBJECTS.

Let $\mathcal{G}$ be a grammar designed for parsing commands for a robot R. Let $L(\mathcal{G})$ be the language generated by the grammar, i.e., the set of all possible sentences that $\mathcal{G}$ can produce. Then, the *complete-grammar cost* ω_G is computed as

$$\omega_G(s, \alpha_G) = \begin{cases} \alpha_G & \text{if } s \in L(\mathcal{G}) \\ 1 & \text{otherwise} \end{cases} \tag{3}$$

where $\alpha_G \in (0, 1]$ is a weight that measures the strength of the violation and can be used to weight the impact of an *"out-of-grammar"* transcription. Notice that the weight α_G can be either set as a subjective confidence or tuned through a set of manually validated hypotheses. If α_G is set to 1, no grammatical constraint is applied and the complete grammar cost has no effect.

Action-dependent cost. Robot specifications enable the construction of the lexicon of potential actions A, hereafter called $\mathcal{L}_A$. Let A be the set of actions that a robot can perform, e.g., MOVE, GRASP, OPEN. For each action $a \in A$, a corresponding set of lexical entries can be used to linguistically refer to a: we will denote such a set as $\mathcal{L}(a) \subset \mathcal{L}_A$.

The *action-dependent cost* ω_A for a transcription $s \in \mathcal{H}(v)$ is thus given by:

$$\omega_A(s, \alpha_A) = \prod_{\forall w \in s} \alpha_A(w) \tag{4}$$

where $\alpha_A(w)$ is defined as:

$$\alpha_A(w) = \begin{cases} \alpha_A & \exists a \in A \text{ such that } w \in \mathcal{L}(a) \\ 1 & \text{otherwise} \end{cases} \tag{5}$$

$\alpha_A \in (0, 1]$ is a weight that favors words corresponding to actions that are in the repertoire of the robot. The weight α_A can be either set as a subjective preference or tuned over a set of manually validated hypotheses. Note that if α_A is set to 1, no action dependent constraint is applied and the corresponding cost is not triggered.

Entity-dependent cost. Exploiting environment observations can be beneficial in interpreting commands. Notice that the objects of the robot's environment are more likely to be referred by correct transcriptions rather than by the wrong ones, as these are usually "out of scope". Let $\mathcal{G}$ be the grammar designed for commands. Given the set of terminals of $\mathcal{G}$, in the lexicon $\mathcal{L}_G$ a specific set of terms is used to make (explicit) reference to objects of the environment. For each entity e (e.g., MOVABLE OBJECTS such as $bottles$, $books$, ..., or FURNITURES, such as $table$ or $armchair$) the set of nouns used to refer to e in the language $L(\mathcal{G})$ is well defined, and it is denoted by $\mathcal{L}(e)$.

The *entity-dependent cost* ω_E for a transcription $s \in \mathcal{H}(v)$ is thus given by:

$$\omega_E(s, \alpha_E) = \prod_{\forall w \in s} \alpha_E(w) \tag{6}$$

where $\alpha_E(w)$ is defined as:

$$\alpha_E(w) = \begin{cases} \alpha_E & \exists \text{ entity } e \text{ such that } w \in \mathcal{L}(e) \\ 1 & \text{otherwise} \end{cases} \tag{7}$$

and $\alpha_E \in (0, 1]$ is a weight that favors words corresponding to entities the robot is able to recognize in the environment. The weight α_E can be either set as a subjective preference or tuned over a set of manually validated hypotheses. Also α_E, when set to 1, produces no entity dependent constraint and corresponds to a null impact on the final cost.

IV. EXPERIMENTAL EVALUATIONS

The grammar employed in these evaluations has been designed in [19], lately improved in [2], and its definition is compliant to the Speech Recognition Grammar Specification [4]. The grammar takes into account 17 frames, each of which is evoked by an average of 2.6 lexical units.

ISSN 2395-8618

On average, for each frame 27.9 syntactic patterns are defined. Entities are clustered in 28 categories, with an average amount of items per cluster of 11.2 elements. We extracted an Actions Lexicon $\mathcal{L}(a)$ containing 44 different verbs. The Entities Lexicon $\mathcal{L}(e)$ is composed of 216 and 97 single and compound words, respectively, with a total amount of 313 entities. The dataset of the empirical evaluation is the HuRIC corpus[1], a collection of utterances annotated with semantic predicates and paired with the corresponding audio file. HuRIC is composed of three different datasets, that display an increasing level of complexity in relationship with the grammar employed.

The **Grammar Generated** dataset (GG) contains sentences that have been generated by the above speech recognition grammar. The **Speaky for Robot** dataset (S4R) has been collected during the *Speaky for Robots project*[2] and contains sentences for which the grammar has been designed, so that the grammar is supposed to recognize a significant number of utterances. While the grammar is expected to cover all the sentences in the GG dataset, this may be not true for the S4R one, as some sentences are characterized by linguistic structures not considered in the grammar definition. The **Robocup** dataset (RC) has been collected during the 2013 *Robocup@Home* competition [20] and it represents the most challenging section of the corpus, given its linguistic variability. In fact, even referring to the same house service robotics, it contains sentences not constrained by the grammar structure, as, during the acquisition process, speakers were allowed to say any kind of sentence related to the domain.

The experimental evaluation aimed at measuring the effectiveness of the approach we proposed. To this end, the cost function $\omega(s)$ has been used in different settings. The α_i can be used to properly activate/deactivate the costs operating on specific evidences. In fact, if $\alpha_i = 1$, the corresponding cost is not triggered. However, whenever a cost is activated, its parameter has been estimated through 5-fold cross validation (with one fold for testing), as well as the θ smoothing parameter of the ranking cost ω_{rc}. Performances have been measured in terms of Precision at 1 (P@1), that is the percentage of correctly transcribed sentences occupying the first position in the rank, and Word Error Rate (or WER). All audio files are analyzed through the official Google ASR APIs [21]. In order to reduce the evaluation bias to ASR errors, only those commands with an available solution within the 5 input candidates were retained for the experiments.

A. Experimental Results

Table I shows the mean and standard deviation of the P@1 and the WER across the 5 folds. The results have been obtained by testing our cost function on the aforementioned HuRIC corpus. The transcription have been gathered in January 2016. The sizes of the GG, S4R and RC datasets were of 100, 97 and 112 utterances, each paired with 5 transcriptions derived from the ASR system.

TABLE I
RESULTS IN TERMS OF P@1 AND WER

	GG		S4R		RC	
	P@1	WER	P@1	WER	P@1	WER
ASR BL	74.00 ±6.52	3.66	84.71 ±7.57	2.61	79.55 ±10.66	3.89
Greedy	**94.79** ±0.12	4.33	93.58 ±4.43	1.09	79.30 ±7.96	5.00
ω_G	90.00 ±3.54	1.13	**93.98** ±6.36	0.89	78.64 ±9.59	3.92
ω_A	80.00 ±7.07	2.22	82.71 ±10.02	2.85	82.27 ±10.21	3.65
ω_E	78.00 ±5.70	2.97	83.66 ±6.04	3.00	**83.18** ±11.32	3.19
$\omega_{G,A}$	90.00 ±3.54	1.13	92.93 ±6.63	1.06	80.45 ±11.54	3.79
$\omega_{E,G}$	90.00 ±3.54	1.13	**93.98** ±6.36	0.89	82.27 ±10.71	3.23
$\omega_{A,E}$	83.00 ±2.74	1.94	86.72 ±5.42	2.21	**83.18** ±10.85	3.71
$\omega_{G,A,E}$	90.00 ±3.54	1.13	92.93 ±6.63	1.06	82.27 ±12.07	3.75

We compared our approach, where hypotheses are re-ranked according to our cost function $\omega(s)$, to two different baselines. In the first baseline (*ASR BL*), the best hypothesis is selected by following the initial guess given by the ASR, i.e., the transcription ranked in first position. The second baseline (*Greedy*) selects the first transcription, occurring within the list, that belongs to the language generated by the grammar. Conversely, the row ω_G refers to the cost function setting when α_A and α_E are set to 1, i.e., just the cost ω_G is actually triggered. In general, $\omega_{i,j,k}$ refers to the cost function when the costs ω_i, ω_j and ω_k are considered.

The *Greedy* approach seems to be effective when the sentences are more constrained by the grammar, i.e., it is likely that the correct transcription is recognized by the grammar. In fact, this approach is able to reach high scores of P@1 in both GG and S4R datasets, i.e., 94.79 and 93.58, respectively. Moreover, when the *complete-grammar cost* is triggered, i.e., ω_G, $\omega_{G,A}$ and $\omega_{G,A,E}$, we get comparable results, specially on the S4R dataset, with a relative increment of $+10.94\%$. These observations do not apply for the RC dataset, where the structures and lexicon of the sentences are not constrained by the grammar. In fact, the *complete-grammar cost* does not seem to provide any actual improvement.

Conversely, we observe a drop of performance when the full constrained grammar is employed, i.e., both *Greedy* and ω_G. On the other hand, when the *action-dependent* and *entity-dependent costs* are considered, we reach the best results. In particular, ω_E and $\omega_{A,E}$ are able to outperform both the *ASR BL* and the grammar constrained approaches. This behavior seems to depict a sort of *scaling-down* strategy: when the grammar does not fully cover the sentence, or it is not available, we can still rely on simpler, but more effective, information. Nevertheless, even though it does not perform the best, the strategy where all costs are triggered, i.e., $\omega_{G,A,E}$, seems to be the most stable across different sentence complexity conditions.

We conducted experiments on the transcription lists that have been employed in [14]. These have been gathered by relying on the same ASR engine, but almost two years earlier (May 2014). Hence, a different amount of sentences are employed in this experiment. In fact, the GG, S4R and RC

[1] http://sag.art.uniroma2.it/demo-software/huric/
[2] http://www.dis.uniroma1.it/~labrococo/?q=node/3

ISSN 2395-8618

TABLE II
RESULTS IN TERMS OF $P@1$ AND WER OBTAINED OVER DATA USED IN [14]

	GG		S4R		RC	
	P@1	WER	P@1	WER	P@1	WER
ASR BL	84.18 ±11.53	2.04	85.48 ±6.80	4.61	78.75 ±8.39	5.15
Greedy	**94.00** ±5.48	2.36	**95.78** ±5.79	0.62	74.96 ±5.33	5.80
ω_G	92.00 ±8.37	0.74	92.60 ±5.48	2.09	80.00 ±8.15	4.82
ω_A	86.00 ±13.42	1.47	85.48 ±6.80	4.30	82.50 ±6.85	2.69
ω_E	84.18 ±11.53	2.04	82.40 ±6.41	3.37	83.75 ±3.42	3.57
$\omega_{G,A}$	92.00 ±8.37	0.74	92.88 ±7.05	1.41	82.50 ±5.23	2.66
$\omega_{G,E}$	92.00 ±8.37	0.74	92.60 ±5.48	2.09	82.50 ±5.23	2.98
$\omega_{A,E}$	86.00 ±13.42	1.47	83.94 ±7.84	3.32	**90.00** ±8.39	1.85
$\omega_{G,A,E}$	92.00 ±8.37	0.74	92.88 ±7.05	1.41	83.75 ±3.42	2.66

datasets are composed of 51, 68 and 80 lists, respectively. The results are shown in Table II. We observe here similar trend, with both *Greedy* and *complete-grammar cost* reaching the highest scores in GG and S4R datasets. Even though the results obtained on these corpora are still comparable with the ones presented in [14], the interesting behavior observed on the RC dataset represents the main substantial difference. Even on this dataset, the trend seems to be the same, with the $\omega_{A,E}$ outperforming any other approach with relative improvements in P@1 up to $+20.06\%$. The trend of $\omega_{G,A,E}$ is confirmed here, making it the best solution as the most stable approach.

V. CONCLUSIONS

In this work, we presented a practical approach to increase the robustness of an *off-the-shelf* free-form Automatic Speech Recognition (ASR) system in the context of Spoken Language Understanding for Human-Robot Interaction (HRI), relying on grammars designed over specific domains. In particular, a cost is assigned to each ASR transcription, that decreases along with the number of constraints satisfied by the sentence with respect to adopted grammar. Despite to the simplicity of the proposed method, experimental results show that the proposed method allows to significantly improve a state-of-the-art ASR system over a dataset of spoken commands for robots.

Future work will consider the adoption of this re-ranking strategy within full chains of Spoken Language Understanding in the context of HRI, as the one presented in [5]. Moreover, the simple proposed method can be jointly used with supervised learning methods ([14]) that may exploit evidenced derived from the grammar to learn more expressive re-ranking functions.

REFERENCES

[1] J. Dowding, J. M. Gawron, D. Appelt, J. Bear, L. Cherny, R. Moore, and D. Moran, "Gemini: A natural language system for spoken-language understanding," AI Center, SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025, Tech. Rep. 527, Apr 1993, presented at the 31st Annual Meeting of the Association for Computational Linguistics, 22-26 June 1993, Columbus, OH.

[2] E. Bastianelli, D. Nardi, L. C. Aiello, F. Giacomelli, and N. Manes, "Speaky for robots: the development of vocal interfaces for robotic applications," *Applied Intelligence*, vol. 44, no. 1, pp. 43–66, 2015.

[3] G.-J. Kruijff, H. Zender, P. Jensfelt, and H. I. Christensen, "Situated dialogue and spatial organization: What, where... and why?" *International Journal of Advanced Robotic Systems*, vol. 4, no. 1, pp. 125–138, Mar 2007, special Issue on Human and Robot Interactive Communication.

[4] A. Hunt and S. McGlashan, "Speech recognition grammar specification," World Wide Web Consortium, Tech. Rep., 2004.

[5] E. Bastianelli, G. Castellucci, D. Croce, R. Basili, and D. Nardi, "Effective and robust natural language understanding for human-robot interaction," in *Proceedings of 21st European Conference on Artificial Intelligence*. IOS Press, 2014, pp. 57–62.

[6] G. Hinton, L. Deng, D. Yu, A. rahman Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. S. G. Dahl, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, November 2012.

[7] M. Doostdar, S. Schiffer, and G. Lakemeyer, *RoboCup 2008: Robot Soccer World Cup XII*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, ch. A Robust Speech Recognition System for Service-Robotics Applications, pp. 1–12.

[8] D. Linares, J.-M. Benedí, and J.-A. Sánchez, "A hybrid language model based on a combination of n-grams and stochastic context-free grammars," *ACM Transactions on Asian Language Information Processing*, vol. 3, no. 2, pp. 113–127, Jun 2004.

[9] R. Jonson, "Grammar-based context-specific statistical language modelling," in *Proceedings of the Workshop on Grammar-Based Approaches to Spoken Language Processing*, ser. SLP 2007, Stroudsburg, PA, USA, 2007, pp. 25–32.

[10] H. Li, T. Zhang, R. Qiu, and L. Ma, "Grammar-based semi-supervised incremental learning in automatic speech recognition and labeling," *Energy Procedia*, vol. 17, Part B, pp. 1843–1849, 2012, 2012 International Conference on Future Electrical Power and Energy System.

[11] R. de Mori, "Spoken language understanding: a survey," in *IEEE Workshop on Automatic Speech Recognition & Understanding, ASRU 2007, Kyoto, Japan, December 9-13, 2007*, S. Furui and T. Kawahara, Eds. IEEE, 2007, pp. 365–376.

[12] E. Bastianelli, G. Castellucci, D. Croce, L. Iocchi, R. Basili, and D. Nardi, "HuRIC: A human robot interaction corpus," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014)*. Reykjavik, Iceland: European Language Resources Association (ELRA), May 2014.

[13] F. Morbini, K. Audhkhasi, R. Artstein, M. Van Segbroeck, K. Sagae, P. Georgiou, D. Traum, and S. Narayanan, "A reranking approach for recognition and classification of speech input in conversational dialogue systems," in *Spoken Language Technology Workshop (SLT), 2012 IEEE*, Dec 2012, pp. 49–54.

[14] E. Bastianelli, D. Croce, R. Basili, and D. Nardi, "Using semantic maps for robust natural language interaction with robots," in *Sixteenth Annual Conference of the International Speech Communication Association*. International Speech Communication Association, 2015, pp. 1393–1397.

[15] M. Levit, S. Chang, and B. Buntschuh, "Garbage modeling with decoys for a sequential recognition scenario," in *IEEE Workshop on Automatic Speech Recognition Understanding, ASRU 2009*, 2009, pp. 468–473.

[16] S. Heinrich and S. Wermter, "Towards robust speech recognition for human-robot interaction," in *Proceedings of the IROS2011 Workshop on Cognitive Neuroscience Robotics (CNR)*, Sep 2011, pp. 23–28.

[17] Q. Lin, D. Lubensky, M. Picheny, and P. S. Rao, "Key-phrase spotting using an integrated language model of n-grams and finite-state grammar," in *EUROSPEECH*, G. Kokkinakis, N. Fakotakis, and E. Dermatas, Eds. ISCA, 1997.

[18] C. J. Fillmore, "Frame semantics and the nature of language," *Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech*, vol. 280, no. 1, pp. 20–32, 1976.

[19] L. C. Aiello, E. Bastianelli, L. Iocchi, D. Nardi, V. Perera, and G. Randelli, "Knowledgeable talking robots," in *Artificial General Intelligence - 6th International Conference, AGI 2013, Beijing, China, July 31 - August 3, 2013 Proceedings*, 2013, pp. 182–191.

[20] T. Wisspeintner, T. van der Zant, L. Iocchi, and S. Schiffer, "RoboCup@Home: Scientific competition and benchmarking for domestic service robots," *Interaction Studies*, vol. 10, no. 3, pp. 392–426, 2009.

[21] C. Chelba, P. Xu, F. Pereira, and T. Richardson, "Distributed acoustic modeling with back-off n-grams," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar 2012, pp. 4129–4132.

ISSN 2395-8618

Arquitectura de control abierta por medio de una PC para sistemas mecatrónicos

Miguel G. Villarreal-Cervantes, Daniel De-la-Cruz-Muciño,
Jesus Said Pantoja-García, and Gabriel Sepúlveda-Cervantes

***Resumen*—En este artículo se presenta el diseño de una arquitectura de control abierta por medio de una computadora personal (personal computer "PC" en inglés) para sistemas mecatrónicos de bajo costo, con la flexibilidad, reconfigurabilidad y versatilidad para realizar una amplia variedad de tareas de manera sencilla. Esta arquitectura se puede utilizar de forma didáctica en las escuelas para la enseñanza teórico-práctica en algunos cursos de ingeniería y posgrado; y además puede ser usada para la investigación al implementar diversas estrategias de control en el sistema real, para así reducir el tiempo empleado en la implementación experimental. Se muestra el desempeño de la arquitectura de control propuesta al comparar los resultados en simulación y experimental en un robot SCARA con un controlador par calculado para el seguimiento de trayectoria.**

***Palabras clave*—Arquitectura abierta, robot didáctico, robot SCARA, robótica, control de sistemas mecatrónicos.**

PC Based Open Control Architecture for Mechatronic Systems

***Abstract*—In this paper, an open control architecture for mechatronic systems is designed based on a personal computer (PC). This architecture is a low cost one with the flexibility, reconfigurability and versatility for carrying out a broad variety of tasks in a simple manner. This architecture can provide theoretical and practical teaching for some courses in engineering and postgraduate studies. In addition, this architecture can be useful in research for fast experimental implementation of diverse control laws. The simulation and experimental results show the performance of the open control architecture in a SCARA robot with a computed torque control for trajectory tracking.**

***Keywords*—Open architecture, didactic robot, SCARA robot, robotics, control of mechatronic systems.**

I. Introducción

ACTUALMENTE la demanda de sistemas mecatrónicos en procesos industriales, robótica de servicios, entre otros, está en constante crecimiento requiriendo de productos de mayor calidad a menor precio y a su vez con un incremento en su productividad. Con ésta idea, se han realizado esfuerzos por parte de investigadores para mejorar el desempeño de los robots, al desarrollar estrategias de control avanzadas, como por ejemplo, control adaptable y predictivo [1], control de fuerza [2], control por modos deslizantes [1], e incluso con estrategias de control inteligente como por ejemplo control por medio de una redes neuronales artificiales [3], [4] y en lógica difusa [5], [6]. Sin embargo la mayoría de estos esfuerzos se han quedado en prototipos de investigación debido a que los robots comerciales presentan limitaciones en su controlador haciendo más tardía su implementación industrial. Es así, que la arquitectura de control de sistemas mecatrónicos se puede clasificar en tres tipos [7], [8]: i) Arquitectura de propiedad (cerrada) en donde la estructura de control así como el protocolo propio de comunicación y detalles técnicos se encuentran ocultas para el usuario y es una tarea difícil o imposible integrar hardware (sensores) y hacer modificaciones en el sistema de control. ii) Arquitectura híbrida en donde la estrategia de control es cerrada pero algunos aspectos permanecen abiertos como la posibilidad de agregar nuevos dispositivos como sensores. iii) Arquitectura de control abierta en donde todos los aspectos de hardware y software pueden ser modificados sin dificultad por el usuario, tales como, sensores, interfaz gráfica de usuario, estrategias de control, etc. incrementando la capacidad y actualización del producto.

A pesar de que la tendencia actual es el diseñar sistemas de arquitectura abierta con el fin de tener un sistema flexible, reconfigurable, versátil, la mayoría de los controladores de robots comerciales (aunque se diga que es de arquitectura abierta), son del tipo cerrado o híbrido. Es así, que se han estado realizando esfuerzos para dotar de arquitectura de control abierta en robots o sistemas industriales. El sistema de arquitectura abierta se puede desarrollar por medio de sistemas embebidos de propósitos específicos o por medio de una computadora, siendo este último la tendencia actual ya que permite mucha mayor flexibilidad y versatilidad en la implementación de dicha arquitectura. A la arquitectura abierta que se desarrolla en una computadora se le nombra arquitectura abierta por medio de una computadora personal. En [9] se adapta un sistema de control de arquitectura abierta en un robot industrial PUMA 560 y se prueban diversas estrategias de control tales como control adaptivo a través de una regresor, controlador PD, entre otras. Se muestra que al

Manuscrito recibido el 6 de febrero de 2015, aceptado para la publicación el 12 de mayo de 2016, publicado el 30 de octubre de 2016.

Los autores se encuentran en el Instituto Politécnico Nacional, CIDETEC, LGAC de Mecatrónica, Departamento de Posgrado, Juan de Dios Bátiz s/n, C.P. 07700 D.F., México (correo electrónico: {mvillarrealc, ddelacruzm, jpantjag}@ipn.mx).

ISSN 2395-8618

implementar sistemas de control avanzados se puede mejorar en gran medida el desempeño de un robot industrial. En [10] se propone un sistema de control distribuido multi-agente de arquitectura abierta para una máquina de control numérico. El resultado obtenido muestra que es fácil implementar estrategias de control y características del monitoreo con el sistema propuesto.

Desde 1980 hasta el día de hoy, la filosofía de diseño de controladores de arquitectura abierta ha tomado grán atención, primeramente en el área de manufactura avanzada en el sistema de control de máquinas de control numérico (MCC). El primer controlador de arquitectura abierta fue el sistema MOSAIC desarrollado por la universidad de Nueva York en 1998 [11]. En 1992 se introduce uno de los sistemas de arquitectura abierta más importantes dentro del proyecto europeo OSACA [12]. Posteriormente en 1994 en Japón, se establece el proyecto OSEC bajo el consorcio de IROFA [13] y posteriormente varios investigadores de Estados Unidos adquieren un progreso sobresaliente en el ámbito de automatización de máquinas y control. Actualmente, en [14] se ha desarrollado una metodología para el diseño de sistemas de arquitectura abierta, que resulta en una arquitectura de hardware y software que es portable e integrada al sistema mecatrónico, de tal manera que se puedan implementar estrategias de control avanzada. En [15] se describe la arquitectura PC-ORC el cual es un sistema de control de arquitectura abierta en una computadora personal que se estableció de manera similar al modelo de referencia OSACA y se ha utilizado para realizar tareas básicas en un robot SCARA. La filosofía de arquitectura abierta se ha aplicado también a robots humanoides [16].

En este artículo se describe el diseño de un sistema de control de arquitectura abierta por medio de una computadora personal el cual proporciona a un sistema mecatrónico la flexibilidad, reconfigurabilidad y versatilidad para realizar una amplia variedad de tareas (dependiendo de la aplicación) considerando bajo costo, para así utilizarlo en las escuelas y/o investigación. Debido a que Matlab-Simulink es el desarrollador líder mundial de programas para la ingeniería, ciencia, gobierno y educación, así como por su fácil manejo, se propone como lenguaje de programación base para el desarrollo e implementación de las tareas a desarrollar por el sistema mecatrónico. Además se presentan los resultados experimentales en un sistema robótico de tres grados de libertad bajo el esquema de control par calculado, ilustrando el desempeño y beneficios de la arquitectura de control propuesta.

II. Arquitectura de control abierta

Siguiendo la filosofía descrita en [14] se establece la arquitectura de control abierta por medio de una computadora personal mostrada en la figura 1. Se observan varios niveles de jerarquía, entre los que se encuentran: i) El nivel de aplicación en donde se implementa la interfaz gráfica de usuario. ii) Nivel de control en donde se programa la estrategia de control. iii) Nivel de interfaz en donde se interpretan las variables físicas del sistema mecatrónico en variables que pueda interpretar el nivel de control y a su vez proporcionar la acción de control. iv) Nivel de dispositivos en donde se encuentran los sensores, actuadores así como acondicionamiento de señal.

Para el caso en particular de la arquitectura de control abierta por medio de una PC, en el nivel más bajo (nivel de dispositivos) están los sensores (decodificadores ópticos, sensor de fuerza, cámara, etc.), la etapa de potencia y los motores de corriente directa (CD) del sistema mecatrónico. La etapa de potencia consiste en el circuito integrado (C.I.) "$LMD18245$" que es un amplificador de potencia de puente completo de bajo costo y que presenta el modo de operación de par (corriente) que es muy importante para emplear el modelo estático del motor y así facilitar la simulación del control de sistemas mecatrónicos. Además el C.I. cuenta con una salida analógica que es proporcional a la corriente que circula por el motor y la cual servirá como sensor de corriente del motor. El nivel de interfaz consiste en los periféricos de entrada y salida que contiene una computadora normalmente (puerto serial, usb, video, audio, etc.) y en una tarjeta de entradas y salidas analógicas y digitales de la empresa Sensoray modelo 626 que se conecta a una PC por medio del puerto PCI (Peripheral Component Interconnect en inglés). El nivel de control será el programa que se instale en la PC y que coordine la entrada y salida de señales a través de la tarjeta Sensoray 626 por medio de una librería de enlace dinámico proporcionada por el fabricante. Para nuestro caso, se utilizó una PC Pentium IV a 3.2GHz con 4GB RAM en Windows y se escogió el programa de Matlab-Simulink en el nivel de control debido a que es un lenguaje gráfico, fácil de usar y que la mayoría de los estudiantes de licenciatura y posgrado están usando actualmente. Además, este programa puede implementarse en tiempo real con la librería "Real time workshop" con un tiempo máximo de muestreo de $1ms$. Así, en el nivel más alto (nivel de aplicación) se podrán crear interfaces gráficas para el usuario utilizando GUIDE (Graphical User Interface Development Environment) de Matlab.

III. Descripción del sistema mecatrónico

El sistema mecatrónico consiste de un robot SCARA o robot RRR de tres grados de libertad mostrándose en la figura 2. Tres servomotores Pittman son requeridos, dos de ellos de la serie $GM9234S033$ con una razón de reducción de $218.14:1$ para actuar los eslabones 1, 2 y uno de la serie $GM9236S015$ con una razón de reducción de $5.9:1$ para actuar el eslabón 3. Los tres cuentan con decodificadores ópticos de 500 pulsos por vuelta.

A. Modelo Cinemático

Considere el diagrama esquemático del robot SCARA mostrado en la figura 3 donde $O_i \ \forall \ i = 0, ..., 4$ es el origen del sistema de coordenadas del sistema inercial, del $i-ésimo$ eslabón y del efector final, respectivamente. La

ISSN 2395-8618

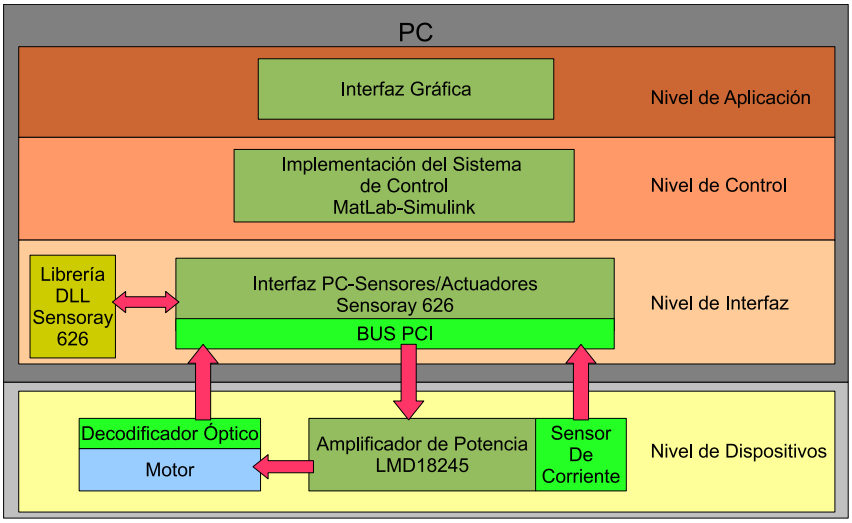

Fig. 1. Arquitectura de control abierta por medio de una PC.

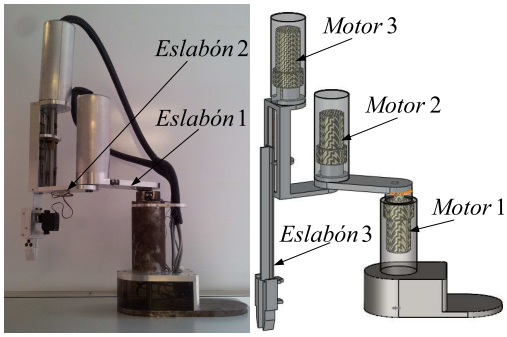

Fig. 2. Robot SCARA.

longitud, longitud del centro de masa, la masa y el momento de inercia expresada en el centro de masa del $i-ésimo$ eslabón se expresan como $l_i \in R$, $l_{c_i} \in R$, $m_i \in R$, $I_{zz_i} \in R^{3\times3}$ $\forall$ $i = 1, 2, 3$, respectivamente.

Definiendo las posiciones y velocidades deseadas en el espacio articular como $\bar{q} = \begin{bmatrix}\bar{q}_1 & \bar{q}_2 & \bar{q}_3\end{bmatrix}^T$ y $\dot{\bar{q}} = \begin{bmatrix}\dot{\bar{q}}_1 & \dot{\bar{q}}_2 & \dot{\bar{q}}_3\end{bmatrix}^T$ y empleando la convención de Denavit-Hartenberg [17], se obtiene la cinemática directa del robot (1)-(3), donde $p_r \in R$ es el paso de rosca, h_b, h_{F_z} son longitudes como se muestra en la figura 3 y $\bar{P}_x$, $\bar{P}_y$, $\bar{P}_z$ son las posiciones en el espacio Cartesiano transformadas del espacio articular.

$$\bar{P}_x = l_1 \cos q_1 + l_2 \cos(q_1 + q_2) \quad (1)$$

$$\bar{P}_y = l_1 \sin q_1 + l_2 \sin(q_1 + q_2) \quad (2)$$

$$\bar{P}_z = (h_b + h_{F_z} - l_3) - \frac{p_r}{2\pi} q_3 \quad (3)$$

Por el método geométrico [2] se calcula la cinemática inversa del robot SCARA presentado por las ecuaciones (4)-(6), donde $\sigma = \pm 1$, que indica las soluciones codo arriba

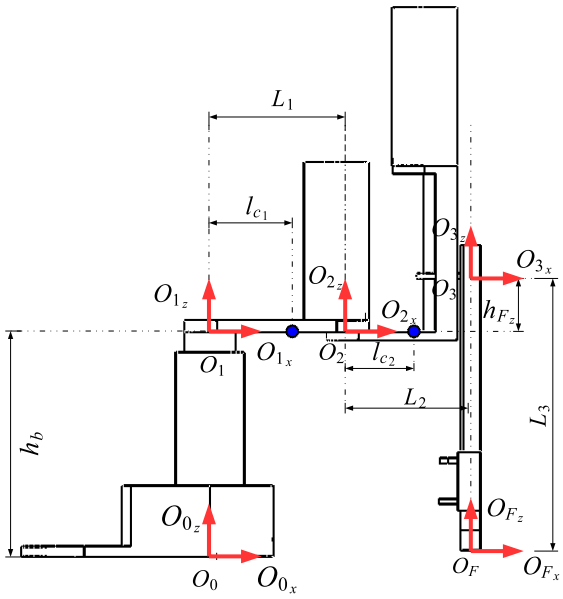

Fig. 3. Diagrama esquemático del robot SCARA.

$(+)$ o codo abajo $(-)$:

$$\bar{q}_1 = \text{atan2}\left(\frac{\bar{P}_y}{\bar{P}_x}\right) - \text{atan2}\left(\frac{\sigma l_2 \sqrt{4l_1^2 l_2^2 - \left(\bar{P}_x^2 + \bar{P}_y^2 - l_1^2 - l_2^2\right)^2}}{2l_1^2 l_2 + l_2\left(\bar{P}_x^2 + \bar{P}_y^2 - l_1^2 - l_2^2\right)}\right) \quad (4)$$

$$\bar{q}_2 = \text{atan2}\left(\frac{\sigma\sqrt{4l_1^2 l_2^2 - \left(\bar{P}_x^2 + \bar{P}_y^2 - l_1^2 - l_2^2\right)^2}}{\left(\bar{P}_x^2 + \bar{P}_y^2 - l_1^2 - l_2^2\right)}\right) \quad (5)$$

$$\bar{q}_3 = \frac{2\pi(h_b + h_{F_z} - l_3 - \bar{P}_z)}{p_r} \quad (6)$$

ISSN 2395-8618

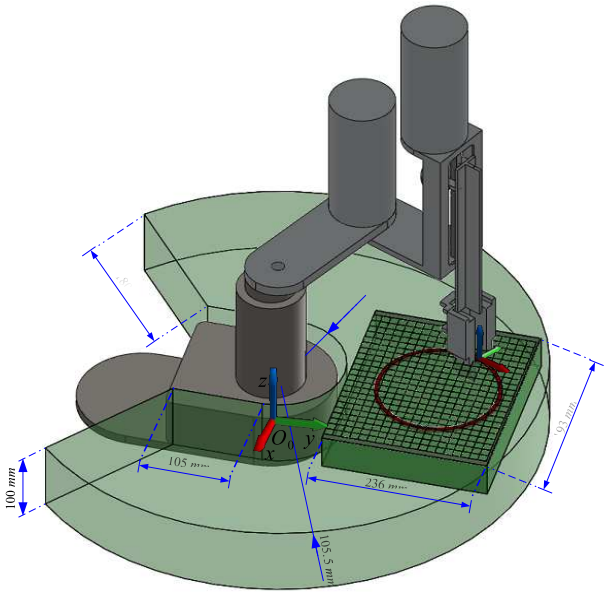

Fig. 4. Espacio de trabajo.

B. Espacio de trabajo

El espacio de trabajo real del robot SCARA se muestra en la figura 4. Para el caso particular se utilizó el espacio de trabajo representado con un prisma rectangular de $0.236\ m$, $0.193\ m$, $0.1\ m$ con un desplazamiento lineal sobre el eje O_{0_z} de $0.05\ m$.

C. Modelo dinámico

Considerando que el momento de inercia del eslabón 3 con respecto al eje z_3 es pequeño y que su movimiento es a través del plano $y-z$ o $x-y$, comparada con los momentos de inercia de los eslabones 1, 2 con respecto a los ejes $z_i\ \forall i = 1, 2$, respectivamente y a su plano de movimiento $x-y$, se propone desacoplar la dinámica del robot SCARA en un robot del tipo RR y otro del tipo R.

Siguiendo el formalismo de Euler-Lagrange [2] se puede obtener el modelo dinámico del robot SCARA desacoplado mostrado en (7).

$$\tau = \begin{bmatrix} \bar{M}_{11} & \bar{M}_{12} & 0 \\ \bar{M}_{21} & \bar{M}_{22} & 0 \\ 0 & 0 & \bar{M}_{33} \end{bmatrix} \ddot{q} + \begin{bmatrix} \bar{C}_{11} & \bar{C}_{12} & 0 \\ \bar{C}_{21} & \bar{C}_{22} & 0 \\ 0 & 0 & 0 \end{bmatrix} \dot{q} + \begin{bmatrix} \bar{\tau}_{f_1} \\ \bar{\tau}_{f_2} \\ \bar{\tau}_{f_3} \end{bmatrix} \tag{7}$$

donde:

$\bar{M}_{11} = m_2 l_1^2 + 2m_2 \cos(q_2) l_1 l_{c_2} + m_1 l_{c_1}^2 + m_2 l_{c_2}^2 + I_{zz_1} + I_{zz_2}$

$\bar{M}_{12} = \bar{M}_{21} = m_2 l_{c_2}^2 + l_1 m_2 \cos(q_2) l_{c_2} + I_{zz_2}$

$\bar{M}_{22} = m_2 l_{c_2}^2 + I_{zz_2}$

$\bar{C}_{11} = -2\dot{q}_2 l_1 l_{c_2} m_2 \sin(q_2)$

$\bar{C}_{12} = -\dot{q}_2 l_1 l_{c_2} m_2 \sin(q_2)$

$\bar{C}_{21} = \dot{q}_1 l_1 l_{c_2} m_2 \sin(q_2)$

$\bar{C}_{22} = 0$

$\bar{M}_{33} = I_{zz_3}$

D. Sistema de control

Se propone un controlador par calculado [2] para seguir una trayectoria, cuyo esquema es el mostrado en (8) donde $e_i = \bar{q}_i - q_i$ es el error de posición angular, $\dot{e}_i = \dot{\bar{q}}_i - \dot{q}_i$ es el error de velocidad angular, $\bar{q}_i$, $\dot{\bar{q}}_i$, $\ddot{\bar{q}}_i$ es la posición, velocidad y aceleración angular deseada del $i-ésimo$ eslabón, respectivamente, $k_{p_i} \in R$ y $k_{d_i} \in R$ es la ganancia proporcional y derivativa.

$$\begin{aligned} u_i &= \bar{M}_{i1} v_1 + \bar{M}_{i2} v_2 + \bar{M}_{i3} v_3 + \bar{C}_{i,1} \dot{q}_1 + \bar{C}_{i,2} \dot{q}_2 \\ v_i &= \ddot{\bar{q}} + k_{p_i} e_i + k_{d_i} \dot{e}_i \ \forall\ i = 1, 2, 3 \end{aligned} \tag{8}$$

Se escoge que el robot realice en el efector final un círculo en $120\ s$ de radio $0.07\ m$ en el plano x-y del espacio Cartesiano con centro en las coordenadas $(0m, 0.17m)$ y en el eje z que realice un movimiento oscilante alrededor de $0.10m$ con una amplitud de $0.04m$ y un periodo de $180s$, por lo que se propone la trayectoria descrita en (9)-(14).

$$\bar{P}_x = -0.08 sin(0.0524t) \tag{9}$$

$$\bar{P}_y = 0.17 + 0.08 cos(0.0524t) \tag{10}$$

$$\bar{P}_z = 0.1 + 0.04 cos(0.0349t) \tag{11}$$

$$\dot{\bar{P}}_x = 0.0042 cos(0.0524t) \tag{12}$$

$$\dot{\bar{P}}_y = -0.0042 sin(0.0524t) \tag{13}$$

$$\dot{\bar{P}}_z = -0.0014 sin(0.0524t) \tag{14}$$

IV. Resultados en simulación y experimental

Para realizar los resultados en simulación se requiere de los parámetros cinemáticos (Param. Cin.) y los parámetros dinámicos (Param. Din.) del robot SCARA. Dichos parámetros se obtienen con la ayuda del programa de diseño paramétrico SolidWorks y se muestran en la tabla I. En esa misma tabla se muestran las ganancias para el sistema de control (Gan. Ctrl.). Las ganancias fueron escogidas a prueba y error.

TABLE I
Parámetros del robot SCARA, ganancias del controlador y coeficientes de fricción de cada unión.

Param. Cin.	Param. Din.	Gan. Ctrl.
l_1= 0.152 m	l_{c_1}= 0.114 m	k_{p_1}= 10000
l_2= 0.1542 m	l_{c_2}= 0.105 m	k_{p_2}= 10000
l_3= 0.282 m	m_1= 1.472 Kg	k_{p_3}= 10000
h_b = 0.2841 m	m_2= 2.207 Kg	k_{d_1}= 140
h_{F_z} = 0.0546 m	$I_{zz_1} = 5E-3\ Kg\ m^2$	k_{d_2}= 140
$p_r = 2E-3\ m$	$I_{zz_2} = 3E-3\ Kg\ m^2$	k_{d_3}= 140
	$I_{zz_3} = 2.8E-4\ Kg\ m^2$	

Se utilizó el programa Matlab-Simulink para realizar la simulación numérica con el método de Runge-Kutta de 4to orden con un tiempo de integración $\Delta t_s = 5E-3s$, un tiempo final $t_f = 120s$ y un vector de condición inicial $q(t_0) = [0, 0, 0]^T$ ubicando al efector final en $(0.3062\ m,\ 0\ m,\ 0.0567\ m)$. Para realizar los resultados experimentales

ISSN 2395-8618

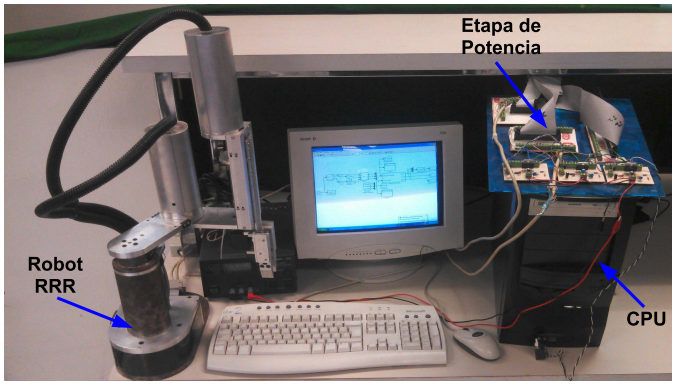

Fig. 5. Sistema experimental.

se consideró la misma condición inicial en el manipulador, se propuso un tiempo de muestreo de $\Delta t_e = 5E-3s$ y se requirió del modelo estático del motor ($\tau_i = k_{m_i} i_{a_i}$) para convertir el par obtenido por la estrategia de control en la corriente proporcional requerida por la etapa de potencia, donde $k_{m_1} = k_{m_2} = 2.13E-2\ \frac{Nm}{A}$, $k_{m_3} = 2.9E-2\ \frac{Nm}{A}$ e i_{a_i} son las constantes del motor y la corriente de armadura del $i-ésimo$ motor. Con el propósito de obtener de una manera más eficiente la relación entre voltaje de entrada V_{in_i} con la corriente de salida i_{a_i} en el i-ésimo amplificador de potencia "LMD18245", se propuso el valor de la resistencia $R_s = 5326.9\Omega$, se estableció al nivel alto las entradas digitales del convertidor digital a analógico $M1-M4$ (ver hoja de datos del circuito integrado) y se propuso la relación voltaje-corriente mostrada en (15), donde $a = 0.0296$, $b = 0.2618$ y $c = 0.0624$ para voltaje-corriente de salida positivo y $a = -0.0369$, $b = 0.2330$ y $c = -0.0398$ para voltaje-corriente de salida negativo. Esta relación se obtuvo a través de un método de regresión por mínimos cuadrados.

$$V_{in_i} = \frac{-b + \sqrt{|b^2 - 4a(c - i_{a_i})|}}{2a} \tag{15}$$

El sistema experimental con la arquitectura de control abierta por medio de una PC se muestra en la figura 5 y en la figura 6 se muestra el diagrama esquemático.

La trayectoria deseada en el espacio articular se calcula en línea. Los resultados en simulación y experimental en el seguimiento de la trayectoria del efector final se muestra en la figura 7 para el plano $x-y$ y en la figura 8 para el eje z. Se observa en la figura 9 que pasando el transitorio inicial, el máximo error en el efector final de la distancia del centro del círculo al radio del círculo de la trayectoria deseada es de $1E-8\ m$ en la simulación y de $9E-4\ m$ en los experimentos; y el error máximo de la trayectoria en z es de $2E-9\ m$ en simulación y de $7E-6\ m$ en los experimentos, los cuales son aceptables para efectos didácticos, de laboratorio y de investigación. Cabe mencionar que el error producido en el tiempo $60s$ de la figura 9a se debe a una falla mecánica la cual provoca que la fricción en el eje 2 sea mayor y su vez se incremente la señal de control, como se observa en la figura 10b. Dicho error puede disminuirse al aplicarle otra estrategia de control que compense las no linealidades producidas por la fricción. Así, en la figura 10 se muestra la señal de control requerida para seguir la trayectoria.

Es importante resaltar la correspondencia entre los resultados de simulación y experimentales, lo cual indica la importancia de modelar el sistema mecatrónico y probar estrategias de control primeramente en simulación y una vez que el desempeño del controlador sea el adecuado implementarlo físicamente, esto con el propósito de evitar daños en el sistema real.

Actualmente el sistema mecatrónico y la arquitectura de control abierta por medio de una PC se está utilizando en el curso de robótica de manipuladores y sistemas de control no lineal que se ofrecen en la Maestría de Tecnología de Cómputo en el Centro de Innovación y Desarrollo Tecnológico en Cómputo del Instituto Politécnico Nacional (CIDETEC-IPN), dando como resultado que los alumnos comprendan y apliquen de una mejor manera los conceptos proporcionados en clases.

A. Análisis comparativo de la arquitectura de control abierta por medio de una PC

Con el propósito de mostrar las bondades de la propuesta de arquitectura de control abierta por medio de una PC, se realiza una comparación con otras dos arquitecturas, una reportada en la literatura [18] y otra producida por la empresa Quanser. Para realizar una comparación justa con arquitecturas similares, se definió como criterio, que debería ser a través de una PC y con compatibilidad con Matlab-Simulink. La comparación se dirige principalmente en el costo.

En [18] se utilizan los productos de dSpace para realizar el enlace entre la computadora y el hardware (adquisición y envío de señales) por lo que además de necesitar un programa de la aplicación en Simulink, se requiere pasar dicho programa en la interfaz de usuario propia de la empresa dSpace (dSPACE CONTROLDESK). Las ventajas de utilizar los productos dSpace es su precisión y su alta tasa de muestreo. Sin embargo para algunas aplicaciones en sistemas robóticos, ésto podría resultar excesivo. Por otro lado, las tarjetas de adquisición de datos de la empresa dSpace normalmente tienen un costo elevado. En el trabajo presentado en [18] se utilizaron seis tarjetas de adquisición de datos (DS1004, DS1003, DS2001, DS2102, DS3002 y DS4001) por lo que el costo de dicha arquitectura es muy elevado a pesar de que no se está considerando el costo del robot.

El otro punto a comparar es con la empresa Quanser, que ha promovido sistemas mecatrónicos didácticos con diseños de arquitectura abierta compatibles con Matlab-Simulink. Ofrecen varios sistemas mecatrónicos cuyo precio es alto y en donde se incluye el robot. Los prototipos que ofrecen son muy básicos en su manufactura y el sistema de sensado. Sin embargo, ofrecen mucha documentación y prácticas de laboratorio muy didácticas, que las hacen muy atractivas en la academia.

ISSN 2395-8618

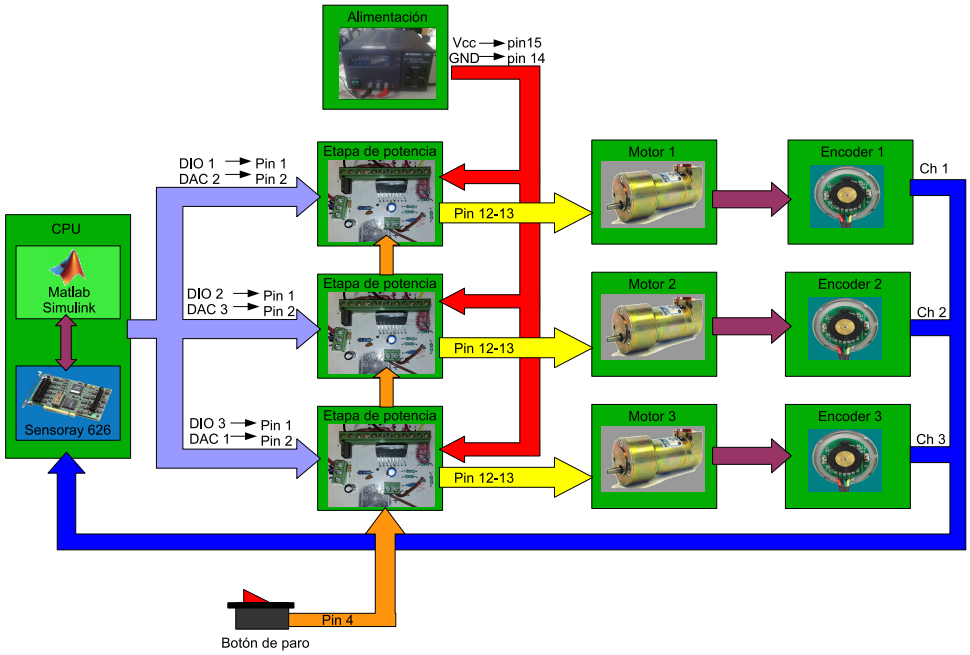

Fig. 6. Diagrama esquemático de la arquitectura de control abierta por medio de una PC.

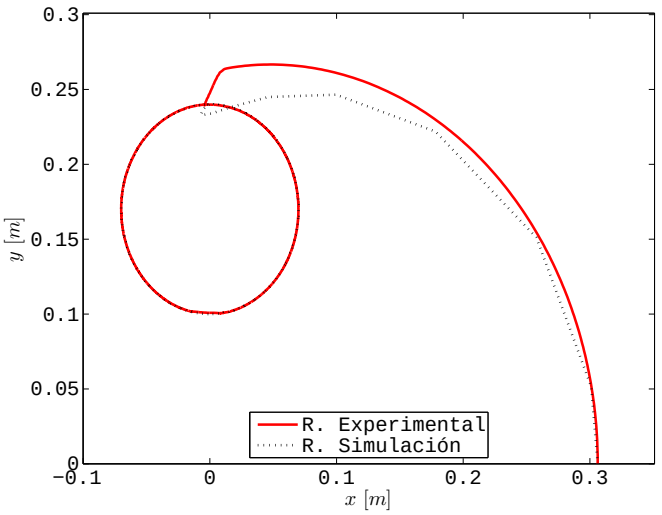

Fig. 7. Resultados en simulación y experimental del seguimiento de la trayectoria en $\bar{P}_x$ y $\bar{P}_y$ en el espacio Cartesiano.

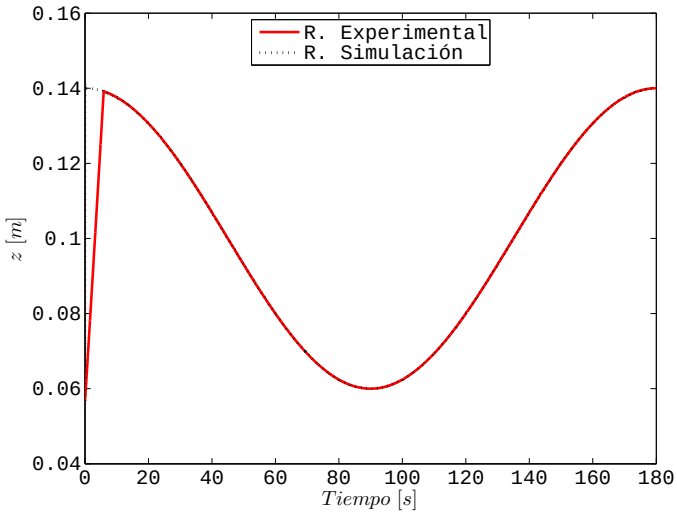

Fig. 8. Resultados en simulación y experimental del seguimiento de la trayectoria en $\bar{P}_z$ en el espacio Cartesiano.

La arquitectura de control por medio de una PC que se propone en este trabajo es muy económico (alrededor de $950.00 dólares estadounidense sin incluir el robot) que puede ser implementado fácilmente en cualquier laboratorio para validar la teoría de una forma rápida y con una precisión aceptable. Además, debido a que trabaja con Matlab-Simulink, se pueden realizar mejoras en el entorno gráfico. Finalmente, en la tabla II se muestra una tabla comparativa de la propuesta de arquitectura abierta presente en este trabajo con respecto a las otras dos arquitecturas mencionadas previamente. Se muestra con el símbolo † la cantidad de veces que se eleva el costo con respecto a la propuesta en este artículo. Se observa que la propuesta de arquitectura abierta mostrada en este trabajo realmente presenta un costo relativamente bajo.

V. Conclusiones

En este artículo se ha presentado una arquitectura genérica para el control de sistemas mecatrónicos por medio de una PC. Esta arquitectura es evaluada en un robot SCARA en donde se le implementó exitosamente una estrategia de control

ISSN 2395-8618

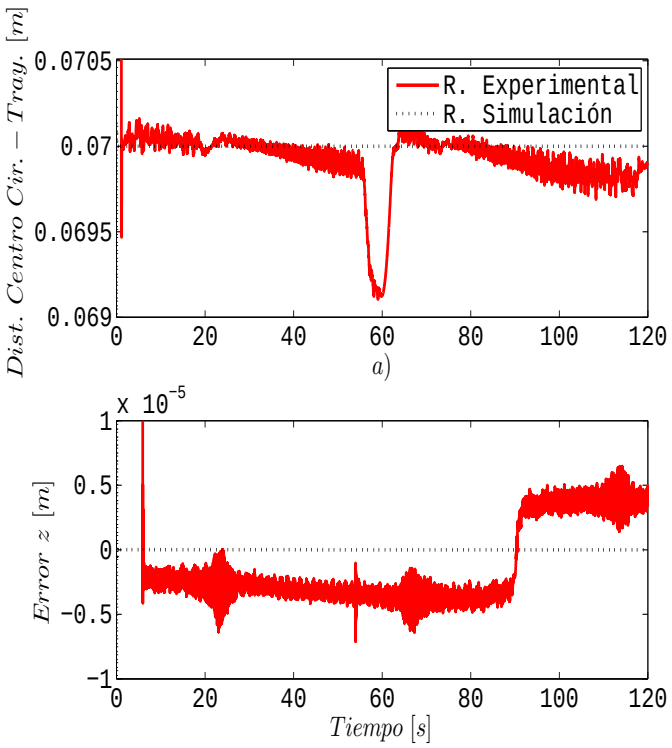

Fig. 9. Divergencia entre la distancia del centro de círculo y la trayectoria real - simulada con el radio del círculo ($0.07m$). Error en el seguimiento de la trayectoria $\bar{P}_z$.

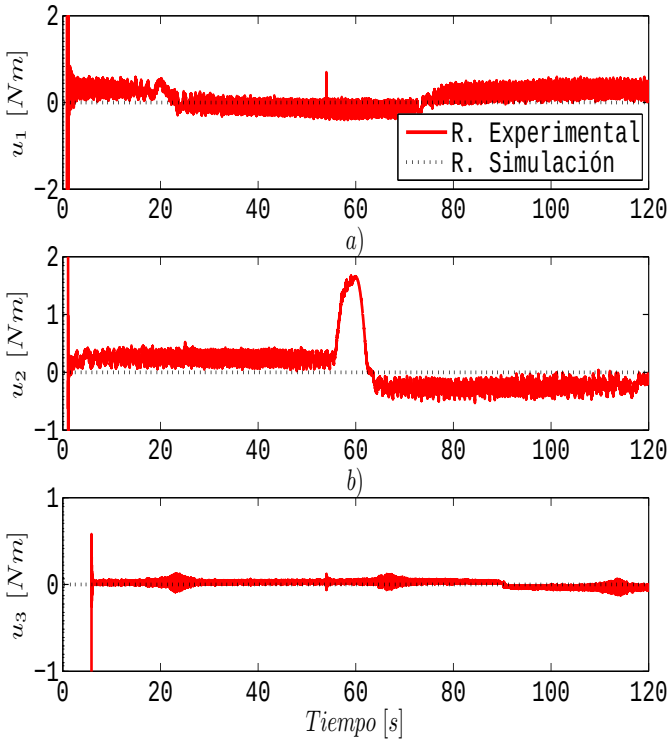

Fig. 10. Señal de control en simulación y experimental para seguir la trayectoria deseada.

TABLE II
TABLA COMPARATIVA DE LAS ARQUITECTURA ABIERTAS POR MEDIO DE UNA PC.

Arquitectura	Costo
Propuesta en este artículo	†
Propuesta en [18]	†††††
Empresa Quanser	††††

para el seguimiento de trayectoria. La arquitectura de control abierta hace posible que la etapa experimental sea más rápida, eficiente y con un costo muy bajo comparado con otras arquitecturas similares.

Por otra parte, en el ámbito académico y de investigación es muy útil esta arquitectura, debido a que se puede validar la teoría con la práctica, al verificar que los resultados en simulación numérica coinciden con un mínimo de error con los experimentales, siempre y cuando se identifiquen adecuadamente los parámetros del modelo del sistema mecatrónico a controlar.

Además es muy importante realizar primeramente las simulaciones numéricas del sistema a controlar antes de llevar a cabo los resultados experimentales para evitar daños en el sistema real debido a una mala sintonización del controlador.

AGRADECIMIENTOS

Este trabajo fué apoyado en parte por la Secretaría de Investigación y Posgrado del Instituto Politécnico Nacional (SIP-IPN) bajo el proyecto $SIP-20120663$ y en parte por el Consejo Nacional de Ciencia y Tecnología (CONACYT) bajo el proyecto 182298. Se agradece el apoyo del Centro Nacional de Actualización Docente (CNAD).

REFERENCES

[1] J. J. E. Slotine and W. Li, *Applied Nonlinear Control*. Prentice-Hall, 1991.

[2] M. W. Spong and M. Vidyasagar, *Robot Dynamics and Control*. John Wiley & Sons, 2004.

[3] J. Freeman and D. Skapura, *Neural networks algorithms, applications, and programming techniques*. Adison-Wesley Publishing Company, USA, 1991.

[4] M. T. Hagan, H. B. Demuth, and M. H. Beale, *Neural network design*. Martin Hagan, 1995.

[5] A. Nürnberger, F. Klawonn, K. Michels, and R. Kruse, *Fuzzy Control: Fundamentals, Stability and Design of Fuzzy Controllers*. Springer, 2006.

[6] A.-M. Zou and K. K. Dev, "Adaptive fuzzy fault-tolerant attitude control of spacecraft," *Control Engineering Practice*, vol. 19, no. 1, pp. 10–21, 2011.

[7] K. S. Fu, R. C. Gonzalez, and C. S. G. Lee, *Robotics: Control, sensing, vision and intelligence*. McGraw-Hill, 1987.

[8] W. E. Ford, "What is an open architecture robot controller," *IEEE International Symposium on Intelligent Control*, pp. 27–32, 1994.

[9] Y. Zhou and C. W. de Silva, "Real-time control experiments using an industrial robot retrofitted with an open-structure controller," *Systems, Man and Cibernetics*, vol. 4, pp. 553–559, 1993.

[10] L. Morales-Velázquez, R. de Jesus Romero-Troncoso, R. A. Osornio-Rios, G. Herrera-Ruiz, and E. Cabal-Yepez, "Open-architecture system based on a reconfigurable hardware–software multi-agent platform for CNC machines," *Journal of Systems Architecture*, vol. 56, pp. 407–418, 2010.

ISSN 2395-8618

[11] S. Schofield and P. Wright, "Open architecture controllers for machine tools, Part 1: Design principles," *Transactions of the ASME Journal of Manufacturing Science and Engineering*, vol. 120, no. 2, pp. 417–424, 1998.

[12] G. Haidegger and J. Nacsa, "Shop-floor communication with OSACA-compliant controllers," *IEEE International Workshop on Factory Communication Systems*, pp. 355–362, 1997.

[13] C. Sawada and O. Akira, "Open controller architecture OSEC-II: Architecture overview and prototype systems," *IEEE Symposium on Emerging Technologies and Factory Automation*, pp. 543–550, 1997.

[14] S. Hassan, N. Anwer, Z. Khattak, and J. Yoon, "Open architecture dynamic manipulator design philosophy," *Robotics and Computer-Integrated Manufacturing*, vol. 26, pp. 156–161, 2010.

[15] K.-S. Hong, K.-H. Choi, J.-G. Kim, and S. Lee, "A PC-based open robot control system: PC-ORC," *Robotics and Computer Integrated Manufacturing*, vol. 17, pp. 355–365, 2001.

[16] G. Metta, P. Fitzpatrick, and L. Natale, "YARP: Yet another robot platform," *International Journal of Advanced Robotic Systems*, vol. 3, no. 1, pp. 43–48, 2006.

[17] J. J. Craig, *Introduction to Robotics: Mechanics and Control*. Prentice Hall, 2004.

[18] K. K. Tan, K. Z. Tang, H. F. Dou, and S. N. Huang, "Development of an integrated and open-architecture precision motion control system," *Control Engineering Practice*, vol. 10, no. 7, pp. 757–772, 2002.

ISSN 2395-8618

Business Process Models Clustering Based on Multimodal Search, K-means, and Cumulative and No-Continuous N-Grams

Hugo Ordoñez, Luis Merchán, Armando Ordoñez, and Carlos Cobos

***Abstract*—Due to the large volume of process repositories, finding a particular process may become a difficult task. This paper presents a method for indexing, search, and grouping business processes models. The method considers linguistic and behavior information for modeling the business process. Behavior information is described using cumulative and no-continuous n–grams. Grouping method is based on k-means algorithm and suffix arrays to define labels for each group. The clustering approach incorporates mechanisms for avoiding overlapping and improve the homogeneity of the created groups using the K-means algorithm. Obtained results outperform the precision, recall and F-measure of previous approaches.**

***Index Terms*—Clustering, business process models, multimodal search, cumulative and no-continuous n-grams.**

I. Introduction

BUSINESS processes (BP) are composed of related and structured activities or tasks that contribute to a business goal. Consequently, BP models allow representing and documenting and sharing companies' internal procedures. These models may be useful also to guide the development of new products and support improvement of processes. Notwithstanding the advantages of BP models, the management of its repositories may become a big challenge. The latter is because commonly these repositories store hundreds or even thousands of BP models, that in turn are made up of tens or hundreds of elements (tasks, roles and so on) [1]. As a result, to find a particular BP matching specific requirements may become a complex task.

Manuscript received on August 12, 2016, accepted for publication on October 4, 2016, published on October 30, 2016.

Hugo Ordoñez and Luis Merchán are with the Facultad de Ingeniería, Universidad de San Buenaventura, Cali, Colombia, and the Laboratorio para investigación en desarrollo de la ingeniería de software (LIDIS) of the Universidad de San Buenaventura, Colombia (e-mail: hugoeraso@gmail.com, lmerchan@usbcali.edu.co).

Armando Ordoñez is with the Facultad de Ingeniería, Fundación Universitaria de Popayán, the group Intelligent Management Systems, Popayán, Colombia, (e-mail: armandoordonez@gmail.com).

Carlos Cobos is with the Departamento de Sistemas, Facultad de Ingeniería Electrónica y Telecomunicaciones, Universidad del Cauca, Colombia (e-mail: ccobos@unicauca.edu.co).

Most of the existing research approaches for BP search are based on typical measures such as linguistics, structure, and behavior. However, other techniques from the field of Information Retrieval (IR) may be applied to improve existing results. Among these IR techniques, the multimodal search reports good results among users; this is in part because multimodal search combines different information types to increase the accuracy of the results [2]. Moreover, clustering techniques have been used in BP search to improve the results display. Clustering techniques create groups of the BPs obtained from the query. These groups are created based on the similarity of the BPs.

This paper presents an approach for clustering of BP models based on multimodal search and cumulative and no-continuous n-grams. Cumulative and no-continuous n-grams allow us to analyze more linguistic information that traditional n-grams. These n-grams are built following a tree shaped path based on syntactical information. This method allows reviewing the branches of the tree [3]. This approach unifies linguistic and behavioral information features in one search space while takes advantage of the clustering techniques for improving results display, thus giving users an clear idea of the retrieved BP [4].

Firstly, this approach includes an indexing and search method based on a multimodal mechanism that considers two dimensions of the BP's information. 1) linguistic information which includes names and descriptions of BP elements (e.g. activities, interfaces, messages, gates, and events). And 2) behavior information represented as codebooks (text strings) which include all the structural components representing sequences of the control flow (i.e. the union of two or more components of the control flow simulates the formation of cumulative and no-continuous N-grams) [5], [6]. Secondly, the present approach includes a technique for grouping BP based on affinity. This grouping uses a clustering technique based on the both dimensions of the BP aforementioned.

The present approach is based on a multimodal mechanism previously described in [4] and introduces improvements in two areas. By using cumulative no-continuous n-grams, more elements of control flow may be represented and analyzed during the search and indexing process. In addition, the

ISSN 2395-8618

clustering mechanism was improved to avoid overlapping (BP results can not belong to many groups simultaneously) and increase the homogeneity of groups. The latter improvements were achieved by implementing k-means algorithm, and by performing more iterations of the algorithm for selecting the best group. Finally clusters are tagged (labeled) based on their functionality using a Suffix Array algorithm.

The evaluation of the proposed approach was done using a BP repository created collaboratively by experts [7]. The results obtained using the present approach, were compared with the results of other state-of-the-art algorithms. This comparison was performed using measures from information retrieval domain.

The rest of this paper is organized as follows: Section 2 presents the related work, Section 3 describes the proposed approach, Section 4 presents the evaluation, and finally Section 5 describes conclusions and future works.

II. Related Works

The proposed approach is focused on two strategies: searching and clustering of BP models. This section presents main research works on both strategies.

Regarding searching, most of the existing approaches for BP search are based on measures such as linguistics, structure, and behavior. Linguistic-based approaches use, for example, the name or description of activities or events. Later during the search process, some techniques are used, such as space-vector representation with a frequency of terms (TF), and cosine similarity to generate the rankings of results [8]. Approaches based on association rules analyze previous executions of business processes using log files. During the search, activity patterns and phrases related to business process activities are identified using domain ontologies. Besides, in order to create a list of results, a heuristic component that determines the frequency of detected patterns is employed [9]. Approaches based on genetic algorithms use formal representations (graphs or state machines) of BPs and include data such as the number of inputs and outputs per node, edge labels, nodes name or description. Although this method may achieve precise results, execution time may be very high [10]. Most of the works in this area merely match inputs and/or outputs, using textual information of BP elements.

Regarding clustering, approaches may be classified into hierarchical and partitional clustering. Hierarchical clustering builds a hierarchy of groups based on structural and behavioral similarity of BP. These proposals allow users to review the hierarchy and choose a set with greater similarity according to their criteria [11], [12], [13]. Partitional clustering uses log files containing previous executions of BP. In this case the clustering algorithm groups BP with similar behavior based on the control-flow and data-flow found in their log files [14], [15], [4].

Unlike these approaches, the present approach uses a multimodal representation of BP models. Equally, clustering techniques are used to improve the results display. This clustering is based on grouping similar BP in the same group, which facilitates the display of results obtained from in a query.

III. Searching and Clustering of BP models

The main tasks of the present approach are i) Indexing (to include a new BP into the repository) and ii) search and clustering (to search BPs similar to the user query). Next, both tasks are described.

A. *Indexing task*

This task uses business rules to manage and pre-process BP models before indexing and storing in the repository. This task includes textual processing and generation of a search index. Next, the modules responsible for implementing these rules are described (Fig. 1), namely: 1) Parser and 2) Indexing and Weighting.

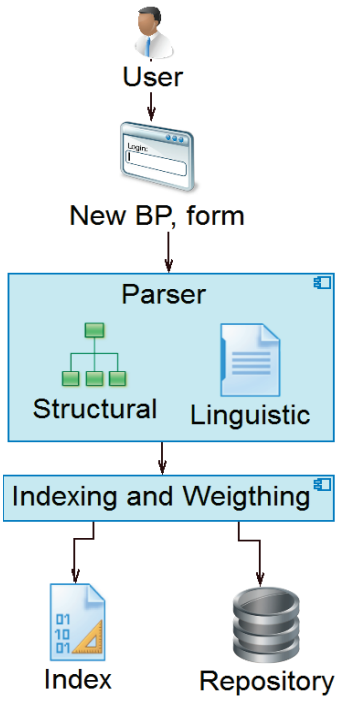

Fig 1. Indexing task: Include a new BP in the repository

Parser: The parser implements an algorithm that takes a BP described in BPMN notation and builds linguistic and structural components (codebook), this component also generates a search index consisting of two arrays by each BP model: an array MC of textual features and another *MCd* of structural components. The algorithm is described below.

Formation of linguistic component (Linguistic): Then the algorithm takes each *BPi*, extracts its textual characteristics Ct

ISSN 2395-8618

(activity name, activity type, and description) and forms a vector $Vtc_i = \{Ct_{i,1}, Ct_{i,2}, ..., Ct_{i,l}, ..., Ct_{i,L}\}$, where L is the number of textual characteristics found in BP_i. at this point, traditional pre-processing task area applied to textual components, namely, tokenize, lower case filtering, stop words removal, and stemming. For each vector Vtc_i, which represents a BP_i, a row of matrix MC_{il} is constructed. This row contains the linguistic component of all *BPs* stored in the repository. In this matrix, array i represents each *BP* and l a textual characteristic for each of them.

Formation of codebook component (Structural): A codebook *Cd* is a set of N structural components describing one or more nodes of the *BP* in the form of text strings. The set of codebooks formed from the whole repository is called the codebook component matrix. This matrix is formed by taking each tree that represent each *BP* in the repository. For example, Fig 2, shows a fragment of a BP_i with its activities. Each activity is represented with a text string defining the node type (StartEvent, TaskUser, TaskService). The node type refers to the functionality of each activity within the *BP*.

Codebooks are formed simulating the technique of traditional n-grams. These codebooks are sequences of textual elements: words, lexical item, grammatical labels, etc. arranged according to the order of appearance in the analyzed information. This method differs from previous works where traditional n-grams are formed with two components (N = 2, bigrams) [4]. Aditionally, in the present approach, the representation includes cumulative and no-continuous n-grams with N = 1 (unigrams), N = 2 (bigrams), N = 3 (trigrams) and so on until a maximum value of N = M. N-grams had shown to be convenient for the tree based representation of business processes. Next, a sample of BP is shown in Fig 2, and then in Table 1 the correspondence between activities of the BP in Fig. 2 and their node types are presented. Next, all codebooks for the BP are shown.

TABLE 1.
EXAMPLE OF THE ACTIVITIES OF THE BP IN FIGURE 2 AND THEIR TYPES

Activity	Type
Start	StartEvent
Evaluate clients payment	TaskUser
Route	RouteParallel
Client status report	TaskService
Recalculating debt	TaskScript

A codebook of n-grams representing the process described in Fig 2 is composed of n-grams which vary in size from 1 to 4 (M=4). n-grams of size 1 are: {*StartEvent, TaskUser, RouteParallel, TaskService, TaskService*}, on the other hand, n-grams of size 2, 3 and 4 are formed as described in Fig. 3, 4, and 5.

In Fig. 3, {StartEvent_TaskUser1, TaskUser_Route Parallel2, RouteParallel_TaskService3, RouteParallel_Task Scrip4}, where StartEvent_TaskUser1 corresponds to the concatenation of Star Event with the Evaluate clients payment activity, similarly to the other components.

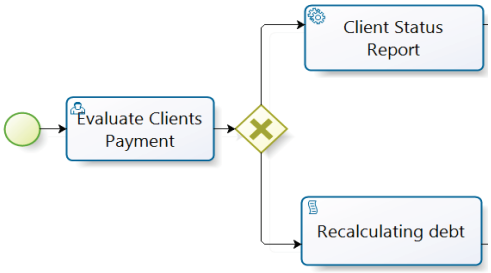

Fig 2. Types of component in Business process

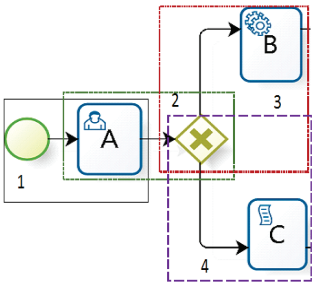

Fig. 3. Size-2 codebook

In Fig 4. {StartEvent_TaskUser_RouteParalle1, Task User_RouteParallel_TaskService2, TaskUser_RouteParallel_ TaskScrip3}.

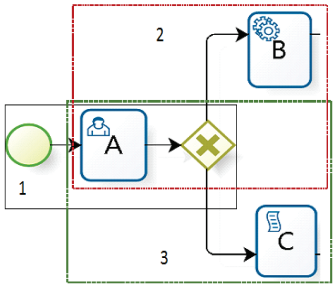

Fig 4. Size-3 codebook

As can be seen, as n-gram size grows a bigger part of the sequential path is covered (by concatenating its components), for example, in logical gates there exist bifurcation. As shown in Fig 4, in the codebook 2 the bifurcation goes from activity A to activity B, consequently, according to the property of cumulative and non continuos n-grams [3], it is possible to form the codebook 3 from activity A to Activity B.

Fig 5, {StartEvent_TaskUser_RouteParalle_Task Service1, StartEvent_TaskUser_RouteParalle_TaskScrip2}.

As can be observed, the cumulative and non- continuous n -grams can cover a significant part of the tree representing the

ISSN 2395-8618

semantic behavior of BP. The latter demonstrates that the control flow of the BP can be fully analyzed.

Unlike traditional n-grams, codebooks formed by cumulative and no-continuous n-grams provide a better and higher representation of processes control flow and behavior semantics. These codebooks allow better representation of the BPs as they are formed by joining control flow sequences. Behavior semantics of business processes describes the activities and its execution order [12]. It is important to note that as codebooks increase in size, they represent better the behavior semantics of processes.

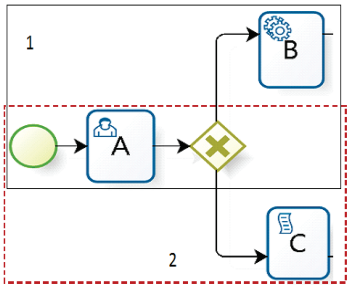

Fig 5. Size-4 codebook

Finally, the codebooks vector for sample BP of Fig. 2 is Vcd_i = {StartEvent, TaskUser, RouteParallel, TaskService, TaskService, StartEvent_TaskUser, TaskUser_RouteParallel, RouteParallel_TaskService, RouteParallel_TaskScrip, Start Event_TaskUser_RouteParalle, TaskUser_RouteParallel_Task Service, TaskUser_RouteParallel_TaskScrip, StartEvent_Task User_RouteParalle_TaskService, StartEvent_TaskUser_Route Paralle_TaskScrip}.

The cumulative and no-continuous n-grams concept can be used for terms (linguistic features) presented in BPs, but in this proposal, they just were used for the behavioral features (control flow).

Indexing and weighting: In this component, the linguistic and codebook components are weighted to create a multimodal search index *MI* composed of the matrix of the linguistic component (*MC*) and the codebook component matrix (MCd) i.e. $MI = \{MC_d \cup MC\}$. The index also saves the reference to the physical file of each of the models stored in the repository.

Weighting: Next, this component built the term by document matrix applies a weighting scheme of terms similar to that proposed in the vector space model for document representation, this approach is described elsewhere [16][17]. This weighting scheme is based on the original proposal of Salton [18].

B. Search task

This task is responsible for allowing users to conduct BP searches using three query options: linguistic, codebook, and multimodal (see Fig. 6). Each query is represented using a terms vector q = {$t1$, $t2$, $t3$, ..., t_j, ..., t_J}. The same pre-processing mechanism applied in the indexing task (parser) is applied to the BP query, thus obtaining the terms of the query vector reduced to their lexical root and the cumulative and continuous n-grams of the query.

Query forms: In this component, the user has forms that correspond to the graphical user interface (GUI). These forms allow selecting the search options and displaying the lists of the results and the created groups.

Conceptual ratings: This component sorts and filters the BP retrieval from the search. The ordering is done using one adaptation of the equation of conceptual score (used by Lucene library) [19].

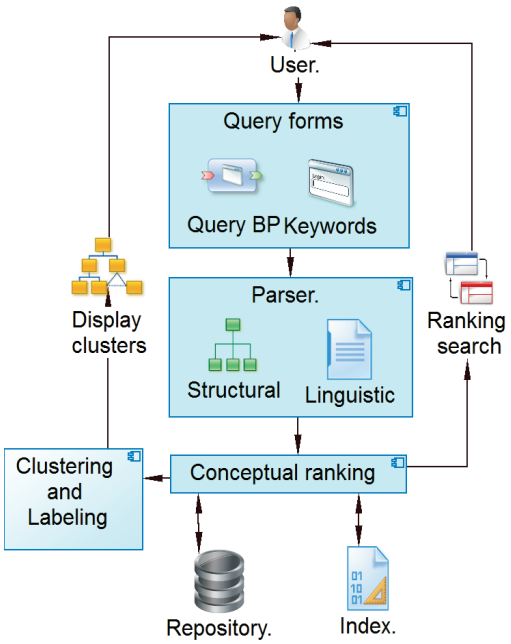

Fig 6. Searching and clustering task

List of results (Ranking): this component shows the results of the search to the user, in order to be analyzed.

Clustering process: Once the results are ranked, they are grouped using the K-means clustering algorithm [20]. Thus, the results are organized in groups of BP which are correlated according to textual and structural features.

K-means: The algorithm receives as input the number of groups (k-clusters, for performing the experiments in the assessment, we use values of k between 4 and 5 based on the recommendation presented in [13] to be formed). Then, k BPs are randomly selected to represent the starting centroids of each group. Later, each BP in the result list is assigned to the closest cluster centroid according to a distance function (where the most used one is the cosines similarity). For each one of the formed groups, the centroid of all these BP is

ISSN 2395-8618

calculated. Centroids are taken as new centers of their respective groups. The steps of the K-means algorithm are described below:

- *Step 1*: the algorithm selects k BP to be used as initial centroids (k is the number of groups to be formed);
- *Step 2*: each BP is added to the group with the highest similarity or proximity;
- *Step 3*: the algorithm calculates the centroid of each group to be new centroids.
- *Step 4:* if a convergence criterion is not reached return to step 2. For example, if the classification of BP is not changed.

Labeling: most of the clustering algorithms create groups without labels that allow identifying its content. Conversely, the present approach adopted a labeling method based on Suffix Arrays to determine the content of each group created (i.e., related to the purpose or functionality of BP models) and to ease user's interaction with the results. Thus, users may get a better idea of groups to review.

The labeling process starts creating a snippet (*S*) using tasks names. These tasks describe functionalities of BP models that compose the group to be labeled. Subsequently, chain *S* is pre-processed and converted to lowercase. Later, special characters and empty words are removed from *S*. Finally, an array of suffixes *As* is created. This array is ordered lexicographically to find most common phrases in *S* that identify the group content.

In the labeling algorithm, *S* is processed as a character set $S = \{s_1, s_2, s_3, s_n\}$. From this set, a new set $S'[i\ ,j]$ is formed, i.e., an array of sub-strings of S, which runs from index *i* to index *j*. After that, an array of integers *As* is created containing initial positions of suffixes in *S* ordered lexicographically. Then, *As*[*i*] stores the starting position of the *i*-th smallest suffix in S. Afterwards, the array of substrings *S´* is traversed using a binary search that aims to find the most common and with higher length suffix. The search starts with a separator of terms (in the case we have used the character $) and the subsequently found suffix is returned for labeling the BP group.

Display clusters: This component displays the formed groups in organized and structured way. This structure enables users to review and select the group with higher similarity with the query.

IV. Evaluation and Results

Results obtained using the present approach, were compared with the results of the manual evaluation performed on a closed test set, which is presented in [7]. This closed test set was created collaborative by 59 experts in business process management. In addition, the results of the present clustering multimodal (from now on *N-gramClusterBP*) approach were also compared with the results of grouping of the *MultiSearchBP* model [21] (from now on *LingoBP*) and *BPClustering* for grouping [22] (from now on *HC*). *LingoBP* uses two component, firstly a multimodal search based on by-grams (n = 2) and then clustering based on the Lingo algorithm. *HC* uses cosine coefficient to measure the similarity between two process models and implements an agglomerative hierarchical algorithm for clustering.

The evaluation was conducted in two phases: 1) internal assessment and 2) external assessment.

The first phase involves the application of internal metrics for clustering analysis that do not require human intervention. These metrics are used to identify how close or distant BPs are from each other in the formed groups. The used metrics are described below.

Sum of squares Between clusters (*SSB*)*:* this measures the separation between clusters (high values are desired). In Equation 3, *k* is the number of clusters, n_j is the number of elements in the cluster *j*, c_j is the centroid of cluster *j, and x* is the mean of the data set [23]:

$$SSB = \sum_{j=1}^{k} n_j\ dist\ (c_j\ -\ \bar{x})^2. \quad (1)$$

Sum-of-squares within cluster (SSW): this measures the variance (low values are desired) within groups, based on each of the existing elements in each group [23]:

$$SSW = \sum_{i=1}^{k} \sum_{x\ \epsilon c_i} dist\ (m_i, x)^2, \quad (2)$$

where *k* is the number of clusters, *x is* a point in the cluster c_i and m_i is the centroid from cluster c_i.

Table 2 shows the results of the internal evaluation. Regarding *SSB*, *N-gramClusterBP* reached an average value of 0.510. This result evidence the high separation of the created groups since the elements are assigned to the cluster having higher similarity and the intermediate elements between groups are removed. *N-gramClusterBP* outperforms *HC* in 0.09 and outperforms *LingoBP* in 0.14. Regarding *SSW*, elements variation between groups created using *N-gramClusterBP* is low. This good result show that BPs in the same group share similarly textual and structural features.

TABLE 2.
Results of internal assessment of the grouping

Algorithm	SSB	SSW
N-gramClusterBP	**0,510**	**0,048**
LingoBP	0,350	0,070
HC	0,420	0,065

The second phase, external assessment is focused on the quality of clustering by comparing groups created by automatic grouping techniques with groups generated by domain experts.

ISSN 2395-8618

In this phase, metrics such as weighing precision, weighing recall, and weighing F-measure were used. To evaluate weighing precision (Equation 3), weighing recall (Equation 4) and weighing F-measure (Equation 5), the groups' set $\{C_1, C_2, \ldots, C_k\}$ automatically created with evaluated approaches were compared with the ideal groups' collection $\{C_1^i, C_2^i, \ldots, C_h^i\}$ generated collaboratively by experts [24]. During assessment, the following steps were performed: (a) for each group C_n^i in the ideal set, a group C_m was found in the automatically generated set which most closely approximates to the first group. Later, the following metrics are calculated: $P(C, C^i)$, $R(C, C^i)$ and $F(C, C^i)$ as defined in Equations 6, 7 and 8; (b) to calculate the weighting precision, weighting recall and weighting F-measure based on Equation 8.

$$P(C, C^i) = \frac{|C \cap C^i|}{|C|} \quad (3)$$

$$R(C, C^i) = \frac{|C \cap C^i|}{|C^i|} \quad (4)$$

$$F(C, C^i) = \frac{2P(C,C^i)R(C,C^i)}{P(C,C^i)+R(C,C^i)} \quad (5)$$

$$P = \frac{1}{T}\sum_{j=1}^{h}|C_j^i|\,P(C_m, C_j^i) \quad (6)$$

$$R = \frac{1}{T}\sum_{j=1}^{h}|C_j^i|\,R(C_m, C_j^i) \quad (7)$$

$$F = \frac{2PR}{P+R};\ T = \sum_{j=1}^{h}|C_j^i| \quad (8)$$

In Equation 8, C is a group of BP models, C^i is a group from the ideal set. Fig 7 shows the average values of Precision, Recall, and F-Measure for the assessment of groups created using of *N-gramClusterBP, LingoBP* and *HC*.

Regarding the precision, best results were achieved with *N-gramClusterBP*. This algorithm increases precision by 16% compared with (*LingoBP*) and 12% compared to *HC*. This result is due to the high number of similar elements of the control flow and textual information that can be found both in the groups generated using *N-gramClusterBP* and in the ideal set. Moreover, the combination of structural and textual information used in *N-gramClusterBP* allows creating groups with greater similarity with the groups created by experts. The latter occurs because human experts consider several data types existing in the *BPs*.

On the other hand, groups formed by the (*LingoBP*) contain shared *BPs*, i.e., *BP* that belong to various groups, which increases the number of Falses Negative (NF) (*BPs* placed in groups different to the one that was expected) and consequently the *Pg* was reduced. Regarding *HC*, the values are explained by the fact that only structural information was used during the formation of groups. Besides, this formation of groups is done statically, that is, one *BP* is assigned to one group and cannot be assigned to another group with higher similarity in a posterior iteration.

Regarding Recall, *N-gramClusterBP* increases Recall by 5% in comparison to *LingoBP* and 3% in comparison to *HC* (*N-gramClusterBP* 47%, *LingoBP* with 44% and *HC* with 45%). The Recall value reached shows that more elements in the groups created with *N-gramClusterBP* were placed in the same groups that the manual (ideal) grouping. The latter can be explained by the absence of overlapping (*BPs* existing in many groups simultaneously) in *N-gramClusterBP*. As a result, *N-gramClusterBP* approach reduces the false negatives (FN) as the groups are assigned to the groups with higher similarity. Conversely, in *LingoBP* the number of elements per group decreases the value of true positives (TP) (*BPs* in the same group which was created by the manual grouping). Regarding F-measure, *N-gramClusterBP* (57%) achieves 12% more than *LingoBP* and 9% more that *HC*. The latter allows inferring that the created groups are more relevant and similar to groups created manually by the experts.

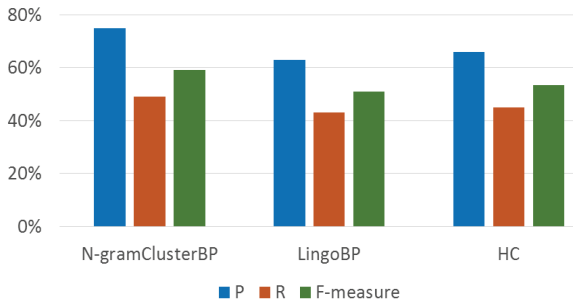

Fig 7. Results in the external clustering evaluation

V. Conclusions and Future Work

This paper presents an approach for improving the recovery, and clustering of business processes (BP) presented in [4]. The presented approach uses a multimodal search method based on cumulative and no-continuous n-grams of behavioral (structural) features. The use of textual information and structural information in multimodal index offers greater flexibility and precision in queries.

Results of the internal assessment show that using textual and structural information offers more compact BP groups because elements in the same group share diverse features. Moreover, by eliminating the overlapping (BP Models that may exist in several groups at the same time), N-gramClusterBP creates groups with more similar elements and also provides greater separation between the created groups.

The grouping process using N-gramClusterBP showed a high similarity (75% of precision) with the grouping performed by experts. This similarity is higher that the similarity achieved by LingoBP (63%) and HC (66%). This result can be explained by the absence of overlapping (elements in many groups simultaneously) and high refinement of groups (by performing iterations for assigning the most similar group) in N-gramClusterBP.

Future work includes adding specific domain ontologies to the proposed model; this will make possible including semantics to the search process, achieving more precise results. Equally, future work will be focused on the

ISSN 2395-8618

assessment of labeling method to determine if created labels help users to identify more easily information and functionality in the created groups. Finally, a hierarchical clustering method will be incorporated to create categories and subcategories of existing BP models in the repository.

REFERENCES

[1] M. La Rosa, "Detecting approximate clones in business process model repositories," *Information Systems*, vol. 49, pp. 102–125, 2015.

[2] J. C. Caicedo, J. Ben Abdallah, F. A. González, and O. Nasraoui, "Multimodal representation, indexing, automated annotation and retrieval of image collections via non-negative matrix factorization," *Neurocomputing*, vol. 76, pp. 50–60, Aug. 2012.

[3] G. Sidorov, "N-gramas sintacticos no-continuos," *Polibits*, no. 48, pp. 69–78, 2013.

[4] H. Ordoñez, J. C. Corrales, and C. Cobos, "Business Processes Retrieval Based on Multimodal Search and Lingo Clustering Algorithm," *IEEE Latin America Transactions*, vol. 13, no. 3, pp 769–776, 2015.

[5] G. Sidorov, *Construcción no lineal de n-gramas en la lingüística computacional*, Mexico DF: Sociedad Mexicana de Inteligencia Artificial, 2013.

[6] G. Sidorov, F. Velasquez, E. Stamatatos, A. Gelbukh, and L. Chanona-Hernández, "Syntactic N-grams as machine learning features for natural language processing," *Expert Systems with Applications*, vol. 41, no. 3, pp. 853–860, 2014.

[7] H. Ordoñez, J. C. Corrales, C. Cobos, L. K. Wives, and L. Thom, "Collaborative Evaluation to Build Closed Repositories on Business Process Models," *ICEIS 2014, Proceedings of the 16th International Conference on Enterprise Information Systems*, vol. 3, SciTePress, pp. 311–318, 2014.

[8] A. Koschmider, T. Hornung, and A. Oberweis, "Recommendation-based editor for business process modeling," *Data Knowl. Eng.*, vol. 70, no. 6, pp. 483–503, 2011.

[9] D. A. Rosso-Pelayo, R. A. Trejo-Ramirez, M. Gonzalez-Mendoza, and N. Hernandez-Gress, "Business Process Mining and Rules Detection for Unstructured Information," *MICAI 2010*, *Ninth Mex. Int. Conf. Artif. Intell.*, IEEE, pp. 81–85, 2010.

[10] C. J. Turner, A. Tiwari, and J. Mehnen, "A genetic programming approach to business process mining," *Proc. 10th Annu. Conf. Genet. Evol. Comput. GECCO 2008*, p. 1307, 2008.

[11] C. Diamantini, D. Potena, and E. Storti, "Clustering of Process Schemas by Graph Mining Techniques" (extended abstract), *Methodology*, vol. 4, p. 7, 2011.

[12] J. Melcher, D. Seese, and I. Aifb, "Visualization and Clustering of Business Process Collections Based on Process Metric Values," *Measurement*, vol. 8, p. 9, 2008.

[13] W. Sheng, X. Liu, and M. Fairhurst, "A Niching Memetic Algorithm for Simultaneous Clustering and Feature Selection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 7, pp. 868–879, 2008.

[14] D. R. Ferreira, "Applied Sequence Clustering Techniques for Process Mining," *Handbook of Research on Business Process Modeling*, IGI Global, pp. 481–502, 2009.

[15] D. Ferreira, M. Zacarias, M. Malheiros, and P. Ferreira, "Approaching Process Mining with Sequence Clustering: Experiments and Findings," *Engineering*, vol. 7, no. 1, pp. 1–15, 2008.

[16] A. Ordonez, H. Ordonez, C. Figueroa, C. Cobos, and J. C. Corrales, "Dynamic reconfiguration of composite convergent services supported by multimodal search," *Lecture Notes in Business Information Processing*, 2015, vol. 208, pp. 127–139.

[17] C. Figueroa, H. Ordoñez, J.-C. Corrales, C. Cobos, L. K. Wives, and E. Herrera-Viedma, "Improving Business Process Retrieval Using Categorization and Multimodal Search," *Knowledge-Based Syst.*, vol. 110, pp. 1–17, 2016.

[18] Christopher D. Manning, Raghavan, Prabhakar, Schütze, *Introduction to Information Retrieval,* Cambridge University Press, 2008.

[19] Y.-C. Hu, B.-H. Su, and C.-C. Tsou, "Fast VQ codebook search algorithm for grayscale image coding," *Image Vis. Comput.*, vol. 26, no. 5, pp. 657–666, May 2008.

[20] T. Handhayani and L. Hiryanto, "Intelligent Kernel K-Means for Clustering Gene Expression," *Procedia Comput. Sci.*, vol. 59, pp. 171–177, 2015.

[21] H. Ordoñez, J. C. Corrales, and C. Cobos, "Business Processes Retrieval based on Multimodal Search and Lingo Clustering Algorithm," *IEEE Lat. Am. Trans.*, vol. 13, no. 9, pp. 40–48, 2015.

[22] L. L. Jae-Yoon Jung, Joonsoo Bae, "Hierarchical clustering of business process models," *Eng. Inf. Syst. Control*, vol. 5, no. 12, pp. 613–616, 2009.

[23] Q. Zhao and P. Fränti, "WB-index: A sum-of-squares based index for cluster validity," *Data Knowl. Eng.*, vol. 92, pp. 77–89, 2014.

[24] H. Ordonez, J. C. Corrales, C. Cobos, and L. K. Wives, "Collaborative grouping of business process models," pp. 1–2, 2014.

ISSN 2395-8618

Cross-Language Information Retrieval with Incorrect Query Translations

Rajendra Prasath and Sudeshna Sarkar

***Abstract*—In this paper, we present a Cross Language Information Retrieval (CLIR) approach using corpus driven query suggestion. We have used corpus statistics to gather a clue on selecting the right query terms when the translation of a specific query is missing or incorrect. The derived set of queries are ranked to select the top ranked queries. These top ranked queries are further used to perform query formulation. Using the re-formulated weighted query, we perform cross language information retrieval. The results are compared with the results of CLIR system with Google translation of user queries and CLIR with the proposed query suggestion approach. We have English and Tamil corpus of FIRE 2012 dataset and analyzed the effects of the proposed approach. The experimental results show that the proposed approach performs well with the incorrect translation of the queries.**

***Index Terms*—Cross-language information retrieval, incorrect query translations, corpus-driven query suggestion, query representation, retrieval performance.**

I. Introduction

ALL information may not be available in all languages. Suppose a user may query for some information in one language. The information may not be present in that language but may be available in another language that could fulfil their information needs. To support users to access information present in a different language, we require information retrieval systems for different language pairs. Such system are called *Cross Language Information Retrieval* (CLIR) systems. The language of the user query is referred to as *Source Language* (SL) and the language in which information is sought is the *Target Language* (TL).

In the simplest implementation of a CLIR system, a query given in the source language needs to be translated in the target language. For this, one may use a SL-TL bilingual dictionary or any other available SL-TL machine translation system. In Natural Language Processing(NLP), the same concepts may be expressed by different terms or phrases. This is called *Synonymy*. Also a term or a phrase can have multiple meanings. This is referred to as *polysemy*. These variations create problems for monolingual searches, but the effects are more in cross language retrieval. The translated query may not be able to retrieve document in the target language because the concepts may be expressed in the target language using different terms. Secondly the bilingual SL-TL dictionary may be incomplete and the query terms may not have right mapping to a query term in the target language. Even if a dictionary is large, it may be not have coverage for technical terms, named entities and so on. Such terms occur very commonly in a user query. Thirdly the SL-TL translation system may be inaccurate and the terms in the source language may be translated to wrong terms in the target language.

There may occur several issues in the translation process of the query from SL to TL. The translation process may result in the following issues:

1) some query terms may not be translated because they are absent in the dictionary.
2) some query terms may be wrongly translated.

In some cases, even when a query term is translated from the source to the target language, the translated term may not be appropriate.

We have listed three queries in Table 1. In this table, the first column shows the original query in Tamil language, the second column shows the actual query intent of user information needs, and the third column shows the dictionary based translation of these three query in the English.

Let us look at the queries listed in Table 1. While using the dictionary based query translation, the query terms underlined in the first column are not translated from source language to the target language. In the first query, the query terms "vengai" has a correct translation in the dictionary: *leopard*. But this query term has another correction translation: *vengai tree* (a kind of tree known for its strength) which is not found in the dictionary. In the context of the query, the term, *vengai tree* is the right translation. In the second query, the query terms *doosu* (*dust* in English) and *padindha* (*ingrained* in Enlgish) are not found in the dictionary. In the third query, the query term *velli* has three different correct translations: *day in a week* or *moon* or *silver metal*. Out of these three senses, the query term *moon* is the correct translation and its sense is appropriate to the actual context of the query.

There may also exist a case in which we may not be able to find the translation of a compound term in the dictionary. For example, the Google translation tool may not be able to translate a term: *marachchattam* in the second query. This term is a compound word composed of the terms: *maram* (*tree* in English) and *chattam* (this term has two translations in English: *law* and *reaper* or *frame*). In this case, *wooden*

Manuscript received on February 20, 2016, accepted for publication on June 16, 2016, published on October 30, 2016.

Rajendra Prasath is with the Department of Computer and Information Science, Norwegian University of Science and Technology, 7491 Trondheim, Norway (email: drrprasath@gmail.com; see http://www.mike.org.in/rajendra).

Sudeshna Sarkar is with the Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur, West Bengal, 721 302, India (e-mail: shudeshna@gmail.com).

ISSN 2395-8618

frame or *wooden reaper* is the correct translation that fits with the context of the actual query.

Sometimes, the correct query translation can be obtained by referring to the entire corpus. Since we can only retrieve documents present in the corpus, we may look for a set of query terms which co-occur together in the given corpus.

In this work, we want to use corpus based evidence for translating the query so that the query is appropriate with respect to the corpus. It is not about whether the translation is good or bad, but we are concerned about whether the query retrieves the documents or not. Secondly, the retrieved documents should satisfy the information needs as expressed in the original query. Here we propose a general methodology that makes use of the corpus in order to find the translation of the query terms used for document retrieval task.

In this paper, we have worked on cross language information retrieval with Tamil-English datasets of FIRE corpus[1] and run experiments for *adhoc* news document retrieval. In this work, the query is given in Tamil language which is the source language and documents retrieved are in English which is our target language.

The paper is organized as follows: The next section presents a comprehensive the review of literature related to various strategies in cross lingual information retrieval. Section III presents motivations and objective of this research work. Then we describe the underlying cross lingual information retrieval problem and the issues associated with CLIR systems in Section IV. Then in Section V, we describe our proposed CLIR approach in the context of Indian language pairs. Then we present our experimental results in Section VI. Finally Section VII concludes the paper.

II. Existing Work

We have presented some work related to the query translation issues in cross language information retrieval. Here we describe some dictionary based approaches related to our proposed approach.

A bi-lingual machine readable dictionary or thesaurus based query translation is well studied for different language pairs in cross language information Retrieval [1], [2], [3], [4].

Xi and Cho [5] proposed a method to automatically construct a dictionary based on co-occurrence from English-Chinese parallel corpus for query translation. They used different approaches to calculate the candidate translation equivalent pair correlation degree.

Hull and Grefenstett [6] developed a multilingual IR systems at Xerox which translated French queries and English documents. Their approach works as follows: after morphological analysis, each term is replaced with its inflectional root and the system forms a translated query by looking at each root in the bilingual transfer dictionary. Missing terms are kept unchanged in the translated query. This translated query is then used to perform monolingual document retrieval.

Ballesteros and Croft [2] proposed phrasal translation approach to handle multi-term phrases in cross language information retrieval. In this work, authors focused on the local context analysis to find words and phrases related to each query. They compared this approach with local feedback approach to address the errors associated with the dictionary based translation of words and phrases.

Gelbukh [7] presented a thesaurus-based information retrieval system that enriches the query with the whole set of the equivalent forms. Their approach considers enriching the query only with the selected forms that really appear in the document base and thereby providing a greater flexibility.

Oard and Ertunc [8] proposed a translation based indexing approach in which translation and indexing processes area integrated to improve query time efficiency. This approach uses machine readable bilingual dictionary in which the document's language is the source language and query language is the target language. The idea is to add every possible translation of each document - language term in the index.

Garain *et* al. [9] described an approach to deal the transliteration of out-of-vocabulary (OOV) terms in English into Bengali to improve English-Bengali cross language information retrieval. They used a statistical translation model as a basis for transliteration, and present evaluation results on the FIRE 2011 datasets. Authors used Indri system with #syn operator to handle OOV terms.

Recently, Ali Hosseinzadeh *et* al. [10] presented a set of experiments in which the impact of applying Google and Bing translation systems for query translation across multiple language pairs has been compared for two very different cross language information retrieval tasks.

III. Objectives

Since the document retrieval process depends on the translation of the user query, getting the correct translation of the user query is of great interest. There could be many issues in getting the right translation. Terms present in the dictionary may have multiple meaning and it is essential to identify and choose the right meaning appropriate for the user information needs. Alternatively, query terms in the source language may or may not be present in the dictionary (for example, name of a person or a place). So the actual query terms in source language has to be mapped appropriately to its related query terms in the target language. So in the presence of resources like incomplete dictionary, inaccurate machine translation system, and insufficient tools, we have to identify the appropriate translation for the original user query. Also there might exist multiple translations for a given query. The right translation pertaining to user information needs has to be identified from multiple translation outputs. The underlying corpus evidence may suggest a clue on selecting a suitable query that could eventually perform better document retrieval.

[1]FIRE corpus is available at: http://www.isical.ac.in/~fire

ISSN 2395-8618

Query in Tamil (Transliterated Query in English)	Actual Meaning of the Query	Dictionary Based Translated Query in English
வேங்கை மரங்கள் கடத்தல் (vengai marangal kadaththal)	Smuggling of special type of trees called "Vengai"	leopard tree trees smuggling passing
தூசு படிந்த மரச்சட்டம் (doosu padindha marachchattam)	A wooden frame with dust ingrained on its surface	a wooden frame
வெள்ளி முளைக்கும் நேரம் (velli mulaikkum neram)	The rising time of the Moon	venus star silver the planet time

Fig. 1. List of query terms in Tamil and English with their meaning in the correct query context

In order to do this, we want to use the corpus in order to find the most appropriate query translation that could be used for better document retrieval.

IV. Cross Language Information Retrieval

In this section, we describe the basic working principles of a cross language information retrieval system. Users search for some information in a language of their choice and we call this language the *source* language. The user looks for information present either in the query language or in a different language which we call the *target* language. Some cross language IR systems first perform the translation of the user query given in the source language to the target language. Then using the translated query, the CLIR system performs document retrieval in that target language and translates the retrieved documents in the source language so that the users can get the relevant information in a language that is different from their own.

In CLIR systems, translation and ranking are two major tasks.

A. Translation Task

In CLIR systems, either a query or a document has to be translated from SL to TL. We describe below both these methods:

a) Query Translation: Since a query is very short and contains a few terms, it is convenient to translate it from SL to TL and this task is much easier than translating the whole document. Then the translated query is used for monolingual retrieval in the target language.

b) Document Translation: Often query translation suffers from certain ambiguities in the translation process, and this problem is amplified when queries are short and under-specified. In these queries, the actual context of the user is hard to capture and this results in translation ambiguity. From this perspective, document translation appears to be more capable of producing more precise translation due to richer contexts.

In this work, query translation is much simpler compared with the document translation. So we used query translation to map the user query from source language to the target language and then performed monolingual document retrieval in the target language.

B. Document Ranking

Once documents are retrieved and translated back into the source language, a ranked list has to be presented based on their relevance to the actual user query in the source language. So a good ranking methodology is important in cross language information retrieval.

V. The Proposed CLIR system

We present an approach to improve the cross lingual document retrieval using corpus driven query suggestion (CLIR-CQS) approach. We have approached this problem of improving query translation process in the cross language information retrieval by accumulating the corpus evidence and using such evidences to re-formulate the user query for better information retrieval. Here we assumed that a pair of languages: (SL, TL) is chosen and a *dictionary* D is given for this pair of languages.

A. Identifying Missing / Incorrect Translations

Any query translation system (using either a dictionary based or machine translation based approach) translates the user query given in the source language SL into the target language TL. For every query term, we may either get one more terms correct meaning from the dictionary. Such terms are referred to as *synonyms*. But there are terms that have multiple meanings. Such terms are referred to as *polysemy* terms. These polysemy terms having multiple meaning may result in multiple terms during the translation. Since the dictionary may have limited number of entries, we may have missing or incorrect translation of the user query in language TL. To compensate for the missing translation of query terms, we could use co-occurrence statistics from the entire corpus. But this would take a substantial amount of query processing time in an online system. So we use an initial set of document

ISSN 2395-8618

for this purpose. We explore additional terms for partially translated query using co-occurrence of terms in this initial set of retrieved documents. We present an approach that handles the missing or incorrect translation of the user query and improves the retrieval of information in the target language TL.

Let us look at a case in which we have a partially correct translation of the original query. In this case, some query terms are translated into the target language and some are not. In case of missing translations, we use the co-occurrence statistics of query terms in language SL and their translated terms in language TL to identify the probable terms for missing translations of query terms that could result in better retrieval of cross lingual information retrieval.

B. Corpus Driven Query Suggestion Approach

In this section, we describe the Corpus driven Query Suggestion(CQS) approach for the missing or incorrect translations.

Let q_{TL} be the translation (may be a correct or partially correct or incorrect translation) of q_{SL}. We consider the case in which some query terms are translated into the target language and some are not. In case of missing translations, we use the co-occurrence statistics of query terms in the initial set of retrieved documents in language SL and their translated terms in language TL to identify the probable terms for missing translations of query terms that could result in better retrieval of cross lingual information retrieval. We say that two terms co-occur if any only if they appear in the same text segment. In our experiments, we used paragraphs as the unit of text segment.

a) Initial Retrieval: At first, the user query in language SL is given to the search engine which performs monolingual document retrieval in the source language SL and retrieve top n documents in that language: $C_{SL,Q}$. From this initial set of documents, segment the text into paragraph units. From these text segments, extract the list of terms that co-occur with any of the query terms in the text segments. Let QCO_{SL} be the list of all terms that co-occur with the original query term in the corpus: $C_{SL,Q;n}$.

b) Query Translation: Now using a bi-lingual dictionary, for each of the original query terms, find its translations in the target language TL. Here we may or may not find the translation for all query terms in the target language. We assume that we are able to find the translation for at least one query term. Then using the translated query Q', we perform monolingual document retrieval in the target language TL and retrieve top n documents: $C_{TL,Q'}$. We identify and extract text segments from these n documents in the target language TL based on paragraphs. From these text segments, we extract the list of terms that co-occur with any of the translated query terms. Let QCO_{TL} be the list of all terms that co-occur with the translated query Q' in the corpus: $C_{TL,Q';n}$.

From these two lists: QCO_{SL} and QCO_{TL}, we organize the terms in the source and target language as shown in Figure 2.

In this figure, each node corresponds to a term. More specifically large circled nodes represent the actual query terms in the source language and small circled nodes are the terms that co-occur with the query terms in the source language. Similarly, each big hexagon shaped node represents the translated query term in the target language and each small hexagon shaped node denotes the terms that co-occur with the translated query term in the target language. As shown in Figure 2, Group (A) contains actual query terms and terms that co-occur with these actual query terms, both in source language. Similarly, Group (B) contains the translated query terms and terms that co-occur with these translated query terms, both in the target language.

To understand the proposed methodology, we present an example illustrated in Figure 3 to show the mapping between the co-occurring terms so that the set of probable terms could be selected for incorrect or missing translations.

We find all terms that co-occur with the query terms in SL using the retrieved set of documents in SL. Similarly we find all terms that co-occur with the translated query terms in TL using the retrieved set of documents in TL. Since the number of co-occurring terms may be very large, we can afford to select only a few of them due to the actual query processing time in online. So we propose a method to score the co-occurring terms. Based on the score, we may select the terms which are more important. The method used for scoring is given below:

Weighting of query terms Using the initial set of documents, we compute the weights of the terms that co-occur with the query terms. We consider the initial set of top n documents retrieved for the user query in the source language SL.

Let $Q = q_1, q_2, \cdots, q_p$ be the query terms and CO_q be the set of terms that co-occur in the same text segment as term q.

We get the set of terms, say QCO_q, that co-occur with all query terms Q as follows:

$$QCO_Q = \bigcup_{i=1}^{p} CO_{q_i} \tag{1}$$

Next, we describe the proposed approach for weighting of co-occurring terms in detail. At first, for a given user query, we retrieve top n documents using any standard monolingual IR system. Then for every term in the actual query Q, we obtain the set of terms that co-occur in the same text segment. In order to get a clue on the importance of the co-occurring terms, we compute a weight for each term. In order to compute the weight, we first define the term frequency and inverse document frequency of a co-occurring term as follows:

The term frequency *tf* of each term is defined as the number of times it occurs in n text segments. The *idf* of a term, say q_i is computed as follows:

$$idf(q_i) = \log \frac{N}{df_{q_i}} \tag{2}$$

The weight of each co-occurring term $ct_i \in QCO_Q$, $(1 \leq$

ISSN 2395-8618

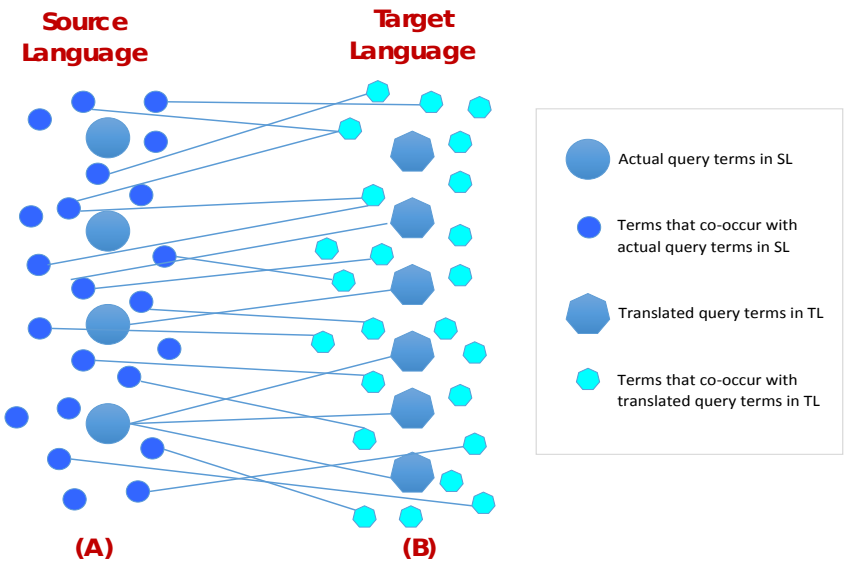

Fig. 2. A conceptual overview of the proposed query suggestion approach

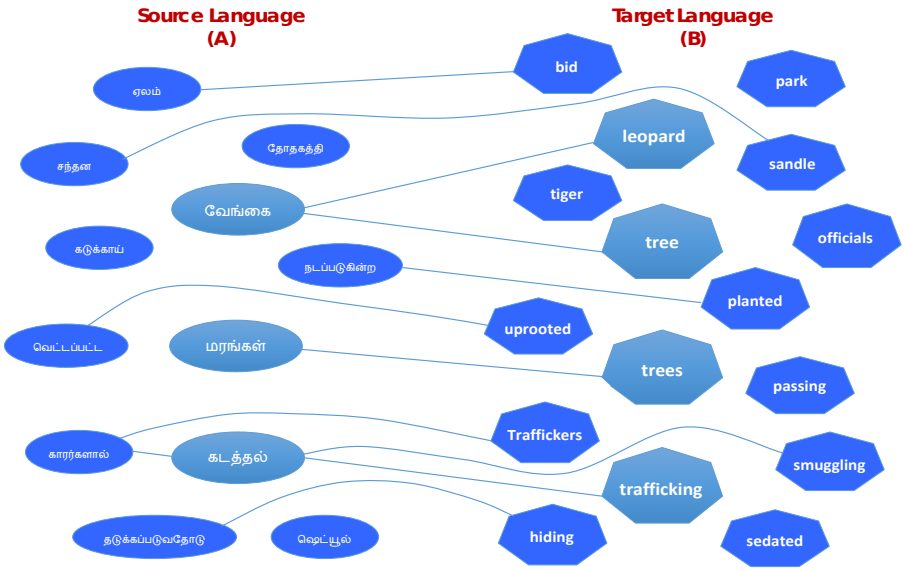

Fig. 3. Example that shows the identification of probable terms for incorrect translation

$i \leq |QCO_Q|$) is computed using the equation:

$$termWeight(ct_i) = tf(ct_i) \times idf(ct_i) \quad (3)$$

Here we explain three different approaches to estimate the weight of the term frequency (in the above equation) across n segments of the retrieved documents:

1) **Term Frequency** (*tf*): The *tf* of a term ct_i is defined as the number of time it occurs in n text segments.
2) **Logarithmic Term Frequency** (log *tf*): Logarithmic value of the term frequency of the term ct_i
3) **Average Term Frequency** (*avg tf*): The *avg tf* of a term ct_i is the ratio between the total number of occurrences of the given term in n text segments and the total number of text segments in which that term occurs.

In this formulation, we use *averageTF* as an indicator for those terms that could either be an entity or the term that tells the type of an entity. For example, named entities may score more term weight giving an indication that its equivalent translation may not exist in the dictionary. Based on the weights, we select the co-occurring terms which are more important in exploring the query terms for missing translations in the target language.

Next, we create a bipartite network by connecting the nodes in the group (A) with the nodes in group (B).

ISSN 2395-8618

c) Bipartite Network: We have two different list of terms: one in the source language SL and another one in the target language TL. Now we create a bipartite network with the terms in QCO_{SL} and QCO_{TL}. Here each term is considered as a node and we add link between a node in QCO_{SL} and a node in QCO_{TL} as follows: A term $q_{SL} \in QCO_{SL}$ is connected to a term $q_{TL} \in QCO_{TL}$ if $< q_{SL}, q_{TL} >$ is found in the dictionary D. We have illustrated an example showing the links between the terms in the SL and the terms in TL in Figure 3.

d) Term Importance: Next we perform the scoring of the co-occurring terms of correct translations in the target language TL. This scoring is used to find a list of candidate terms for missing translations in the target language. We use the bipartite network to find the importance of the terms in QCO_{TL}. For this, we estimate term score $tscore(q_j)$ for each term q_j that has a link to a term in QCO_{SL}.

This term score $tscore(q_j)$ is calculated as follows:

For each term q_j $1 \leq j \leq |QCO_{TL}|$, $tscore(q_j)$ of j^{th} term in QCO_{TL} is computed as:

$$tscore(q_j) = deg(q_j) + \alpha * termWeight(q_j) \qquad (4)$$

where α is a factor to scale the term weights in the retrieved document collection.

e) Identify Probable Query terms for Missing Translations: Based on the computed term scores, we sort the terms in QCO_{TL} and selected the terms with high term scores. Since our method may not be able to get the exact matching terms for missing translations, we add multiple terms that represent different aspects of the missing translation. Then the number of terms with high score is chosen as follows: We assume that a set of topics denoted it by $ntopics$, would be better to represent the missing terms in TL during the translation process. Let nt be the number of terms (in the original query) for which no equivalent translation exists. Now we choose ($ntopics \times nt$) terms from QCO_{TL} associated with each missing query term. This list of probable terms is a representative list for missing or incorrect translation of the query terms in the target language TL.

f) Query Formulation: Using the list of probable terms and their associated term scores, we perform a new weighted query formulation. In this query formulation, we use the term score of each probable term in the target language TL as the boost factor and form a single weighted query. We give more boosting score for the terms for which the one correct translation exists in the dictionary. Otherwise, we distribute the score equally likely to all correct translations. The reformulated weighted query is used by the searcher to perform document retrieval.

g) Document Retrieval and Ranking: Now using the new weighted and reformulated query, we perform document retrieval using BM25 as the ranking function as described in Section VI-A. In fact, we use the default parameters of BM25 ranking approach unchanged.

h) Output: Finally, we return the ranked list of top k documents from the retrieved and ranked set of documents.

We present the pseudocode of the proposed approach in Algorithm V-B0h.

Algorithm 1 CLIR Using Our Query Suggestion Approach

Input: A query having p terms: $Q = \{q_1, q_2, \cdots, q_p\}$ $p > 0$
Index: Documents indexed using Lucene

Description:

1: Get the user query in the source language SL
2: Using this query, retrieve an initial set of n documents in SL
3: Using the dictionary based approach, find the translation of the user query in the target language TL
4: Using this translated query, retrieve an initial set of n documents in the target language
5: Using the documents retrieved for the actual query, identify co-occurring terms of the actual query terms in SL. We call this list as QCO_{SL}.
6: Using the documents retrieved for the translated query, identify the co-occurring terms in the target language TL. We call this list as QCO_{TL}.
7: For each term in QCO_{SL} and QCO_{TL}, we compute a term weight.
8: Based on the term weights, we select the top scoring terms in QCO_{SL} and QCO_{TL} separately.
9: Using the selected top scoring terms, we create a bipartite network: Terms are referred to as nodes. An edge from a node x in QCO_{SL} to a node y in QCO_{TL} is added if and only if the pair $< x, y >$ exists in the dictionary
10: Now compute the term importance score for each term in the target language using the Equation 4
11: Based on the term importance score, rank the terms in QCO_{TL} and choose top d terms with their term importance score.
12: Formulate a single new weighted query in the target language using the terms and their term importance scores which are used as boost factors
13: Perform document retrieval in the target language TL using the newly formed weighted query and BM25 as the ranking function.
14: Generate the final ranked list of documents in the target language TL
15: **return** top k documents in TL as final search results

Output: The ranked list of top k documents in the target language TL.

In the next section, we present the details of our experiments with the proposed cross language document retrieval approach.

VI. Experimental Results

A. Corpus

In this experiment, we considered cross language information retrieval approach on *Tamil* and *English* languages. We

ISSN 2395-8618

have used the multi-lingual adhoc news documents collection of FIRE[2] datasets for our experiments. More specifically, we have used English and Tamil corpus of FIRE 2012 dataset and analyzed the effects of the proposed approach. FIRE 2012 is an incrementally added collection documents from FIRE 2008, FIRE 2010 and FIRE 2011 corpus. The coverage of documents in each of FIRE 2012 collection is listed in Table. I. In this collection, Tamil collection contains news documents more than English collection. English news documents consists of more news documents at the national level where as Tamil news collection covers more regional news.

TABLE I
FIRE 2012 AD HOC DATASET USED IN THIS CLIR EXPERIMENT

Language	# documents	# terms
English	392,577	1,427,986
Tamil	568,335	3,494,299

We have considered a set of 10 queries selected in *Tamil* which are listed in Table II. We have used a Tamil-English bilingual dictionary with 44,000 entries in which there are 20,778 unique entries and 21,135 terms have more than one meaning. We have used this dictionary for translating query terms and also to map the terms co-occurring with the correctly translated pairs.

We use Lucene[3] for indexing and retrieval system with Okapi Best Matching 25 (BM25) ranking used in this paper.

a) Okapi BM25: To rank the final set of the retrieved documents, Okapi BM25 [11], [12] may be used as a ranking function. BM25 retrieval function ranks a set of documents based on the query terms appearing in each document, regardless of the inter-relationship between the query terms within a document. Given a query Q and a document D, the similarity score between them is computed using BM25 ranking function as follows:

$$sim(Q,D) = \sum_{i}^{n} idf_i \cdot \frac{tf_{i,D} \cdot (k_1+1)}{tf_{i,D} + k_1 \cdot (1 - b + b \cdot \frac{|D|}{\text{avgdoclength}})} \quad (5)$$

where $tf_{i,D}$ is the term frequency of the query term i in the document D; $|D|$ is the length of the document D and *avgdoclength* is the average document length in the text collection; k_1, $k_1 \in \{1.2, 2.0\}$ and b, $b = 0.75$ are parameters; and idf_i is the inverse document frequency of the query term i.

Then we retrieve top 20 documents for each query to perform the scoring of candidate terms. We have used three different approaches in weighting the co-occurring terms:

– **Term Frequency**: We use the standard counting of the term frequency across n text segments to compute the weight of each co-occurring term ct_i in SL and TL languages:

$$termWeight(ct_i) = tf(ct_i) \times idf(ct_i) \quad (6)$$

– **Logarithmic Term Frequency**: We use logarithmic function to scale the term frequency using the following equation:

$$termWeight(ct_i) = (1 + \log(tf(ct_i))) \times idf(ct_i) \quad (7)$$

– **Average Term Frequency**: *avg tf* is used to compute the weight of the co-occurring term ct_i in both languages:

$$termWeight(ct_i) = \frac{tf(ct_i)}{m} \times idf(ct_i) \quad (8)$$

where m denotes the total number of segments in which the term ct_i occurs.

We have found that *Average Term Frequency (avgtf)* captures query terms that maps to the translations of the query terms in the target language related to the actual context of the user query. Table III shows the lists of top terms ranked by different term weighting approaches.

Table IV shows the monolingual retrieval performance with FIRE 2012 Tamil Corpus. This is to show the coverage of news documents for the selected query terms in the source language: *Tamil*.

B. Comparisons

We have considered 4 different methods to evaluate the proposed approach.

– **CLIR with Dictionary Based Approach (CLIA-DICT)**: In this experiment, we used the dictionary based approach to translate the user query in source language into the target target language and then documents are retrieved in the target language.
– **CLIR with Google Translation Tool (CLIA-GTT)**: In this experiment, we considered the machine translated query in English using Google translation tool [4] in the period between January 30 and February 9, 2015.
– **CLIR with the Proposed Approach (CLIA-CQS)**: The proposed approach is applied to perform translation of the query from SL to TL using the procedure described in section V-B. In this experiment, we have used top 20 documents are used as the initial set of relevant documents and additional terms are explored for the missing translations.
– **CLIR with the Manual Reference Translation (CLIA-REF)**: Finally we have manually translated the user query into the target language and then performed document retrieval in the target language.

We have manually evaluated the top 10 retrieved documents for each query in the 3-points scale: relevant (1.0), partially relevant (0.5) and irrelevant (0). We used the measure *precision @ top n* documents for each query and tabulated the

[2]Forum for Information Retrieval Evaluation, http://www.isical.ac.in/~fire/
[3]Lucene:www.apache.org/dist/lucene/java/
[4]https://translate.google.com

ISSN 2395-8618

TABLE II
LIST OF QUERIES USED IN OUR EXPERIMENTS

No	Queries in Tamil	Reference Translation in English	Dictionary Based Translation	Translation by Google Translation Tool	Translation by the Proposed approach
1	வேங்கை மரங்கள் கடத்தல்	vengai trees smuggling	leopard tree trees smuggling passing	Leopard trees trafficking	leopard, tree trees smuggling, passing bid scythes tress traffickers planted afforestation axing firewood harms uproot chopping trimming nailed concedes officials
2	தூசு படிந்த மரச்சட்டம்	dirt ingrained wooden frame	a wooden frame	Grime maraccattam	wooden frame clumpy skimmer clods crossovers squalor ingrained railing dislodge mound transformer trays
3	மேற்கில் ஞாயிறு மறைவு	Sun sets in west	the sun as a planet shelter a hiding place secret obscurity	The death Sunday in the West	sun planet secrecy concealment secret hiding place secret obscurity lapses pm skies lifts expiry nightmare disclose
4	சேலம் வீரபாண்டி சிறையில் கலாட்டா	outbreak in Salem Veerapandi prison	in prison comedy	Salem Veerapandi booed in prison	prison comedy hafta slashes coerce panicking sniffs amass prodding extradited accomplice culpable barracks deported
5	சசிகலா ஆதிமுக கட்சியில் இருந்து நீக்கம்	Sasikala expelled from ADMK party	from clearing passing away as clouds darkness fear sleep c, an opening	Shashikala atimuka removal from the party	from clearing passing away as clouds darkness fear sleep c, an opening whines wipeout broadens indistinguishable seeped catcher paradoxically defection disconnect deleted beg boycotting impartial
6	தமிழக மீனவர்கள் போராட்டம்	Tamilnadu fishermen struggle	fishermen diversity of opinions rivalry	Fishermen struggle	fishermen diversity opinions rivalry pirates hostages impounding blockading incarcerated despondent trawlers repatriation encroachment assaults distracted evicted
7	சம்பா பயிர்கள் தண்ணீர் இன்றி வாட்டம்	samba crops fade out without water	cold water distress withered emaciated faintness drooping plants countenance	Samba crops without water gradient	cold water distress, withered emaciated faintness drooping megaliters kuruvai optimized unfeasible eggplant agribusiness percent aquifers contaminating jowar ravaging rice hose overflowed cusecs
8	ஊட்டியில் மலர் கண்காட்சி நிறைவு விழா	closing ceremony of flower exhibition in Ooty	exhibition completion fullness abundance plenteousness completeness festival	Ooty flower show at the closing ceremony	exhibition completion fullness abundance plenteousness, plentifulness, completeness much festival presents pm exhibition paintings tasar workshop armband seamlessly sandalwood art splash
9	கோவையில் முக்கிய பிரமுகர் கைது	important person arrested in Coimbatore	Arrest	The main figure arrested in Coimbatore	arrest gangraped discharged lawful fidayeen offenders escapes conversant arrester tractor assisting disclosure
10	வெள்ளி முளைக்கும் நேரம்	Moon rising time	venus star silver the planet time	Silver germination time	venus star silver planet time weekend flights eclipse astronauts tsunami perigee spaceman amavasya bluish apogee gravitational mavens

results. Table V presents the details of our experiments done in CLIR with Dictionary based Translation (CLIR-DICT); CLIR with machine translation of user queries with Google translation tool (CLIR-GTT); CLIR with the proposed corpus based query selection approach (CLIR-CQS); and CLIR with Manual Reference Translation (CLIR-REF). We used Google translation tool (GTT) [5] to translate the user query given in Tamil language into English language.

C. Discussion

Consider the query ID 1. In this query, there are three tamil query terms: { *Vengai*, *Marangal*, *Kadaththal* }. The term *Vengai* may refer to two variations: *Vengai*, type of a tree whose botanical name is *Pterocarpus marsupium* or *leopard*, and animal; *Marangal*, trees: the correct translation; and finally *Kadaththal* may refer to at least three variations: *trafficking* or *smuggling* or *stealing*. This would give 2 x 1 x 3 = 6 different queries. We identify a set of terms that boosts these query variations and then choose top k terms to form the single weighted query using query terms weighting approach.

[5]Google Translation Tool is available at: http://translate.google.com/

During the evaluation of the proposed approach, we have used 3-points scale for making relevant judgments. We have considered top 10 documents for each query and manually evaluated the retrieved results using the metric: *precision @ top* k documents. The preliminary results show that the proposed approach is better in disambiguating the query intent when query terms that have multiple meanings are given by the users. The average access time for terms set in Tamil is 765.3 milliseconds and 97.8 milliseconds in English. Since the retrieval of initial set of documents and finding co-occurrence terms from this initial set of documents take very negligible amount of time (less than 2 seconds even for top 50 documents), we did not consider the retrieval time comparison in this work.

VII. Conclusion

We have presented a cross language document retrieval approach using corpus driven query suggestion approach. In this work, we have used corpus statistics that could provide a clue on selecting the right query terms when translation of a specific query term is missing or incorrect. Then we rank the set of the derived queries and select the top ranked queries to perform query formulation. Using the re-formulated weighted

ISSN 2395-8618

TABLE III
LIST OF HIGHLY WEIGHTED PROBABLE QUERY TERMS USING THREE DIFFERENT TERM WEIGHTING APPROACHES: tf, log tf AND $avg\ tf$

QID	Actual Queries [Reference Translation]	Highly Weighted Query Terms		
		Term Frequency *(tf)*	**Logarithmic Term Frequency** *(log tf)*	**Average Term Frequency** *(avg tf)*
1	வேங்கை மரங்கள் கடத்தல் [vengai trees smuggling]	2003 reason Twenty year numerous allowed nailed areas oversee curb trees cover	reason Twenty year numerous allowed nailed areas oversee curb trees cover properly decided suffer National fells harms	Smuggling passing bid scythes tress traffickers planted afforestation axing firewood harms uproot chopping trimming nailed concedes officials
4	சேலம் வீரபாண்டி சிறையில் கலாட்டா [outbreak in Salem Veerapandi prison]	2004 1993 hands spoke allowed discussion fighting guards decided Indians Sections framed liable Ensuring represented prison	hands spoke allowed discussion fighting guards decided Indians Sections framed liable Ensuring represented prison landed shortest Qayyum	prison comedy hafta slashes coerce panicking sniffs amass prodding extradited accomplice culpable barracks deported
7	சம்பா பயிர்கள் தண்ணீர் இன்றி வாட்டம் [samba crops fade out without water]	350 2003 318 2015 Water half overflowing year reservoirs areas suffer investing grow dealing Crops impact investment require	Water half overflowing year reservoirs areas suffer investing grow dealing Crops impact investment require easy plant contaminated drums	cold water distress, withered emaciated faintness drooping megaliters kuruvai optimized unfeasible eggplant agribusiness percent aquifers contaminating jowar ravaging rice hose overflowed cusecs
9	கோவையில் முக்கிய பிரமுகர் கைது [important person arrested in Coimbatore]	168 inform thought Nazir Vaiko professional absconding Manoharan farm warrant July blasts based court persons night	inform thought Nazir Vaiko Delhis professional Saturday absconding Manoharan farm warrant July thinks defined blasts based	gangraped discharged lawful fidayeen offenders escapes conversant arrester tractor assisting disclosure

query, cross language information retrieval is performed. We have presented the comparison results of CLIR with Google translation of the user queries and CLIR with the proposed corpus based query suggestion. The preliminary results show that the proposed approach seems to be promising and we are exploring this further with graph based approach that could unfold the hidden relationships between query terms in a given pair of languages.

ACKNOWLEDGMENT

Authors gratefully acknowledge the support extended by Dr. Philip O'Reilly of University College Cork, Cork, Ireland during the last stage of this work.

REFERENCES

[1] G. Salton, "Experiments in multi-lingual information retrieval," Department of Computer Science, Cornell University, Ithaca, NY, USA, Tech. Rep., 1972.

[2] L. Ballesteros and W. B. Croft, "Phrasal translation and query expansion techniques for cross-language information retrieval," in *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR 1997. New York, NY, USA: ACM, 1997, pp. 84–91. [Online]. Available: http://doi.acm.org/10.1145/258525.258540

[3] J. Capstick, A. K. Diagne, G. Erbach, H. Uszkoreit, A. Leisenberg, and M. Leisenberg, "A system for supporting cross-lingual information retrieval," *Inf. Process. Manage.*, vol. 36, no. 2, pp. 275–289, Jan 2000. [Online]. Available: http://dx.doi.org/10.1016/S0306-4573(99)00058-8

[4] D. Zhou, M. Truran, T. Brailsford, V. Wade, and H. Ashman, "Translation techniques in cross-language information retrieval," *ACM Comput. Surv.*, vol. 45, no. 1, pp. 1:1–1:44, Dec 2012. [Online]. Available: http://doi.acm.org/10.1145/2379776.2379777

[5] S.-M. Xi and Y.-I. Cho, "Study of query translation dictionary automatic construction in cross-language information retrieval," in *Intelligent Autonomous Systems 12*, ser. Advances in Intelligent Systems and Computing, S. Lee, H. Cho, K.-J. Yoon, and J. Lee, Eds. Springer Berlin Heidelberg, 2013, vol. 194, pp. 585–592. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-33932-5_54

[6] D. A. Hull and G. Grefenstette, "Querying across languages: A dictionary-based approach to multilingual information retrieval," in *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR 1996. New York, NY, USA: ACM, 1996, pp. 49–57. [Online]. Available: http://doi.acm.org/10.1145/243199.243212

[7] A. F. Gelbukh, "Lazy query enrichment: A method for indexing large specialized document bases with morphology and concept hierarchy," in *Proceedings of the 11th International Conference on Database and Expert Systems Applications*, ser. DEXA 2000. London, UK, UK: Springer-Verlag, 2000, pp. 526–535. [Online]. Available: http://dl.acm.org/citation.cfm?id=648313.755685

[8] D. W. Oard and F. Ertunc, "Translation-based indexing for cross-language retrieval," in *Advances in Information Retrieval, 24th BCS-IRSG European Colloquium on IR Research Glasgow, UK,*

ISSN 2395-8618

TABLE IV
SELECTED QUERIES IN TAMIL, THE DICTIONARY TRANSLATIONS IN ENGLISH AND THE RETRIEVAL EFFICIENCY IN TAMIL MONOLINGUAL RETRIEVAL

QID	Query in TAmil	Translated Query in ENglish Google Translate / (Derieved Query terms)	User Info Need	p@5	p@10
1	வேங்கை கடத்தல்	Wang conduction / (வேங்கை tree[273] smuggling[110] cut[88] sandle wood[71] tiger[70] வனத்துறையினர்[62] near[50] people[50], steps[45] area[44])	Info about smuggling of Venghai (tree)	0.8	0.65
2	தூசு படிந்த மரச்சட்டம்	Dust-stained maraccattam (dust[128] stained[115] wood[95] coated[75] glass[72] frame[61] time[58] police[52] road[50] people[49] நடவடிக்கை[38])	Info about the dust stained wooden frame	0.7	0.6
3	மேற்கில் ஞாயிறு மறைவு	Sunday on the west side (west[210] india[111] power[106] bengal[105] side[107] sets[101] indies[95] மறைவு[51] ஞாயிறு[48] இரங்கல்[31])	Sun sets on the west	0.6	0.55
4	சேலம் வீரபாண்டி சிறையில் கலாட்டா	Create virapanti Salem in jail (jail[802] வீரபாண்டி[499] ஆறுமுகம்[287] former[149] திமுக[144] court[102] central[98] police[79] authorities[74] prison[70])	Issues made by Salem Veerapandi in prison	0.7	0.5
5	சசிகலா ஆதிமுக கட்சியில் இருந்து நீக்கம்	Athimuka Shashikala from the disposal (சசிகலா[230] அதிமுக[211] party[192] court[166] ஜெயலலிதா[128] disposal[127] chief[118] state[83] minister[82] cases[81])	News about the Sasikala's suspension in ADMK party	0.65	0.6

* Calcutta and Telegraph are the most frequent terms occur in most of the documents. So these terms are not included in our derived query terms

TABLE V
COMPARISON OF RETRIEVAL EFFICIENCY OF TOP 10 SEARCH RESULTS: CLIR-DICT, CLIR-CQS, CLIR-REF AND CLIR-GTT APPROACHES

QID	Precision @ top 5				Precision @ top 10			
	CLIR-DICT	CLIR-GTT	CLIR-CQS	CLIR-REF	CLIR-DICT	CLIR-GTT	CLIR-CQS	CLIR-REF
1	0.05	0.10	0.15	0.20	0.05	0.15	0.25	0.30
2	0.20	0.15	0.40	0.50	0.20	0.25	0.35	0.45
3	0.15	0.10	0.20	0.25	0.15	0.10	0.20	0.25
4	0.15	0.20	0.25	0.30	0.15	0.10	0.20	0.25
5	0.20	0.10	0.35	0.50	0.20	0.20	0.40	0.45
6	0.10	0.05	0.35	0.20	0.10	0.10	0.10	0.15
7	0.05	0.10	0.20	0.30	0.05	0.05	0.20	0.25
8	0.10	0.15	0.10	0.20	0.10	0.10	0.20	0.25
9	0.05	0.10	0.25	0.30	0.05	0.15	0.20	0.25
10	0.10	0.15	0.30	0.40	0.10	0.20	0.20	0.35
Avg	**0.11**	**0.11**	**0.24**	**0.295**	**0.11**	**0.125**	**0.205**	**0.265**

March 25-27, 2002 Proceedings, ser. Lecture Notes in Computer Science, F. Crestani, M. Girolami, and C. J. van Rijsbergen, Eds., vol. 2291. Springer, 2002, pp. 324–333. [Online]. Available: http://dx.doi.org/10.1007/3-540-45886-7_21

[9] U. Garain, A. Das, D. S. Doermann, and D. W. Oard, "Leveraging statistical transliteration for dictionary-based English-Bengali CLIR of OCR'd text," in *COLING 2012, 24th International Conference on Computational Linguistics, Proceedings of the Conference: Posters, 8–15 December 2012, Mumbai, India*, 2012, pp. 339–348. [Online]. Available: http://aclweb.org/anthology/C/C12/C12-2034.pdf

[10] A. Hosseinzadeh Vahid, P. Arora, Q. Liu, and G. J. Jones, "A comparative study of online translation services for cross language information retrieval," in *Proceedings of the 24th International Conference on World Wide Web Companion*, ser. WWW 2015 Companion. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2015, pp. 859–864. [Online]. Available: http://dx.doi.org/10.1145/2740908.2743008

[11] S. E. Robertson and S. Walker, "Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval," in *Proc. of the 17th ACM SIGIR conference on Research and development in IR*, ser. SIGIR 1994. New York, NY, USA: Springer-Verlag New York, Inc., 1994, pp. 232–241. [Online]. Available: http://dl.acm.org/citation.cfm?id=188490.188561

[12] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," *Found. Trends Inf. Retr.*, vol. 3, no. 4, pp. 333–389, Apr. 2009. [Online]. Available: http://dx.doi.org/10.1561/1500000019

ISSN 2395-8618

Unsupervised Word Sense Disambiguation Using Alpha-Beta Associative Memories

Sulema Torres-Ramos, Israel Román-Godínez, and E. Gerardo Mendizabal-Ruiz

***Abstract*—We present an alternative method to the use of overlapping as a distance measure in simple Lesk algorithm. This paper presents an algorithm that uses Alpha-Beta associative memory type Max and Min to measure a given ambiguous word's meaning in relation to its context, assigning to the word the meaning that is most related. The principal advantage of using this algorithm is the ability to deal with inflectional and derivational forms of words, enabling the possibility of bypassing the stemming procedure of words involved in the disambiguation process. Different experiments were performed, with two parameters as variables: the context window size, and whether stemming was applied or not. The experimental results (F1-score) show that our algorithm performs better than the use of the overlapped metric in the simple Lesk algorithm. Moreover, the experiments show that as more information is added to the sense or meaning, and the overlap metric is used, the precision of the simple Lesk algorithm is decreased-in contrast to the performance of our algorithm.**

***Index Terms*—Word sense disambiguation, simple Lesk algorithm, Alpha-Beta associative memories.**

I. Introduction

Natural Language Processing (NLP) is a multidisciplinary area of research, in which the main objective is to develop theories, algorithms, and technologies that enable and strengthen communication between computers and humans using languages that have naturally evolved in human societies (e.g., English, Spanish, French, among others.) instead of the constructed, formal languages that have been employed to program computers. Examples of NLP applications include knowledge management and discovery, information retrieval, question answering, and machine translation [1].

One of the biggest obstacles to human-computer interaction is the prevalence of homonyms in many natural languages (i.e. words that are said or spelled the same way but have different meanings). For example, the word "bass" can refer to a musical instrument, or a freshwater fish. In general, humans are very good at figuring out the meaning of ambiguous words; however, the automatic disambiguation of words remains a difficult task for computers.

Manuscript received on April 24, 2016, accepted for publication on July 9, 2016, published on October 30, 2016.

The authors are with the Department of Computer Science, CUCEI – Universidad de Guadalajara, Guadalajara, Mexico. Corresponding author: Sulema Torres-Ramos (e-mail: sulema.torres@cucei.udg.mx).

Word sense disambiguation (WSD) is one of the central topics of NLP [2]. WSD consists of automatically finding the correct meaning of an ambiguous word in a text, simply by analyzing the context in which it exists. Current WSD methods can be classified into four categories [3]: supervised, unsupervised, semi-supervised, and knowledge-based.

Supervised methods are characterized by the employment of machine-learning techniques, for the purpose of creating classification models based on a training set of hand-labeled corpus that indicates the correct meaning of each ambiguous word in a text. Unsupervised methods do not rely on training; instead, they attempt to provide sense (i.e. meaning) labels by generating clusters of word occurrences. Semi-supervised methods start with a small hand-labeled training set, and progressively improve the classification model, as it is used. Knowledge-based methods make use of knowledge sources such as collocations, thesauri, and dictionaries to assign a sense to an ambiguous word, first by comparing each of its possible definitions with those of other words in the context, and then computing a semantic similarity metric of the definitions.

Knowledge-based methods have recently been proven to outperform supervised approaches in the presence of enough knowledge, or within a knowledge-based domain, while providing at the same time much wider coverage [4].

One of the main challenges of using a dictionary for knowledge-based WSD methods is that the words in the dictionary may be in different forms (e.g., verb, plural, root, etc.), making it difficult to determine the degree of overlap between a word and its respective meanings in the dictionary. To overcome this problem, many of the knowledge-based methods incorporate a stemming step in their algorithms, which consist of reducing inflected (or sometimes derived) words to their word stem, base, or root form.

On the other hand, an associative memory is a computational tool that consists of structures that relate one or more input patterns with an output pattern [5]. One of the foremost properties and fundamental purposes of associative memories is their ability to recall output patterns, despite possible alterations or noise present in input patterns [6]. Associative memories eliminate the exhaustive search operations common in indexed memory, and therefore are very attractive in applications such as data mining and the implementation of sets, where the computations can benefit from the application's specific functioning [7].

ISSN 2395-8618

Associative memory has been an active topic of research for more than 50 years, and is still investigated both in neuroscience and in artificial neural networks [8]. In particular, Alpha-Beta associative memories have been proven to be a powerful tool for pattern recognition tasks when used in various scientific and technologic applications, such as the classification of patterns in bioinformatics databases [1], prediction of contaminant levels [10], image encryption [11], and translation of Spanish to English [5].

In this paper, we present a method that employs Alpha-Beta associative memory types Max and Min to determine how related each definition of a word is to its context, and then choose the correct definition or sense. Our method was tested using the dataset for the SENSEVAL-2 "All-words" task, with WordNet as the lexical resource. Six different experiments were made, four of them not using a back-off strategy and the remaining two, using it. A back-off strategy is an alternative method that takes a decision when the principal method cannot; the most common strategies used in WSD are: random sense, and most frequent sense. For our purposes we use random sense, because is considered an unsupervised method.

Moreover, to measure the performance of the six different experiments, three statistical metrics were used: precision, recall, and F1-score. All of them were used when our method does not implement a back-off strategy, conversely, when it was used, we only report the F1-score. The latter given that, when a method always take a decision (i.e. the coverage is one hundred percent), the precision, recall, and F1-score are the same.

The rest of the paper is organized as follows: Section II presents the background and related work of simplified Lesk and Alpha-Beta associative memories. Section III presents our proposed method to replace overlapped metric. Section IV describes the experimental resources and results, and in Section V, conclusions derived from the experimental analysis are presented.

II. Background and Related Work

A. Simplified Lesk Algorithm

One of the most popular knowledge-based methods for WSD is the Lesk algorithm [12], which is based on the assumption that words occurring in a given section of text will tend to share a common topic. This method consists of obtaining definitions in a dictionary for each word in a given text, and computes the relatedness between all those definitions. The definitions with the greatest relatedness are chosen as the correct senses of the words.

Since the Lesk algorithm may be computationally expensive, a simple Lesk algorithm was proposed [13]. In this method, the meaning of a word is determined by locating the sense that overlaps the most between the definition of the word in a dictionary, and neighboring words (context) of the ambiguous word. In this approach, each word is processed individually and independently of the meaning of other words occurring in the same context.

B. Alpha-Beta Associative Memories

An associative memory is conceived as a system that associates an input pattern (**x**) with an output pattern (**y**), through a series of steps known as the learning phase building matrix (**M**); on the contrary, to retrieve the input's corresponding output pattern, we present the input pattern to the matrix according to the recall phase. The *k*-th associations are stored in the matrix (**M**) and its *ij*-th component is denoted by m_{ij}.

The associative memory **M** is built from a finite set of pre-associated patterns, known as the fundamental set, and is expressed as follows:

$$\{ (x^{\mu}, y^{\mu}) \mid \mu = 1,2, \dots, p \} \quad (1)$$

p being the cardinality of the fundamental set. Each pattern in the fundamental set is called a fundamental pattern.

There are two categories for an associative memory: if it holds for all fundamental patterns that the input and output patterns to be associated are equals, then the memory **M** is auto-associative, i.e. $x^{\mu} = y^{\mu} \ \forall \mu \in \{1, 2, ..., p\}$. Otherwise, if there exists one association where the input pattern is different from the output pattern, then the memory **M** is called hetero- associative i.e. $\exists \mu \in \{1, 2, ..., p\}$, for which $x^{\mu} \neq y^{\mu}$.

One of the most important characteristics of an associative memory is its ability to deal with a distortion or altered version of the input vectors. It is expected that, if an altered fundamental input vector ($\tilde{x}^k$) is presented to the associative memory, then the fundamental output pattern y^k is recalled. When this happens, we say that the recall is correct.

According to [14], the Alpha-Beta model presents two binary operators designed specifically for these memories. First, we defined the sets A = {0, 1} and B = {0, 1, 2}, and operators α and β are defined in table 1. The sets A and B, the α and β operators (see Table I), along with the usual ∧ (minimum) and ∨ (maximum) operators, form the algebraic system (A, B, α, β, ∧, ∨) which is the mathematical basis for the Alpha-Beta associative memories. This system presents two types of memories: Alpha-Beta associative memory types *Max* and *Min*; its name, functionality, and capacity to deal with altered patterns depend on the use of minimum or maximum operators in both learning and recalling phases.

The building of both Max and Min associative memories is denoted by the operator ⊠, which is defined in Equation 2:

$$[y^{\mu} \boxtimes (x^{\mu})^t]_{ij} = \alpha\left(y_i^{\mu}, x_i^{\mu}\right); \quad \mu \in \{1,2, \dots, p\}, i \in \{1,2, \dots, m\}, \ j \in \{1,2, \dots, n\} \quad (2)$$

C. Alpha-Beta Heteroassociative Memories with correct recall

Alpha-Beta heteroassociative memories, unlike the original [14] model and others [15], guarantee the correct recall of the fundamental set [16]. In the following sections,

ISSN 2395-8618

we present the Alpha-Beta heteroassociative memory types Max and Min, with which the complete recall of the fundamental set is guaranteed [16].

TABLE I
DEFINITIONS OF THE ALPHA AND BETA OPERATORS

$\alpha : A \times A \to B$		
x	y	$\alpha(x,y)$
0	0	1
0	1	0
1	0	2
1	1	1

$\beta : B \times A \to A$		
x	y	$\beta(x,y)$
0	0	0
0	1	0
1	0	0
1	1	1
2	0	1
2	1	1

Let A = {0,1}, n, $p \in Z^+$, $\mu \in \{1, 2, ..., p\}$, $i \in \{1, 2, ..., p\}$ and $j \in \{1, 2, ..., n\}$, and let $\mathbf{x} \in A^n$ and $\mathbf{y} \in A^p$ be input and output vectors, respectively. The corresponding fundamental set is denoted by $\{(\mathbf{x}^\mu, \mathbf{y}^\mu) \mid \mu = 1, 2, ..., p\}$.

C.1. *Alpha-Beta Heteroassociative Memories type Max*

Learning phase

The fundamental set must be built according to the following rules: first, all **y** vectors must be built according to the o*ne-hot* codification, assigning for $\mathbf{y}^\mu$ the following values: $y_k^\mu = 1$, and $y_k^\mu = 0$ $for\ j \in \{1,2,\dots,k-1,k+1,\dots,m\}$ where $k \in \{1,2,3,\dots m\}$. Second, each $\mathbf{y}^\mu$ vector must correspond to *one and only one* $\mathbf{x}^\mu$ vector, this is, both vectors must belong to only one binary tuple ($\mathbf{x}^\mu$, $\mathbf{y}^\mu$) in the fundamental set.

Step 1: For each $\mu \in \{1,2,\dots,p\}$ from the couple (x^μ, y^μ), build the matrix: $[y^\mu \boxtimes (x^\mu)^t]_{m\times n}$

Step 2: Apply the binary $\vee$ operator to the matrices obtained in step 1 to get the new Alpha-Beta heteroassociative memory. Assign Max $\boldsymbol{V}$ as follows: $\boldsymbol{V} = \bigvee_{\mu=1}^{P}[y^\mu \boxtimes (x^\mu)^t]$, with the *ij*-th component given by:

$$v_{ij} = \bigvee_{\mu=1}^{P} \alpha\left(y_i^\mu, x_j^\mu\right) \tag{3}$$

Recalling phase

Step 1: Present pattern $\boldsymbol{x}^\omega$ to $\boldsymbol{V}$**,** complete the Δ_β operation, and assign the resulting vector to a vector called $\boldsymbol{z}^\omega$: $\boldsymbol{z}^\omega = \boldsymbol{V}\Delta_\beta \boldsymbol{x}^\omega$. The *i*-th component of the resulting column vector is:

$$\boldsymbol{z}_i^\omega = \bigwedge_{j=1}^{n} \beta\left(v_{ij}, x_j^\omega\right) \tag{4}$$

Step 2: It is necessary to build a *max sum vector* **s** according to Equation 5:

$$s_i = \sum_{j=1}^{n} T_j \tag{5}$$

where $T \in B^n$ and its components are defined as

$$T_i = \begin{cases} 1 \leftrightarrow v_{ij} = 1 \\ 0 \leftrightarrow v_{ij} \neq 1 \end{cases}$$
$$\forall j \in \{1,2,\dots,n\}\ and\ the\ s_i\ with\ \boldsymbol{s} \in \boldsymbol{Z}^p$$

Therefore, the corresponding $\boldsymbol{y}^\omega$ is given as

$$y_i^\omega = \begin{cases} 1 \text{ if } s_i = \bigvee_{k\,\in\,\theta} s_k \ \wedge\ z_i^\omega = 1 \\ 0 \text{ otherwise} \end{cases} \tag{6}$$

where $\theta = \{i | z_i^\omega = 1\}$ with $\omega \in \{1,2,\dots,n\}$

C.2. *Alpha-Beta Heteroassociative Memories type Min*

Learning phase

The fundamental set must be built according to the following rules: first, all **y** vectors must be built according to the *zero-hot* codification, assigning for $\mathbf{y}^\mu$ the following values: $y_k^\mu = 0$, and $y_k^\mu = 1$ $for\ j \in \{1,2,\dots,k-1,k+1,\dots,m\}$ where $k \in \{1,2,3,\dots m\}$. Second, each $\mathbf{y}^\mu$ vector must correspond to *one and only one* $\mathbf{x}^\mu$ vector, this is, both vectors must belong to only one binary tuple ($\mathbf{x}^\mu$, $\mathbf{y}^\mu$) in the fundamental set.

Step 1: For each $\mu \in \{1,2,\dots,p\}$ from the couple (x^μ, y^μ), build the matrix: $[y^\mu \boxtimes (x^\mu)^t]_{m\times n}$

Step 2: Apply the binary $\wedge$ operator to the matrices obtained in step 1, to get the new Alpha-Beta heteroassociative memory. Assign Min $\boldsymbol{\Lambda}$ as follows: $\boldsymbol{\Lambda} = \bigwedge_{\mu=1}^{P}[y^\mu \boxtimes (x^\mu)^t]$, with the *ij*-th component given by:

$$\lambda_{ij} = \bigwedge_{\mu=1}^{P} \alpha\left(y_i^\mu, x_j^\mu\right) \tag{7}$$

Recalling phase

Step 1: Present pattern $\boldsymbol{x}^\omega$ to $\boldsymbol{\Lambda}$**,** finish the ∇_β operation, and assign the resulting vector to a vector called $\boldsymbol{z}^\omega$: $\boldsymbol{z}^\omega = \boldsymbol{\Lambda}\nabla_\beta \boldsymbol{x}^\omega$. The *i*-th component of the resulting column vector is:

$$\boldsymbol{z}_i^\omega = \bigwedge_{j=1}^{n} \beta\left(\lambda_{ij}, x_j^\omega\right) \tag{8}$$

Step 2: It is necessary to build a *min sum vector* $\boldsymbol{r}$ according to equation 9:

$$r_i = \sum_{j=1}^{n} T_j \tag{9}$$

where $T \in B^n$ and its components are defined as

$$T_i = \begin{cases} 1 \leftrightarrow \lambda_{ij} = 0 \\ 0 \leftrightarrow \lambda_{ij} \neq 0 \end{cases}$$
$$\forall j \in \{1,2,\dots,n\}\ and\ the\ r_i\ with\ \boldsymbol{r} \in \boldsymbol{Z}^p$$

Therefore, the corresponding $\boldsymbol{y}^\omega$ is given as

$$y_i^\omega = \begin{cases} 0 \text{ if } r_i = \bigwedge_{k\,\in\,\theta} r_k \ \wedge\ z_i^\omega = 0 \\ 1 \text{ otherwise} \end{cases} \tag{10}$$

where $\theta = \{i | z_i^\omega = 0\}$ with $\omega \in \{1,2,\dots,n\}$.

ISSN 2395-8618

III. PROPOSED ALGORITHM

Considering that inflectional and derivational forms of words affect the process of word sense disambiguation, we propose an algorithm that diminishes the influence of those syntactic phenomena present in the simple Lesk algorithm.

The proposed method replaces the overlap method used in the original simple Lesk algorithm (with the use of Alpha-Beta associative memory types Max and Min), providing one with the ability to deal with an altered version of the words. The following steps show the process of building an associative memory per sense (i.e. one Max and one Min). In the learning phase, the words in the definition of an ambiguous word are used as a fundamental input pattern. Once the memories are built, to assign a sense to an ambiguous word, the context words (which may be an altered version of any fundamental input pattern) are presented to each pair of memories. At the end, a voting strategy applied to the output patterns is used to assign a correct sense.

For example, take the sentence, "The man plays an instrument in a band". To disambiguate the word *play*, then:

1. The surrounding words and definitions (glosses) are separated in different sets of words, one representing the context and the remaining sets (as many sets as there are meanings for the ambiguous word) corresponding to the senses of the ambiguous word. For this example, we only use the first three senses of the ambiguous word:

   C1 = {instrument, band, man}
   S1 = {game, sport, hocky, afternoon, cards}
   S2 = {act, have, effect, specified}
   S3 = {music, instrument, band, night}

2. Due to the binary domain of associative memory operators, the words in the senses, and the context words, are mapped to their corresponding binary representation; for simplicity, we used the ASCII code.

   C1 = {
   c^1 = (0110100101101110011100110111010001110010011
   1010101101101011001010110111001110100),
   c^2 = (01100010011000010110111001100100),
   c^3 = (011011010110000101101110) }
   S1 = {
   x^1 = (01100111011000010110110101100101),
   x^2 = (0111001101110000011011110111001001110100),
   x^3 = (0110100001101111011000110110101101111001),
   x^4 = (0110000101100110011101000110010101110010011
   01110011011110110111101101110),
   x^5 = (0110001101100001011100100110010001110011) }
   S2 = {
   x^1 = (011000010110001101110100),
   x^2 = (01101000011000010111011001100101),
   x^3 = (01100101011001100110011001100101011000110111
   0100),
   x^4 = (01110011011100000110010101100011011010010110
   0110011010010110010101100100) }
   S3 = {
   x^1 = (0110110101110101011100110110100101100011),
   x^2 = (01101001011011100111001101110100011100100111
   010101101101011001010110111001110100),
   x^3 = (01100010011000010110111001100100),
   x^4 = (0110111001101001011001110110100001110100) }

3. In order to have vectors with the same dimensions, the missing components are filled with *zeros* or *ones* depending on the Alpha-Beta associative memory used, *zeros* for Max types and *ones* for Min types. In this example, we filled them with zeros.

   C1 = {
   c^1 = (0110100101101110011100110111010001110010011
   1010101101101011001010110111001110100),
   c^2 = (0110001001100001011011100110010000000000000
   0000000000000000000000000000000000000),
   c^3 = (0110110101100001011011100000000000000000000
   0000000000000000000000000000000000000) }
   S1 = {
   x^1 = (01100111011000010110110101100101000000000000
   0000000000000000000000000000),
   x^2 = (01110011011100000110111101110010011101000000
   0000000000000000000000000000),
   x^3 = (01101000011011110110001101101011011110010000
   0000000000000000000000000000),
   x^4 = (01100001011001100111010001100101011100100110
   1110011011110110111101101110),
   x^5 = (01100011011000010111001001100100011100110000
   0000000000000000000000000000) }
   S2 = {
   x^1 = (0110000101100011011101000000000000000000000
   00000000000000000000000000000),
   x^2 = (0110100001100001011101100110010100000000000
   00000000000000000000000000000),
   x^3 = (0110010101100110011001100110010101100011011
   10100000000000000000000000000),
   x^4 = (0111001101110000011001010110001101101001011
   00110011010010110010101100100) }
   S3 = {
   x^1 = (01101101011101010111001101101001011000110000
   000000000000000000000000000000000000),
   x^2 = (01101001011011100111001101110100011100100111
   010101101101011001010110111001110100),
   x^3 = (01100010011000010110111001100100000000000000
   000000000000000000000000000000000000),
   x^4 = (01101110011010010110011101101000011101000000
   000000000000000000000000000000000000) }

4. For each sense, two fundamental sets are built: one according to the associative memory type max (C.1), and one for the associative memory type Min (C.2). Each word in the sense is considered as a fundamental input pattern.

   Input vectors:
   Sense1 = { x^1, x^2, x^3, x^4, x^5}
   Sense2 = { x^1, x^2, x^3, x^4}
   Sense3 = { x^1, x^2, x^3, x^4}

ISSN 2395-8618

Output vectors for type Max:

Sense1 = { $yMax^1 = (10000)^t$, $yMax^2 = (01000)^t$, $yMax^3 = (00100)^t$, $yMax^4 = (00010)^t$, $yMax^5 = (00001)^t$ }

Sense2 = { $yMax^1 = (10000)^t$, $yMax^2 = (01000)^t$, $yMax^3 = (00100)^t$, $yMax^4 = (00010)^t$ }

Sense3 = { $yMax^1 = (10000)^t$, $yMax^2 = (01000)^t$, $yMax^3 = (00100)^t$, $yMax^4 = (00010)^t$ }

Output vectors for type Min:

Sense1 = { $yMin^1 = (01111)^t$, $yMin^2 = (10111)^t$, $yMin^3 = (11011)^t$, $yMin^4 = (11101)^t$, $yMin^5 = (11110)^t$ }

Sense2 = { $yMin^1 = (0111)^t$, $yMin^2 = (1011)^t$, $yMin^3 = (1101)^t$, $yMin^4 = (1110)^t$ }

Sense3 = { $yMin^1 = (01111)^t$, $yMin^2 = (10111)^t$, $yMin^3 = (11011)^t$, $yMin^4 = (11101)^t$}

Six different fundamental sets are built, two per sense.

Sense 1

FSS1Max = { $(x^1,yMax^1)$, $(x^2,yMax^2)$, $(x^3,yMax^3)$, $(x^4,yMax^4)$, $(x^5,yMax^5)$ }

FSS1Min = { $(x^1,yMin^1)$, $(x^2,yMin^2)$, $(x^3,yMin^3)$, $(x^4,yMin^4)$, $(x^5,yMin^5)$ }

Sense 2

FSS2Max = { $(x^1,yMax^1)$, $(x^2,yMax^2)$, $(x^3,yMax^3)$, $(x^4,yMax^4)$ }

FSS2Min = { $(x^1,yMin^1)$, $(x^2,yMin^2)$, $(x^3,yMin^3)$, $(x^4,yMin^4)$ }

Sense 3

FSS3Max = { $(x^1,yMax^1)$, $(x^2,yMax^2)$, $(x^3,yMax^3)$, $(x^4,yMax^4)$ }

FSS3Min = { $(x^1,yMin^1)$, $(x^2,yMin^2)$, $(x^3,yMin^3)$, $(x^4,yMin^4)$ }

5. For each fundamental set, the corresponding associative memory types Max and Min are built according to step 2 of sections C.1 and C.2of their respective learning phases. At the end, two associative memories have been built for each sense. We show the building of the matrices corresponding to the third sense (MMax3 and MMin3).

Step 1:

$$MMax3^1 = \begin{pmatrix}1\\0\\0\\0\end{pmatrix} \boxtimes \begin{pmatrix}0 & 1 & 1 & 0 & 1 & 1 & 0 & \dots & 0\end{pmatrix} = \begin{pmatrix}2 & 1 & 1 & 2 & 1 & 1 & 2 & \dots & 2\\1 & 0 & 0 & 1 & 0 & 0 & 1 & \dots & 1\\1 & 0 & 0 & 1 & 0 & 0 & 1 & \dots & 1\\1 & 0 & 0 & 1 & 0 & 0 & 1 & \dots & 1\end{pmatrix}$$

$$MMin3^1 = \begin{pmatrix}0\\1\\1\\1\end{pmatrix} \boxtimes \begin{pmatrix}0 & 1 & 1 & 0 & 1 & 1 & 0 & \dots & 0\end{pmatrix} = \begin{pmatrix}1 & 0 & 0 & 1 & 0 & 0 & 1 & \dots & 1\\2 & 1 & 1 & 2 & 1 & 1 & 2 & \dots & 2\\2 & 1 & 1 & 2 & 1 & 1 & 2 & \dots & 2\\2 & 1 & 1 & 2 & 1 & 1 & 2 & \dots & 2\end{pmatrix}$$

$$MMax3^2 = \begin{pmatrix}0\\1\\0\\0\end{pmatrix} \boxtimes \begin{pmatrix}0 & 1 & 1 & 0 & 1 & 0 & 0 & \dots & 0\end{pmatrix} = \begin{pmatrix}1 & 0 & 0 & 1 & 0 & 1 & 1 & \dots & 1\\2 & 1 & 1 & 2 & 1 & 2 & 2 & \dots & 2\\1 & 0 & 0 & 1 & 0 & 1 & 1 & \dots & 1\\1 & 0 & 0 & 1 & 0 & 1 & 1 & \dots & 1\end{pmatrix}$$

$$MMin3^2 = \begin{pmatrix}1\\0\\1\\1\end{pmatrix} \boxtimes \begin{pmatrix}0 & 1 & 1 & 0 & 1 & 0 & 0 & \dots & 0\end{pmatrix} = \begin{pmatrix}2 & 1 & 1 & 2 & 1 & 2 & 2 & \dots & 2\\1 & 0 & 0 & 1 & 0 & 1 & 1 & \dots & 1\\2 & 1 & 1 & 2 & 1 & 2 & 2 & \dots & 2\\2 & 1 & 1 & 2 & 1 & 2 & 2 & \dots & 2\end{pmatrix}$$

$$MMax3^3 = \begin{pmatrix}0\\0\\1\\0\end{pmatrix} \boxtimes \begin{pmatrix}0 & 1 & 1 & 0 & 0 & 0 & 1 & \dots & 0\end{pmatrix} = \begin{pmatrix}1 & 0 & 0 & 1 & 1 & 1 & 0 & \dots & 1\\1 & 0 & 0 & 1 & 1 & 1 & 0 & \dots & 1\\2 & 1 & 1 & 2 & 2 & 2 & 1 & \dots & 2\\1 & 0 & 0 & 1 & 1 & 1 & 0 & \dots & 1\end{pmatrix}$$

$$MMin3^3 = \begin{pmatrix}1\\1\\0\\1\end{pmatrix} \boxtimes \begin{pmatrix}0 & 1 & 1 & 0 & 0 & 0 & 1 & \dots & 0\end{pmatrix} = \begin{pmatrix}2 & 1 & 1 & 2 & 2 & 2 & 1 & \dots & 2\\2 & 1 & 1 & 2 & 2 & 2 & 1 & \dots & 2\\1 & 0 & 0 & 1 & 1 & 1 & 0 & \dots & 1\\2 & 1 & 1 & 2 & 2 & 2 & 1 & \dots & 2\end{pmatrix}$$

$$MMax3^4 = \begin{pmatrix}0\\0\\0\\1\end{pmatrix} \boxtimes \begin{pmatrix}0 & 1 & 1 & 0 & 1 & 1 & 1 & \dots & 0\end{pmatrix} = \begin{pmatrix}1 & 0 & 0 & 1 & 0 & 0 & 0 & \dots & 1\\1 & 0 & 0 & 1 & 0 & 0 & 0 & \dots & 1\\1 & 0 & 0 & 1 & 0 & 0 & 0 & \dots & 1\\2 & 1 & 1 & 2 & 1 & 1 & 1 & \dots & 2\end{pmatrix}$$

$$MMin3^4 = \begin{pmatrix}1\\1\\1\\0\end{pmatrix} \boxtimes \begin{pmatrix}0 & 1 & 1 & 0 & 1 & 1 & 1 & \dots & 0\end{pmatrix} = \begin{pmatrix}2 & 1 & 1 & 2 & 1 & 1 & 1 & \dots & 2\\2 & 1 & 1 & 2 & 1 & 1 & 1 & \dots & 2\\2 & 1 & 1 & 2 & 1 & 1 & 1 & \dots & 2\\1 & 0 & 0 & 1 & 0 & 0 & 0 & \dots & 1\end{pmatrix}$$

Step 2:

$$MMax3 = \begin{pmatrix}2 & 1 & 1 & 2 & 1 & 1 & 2 & \dots & 2\\2 & 1 & 1 & 2 & 1 & 2 & 2 & \dots & 2\\2 & 1 & 1 & 2 & 2 & 2 & 2 & \dots & 2\\2 & 1 & 1 & 2 & 2 & 2 & 1 & \dots & 2\end{pmatrix}$$

$$MMin3 = \begin{pmatrix}1 & 0 & 0 & 1 & 0 & 0 & 1 & \dots & 1\\1 & 0 & 0 & 1 & 0 & 1 & 1 & \dots & 1\\1 & 0 & 0 & 1 & 1 & 1 & 0 & \dots & 1\\1 & 0 & 0 & 1 & 0 & 0 & 0 & \dots & 1\end{pmatrix}$$

The learning matrices MMax1, MMin1, MMax2, MMin2 are computed in the same fashion.

6. In order to assign a sense to an ambiguous word, its context words are presented to each pair of associative

ISSN 2395-8618

memories. Given that each associative memory corresponds to a sense, the resulting output vectors represent the relation of the context word with said sense. In this example, we present c^3 vector to the MMax3 and MMin3 matrices.

Step 1 Max:

$$MMax3\Delta_\beta c^3 = \begin{pmatrix} 2 & 1 & 1 & 2 & 1 & 1 & 2 & \dots & 2 \\ 2 & 1 & 1 & 2 & 1 & 2 & 2 & \dots & 2 \\ 2 & 1 & 1 & 2 & 2 & 2 & 2 & \dots & 2 \\ 2 & 1 & 1 & 2 & 2 & 2 & 1 & \dots & 2 \end{pmatrix} \Delta_\beta \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ \vdots \\ 0 \end{pmatrix}$$

$$= \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

Step 2 Max:

$$z^3 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, s = \begin{pmatrix} 23 \\ 45 \\ 14 \\ 21 \end{pmatrix}$$

$$yMax^3 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

Step 1 Min:

$$MMin3\nabla_\beta c^3 = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & \dots & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & \dots & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & \dots & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & \dots & 1 \end{pmatrix} \nabla_\beta \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ \vdots \\ 0 \end{pmatrix}$$

$$= \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

Step 2 Min:

$$z^3 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}, s = \begin{pmatrix} 23 \\ 45 \\ 14 \\ 21 \end{pmatrix}$$

$$yMin^3 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

7. To adjust the resulting output vectors, derived from the recall phase of the associative memory type Min (in correspondence to the output vectors from associative memory type Max), all their components are negated. This is, a zero value is exchanged for 1, and vice versa.

$$yMin^3 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

8. For all output vectors related to each learning matrix, the sum of all components equal to 1 are computed (voting):

Learning Matrix	Context vector	Output vector	Sum of components
MMax1	c^1	$(00000)^t$	0
	c^2	$(00000)^t$	0
	c^3	$(00000)^t$	0
MMin1	c^1	$(00000)^t$	0
	c^2	$(00000)^t$	0
	c^3	$(00000)^t$	0
		Total	**0**
MMax2	c^1	$(0000)^t$	0
	c^2	$(0000)^t$	0
	c^3	$(0000)^t$	0
MMin2	c^1	$(0000)^t$	0
	c^2	$(0000)^t$	0
	c^3	$(0000)^t$	0
		Total	**0**
MMax3	c^1	$(0100)^t$	1
	c^2	$(0010)^t$	1
	c^3	$(0000)^t$	0
MMin3	c^1	$(0100)^t$	1
	c^2	$(0010)^t$	1
	c^3	$(0000)^t$	0
		Total	**4**

9. The sense corresponding to the learning matrix that has the greatest votes is selected as the correct sense. If more than one sense is selected, then the method is considered unable to determine the sense for the ambiguous word. In this example, the sense selected for the ambiguous word, with a score of four, is the third sense.

IV. Experiments

The performance of the proposed algorithm was assessed using a semantically annotated corpus for SENSEVAL-2 English all-words task [17], and it was compared with results from the simple Lesk algorithm.

SENSEVAL-2 is a dataset that consists of three documents with 2,456 words in 238 sentences. It consists of three tasks: 1) "all-words", "lexical sample", and "translation task". Our comparison is extracted from performances on the "all-words" task. Our proposal, as with any other knowledge-based algorithm, uses a machine readable dictionary; in this case, we used WordNet.

To measure the performance of the two algorithms, the statistical metrics precision, recall, and F1-score were employed. They are statistical measures that evaluate several aspects of the algorithms [18].

Precision indicates the fraction of retrieved instances that are relevant. This is determined by the number of correct answers, divided by the number of answers given by the algorithm.

ISSN 2395-8618

Recall is the fraction of relevant instances that are retrieved, and is computed by the number of correct answers, divided by the total number of words for which there is an answer.

F1-score is considered as a weighted average of precision and recall. It is determined by (2PR) / (P + R).

Then, for each sentence in the corpus, and for each word in the sentence, the word to be evaluated (the ambiguous word) is separated from the surrounding words (context). Usually, the senses of each word are expressed in a dictionary (WordNet), as a definition or gloss. In addition to the gloss, there is other information that could be used as an addendum to increase the performance of the disambiguation algorithms. Examples of such information are Synonyms (Syns) and Hyponyms (Hypo). The former, are sets of words that have similar meanings, the latter is a set of more specific synonyms.

Four different experiments were prepared using the information source mentioned before:

1) Gloss (G): only the information of the gloss

2) Gloss + Syns (G+S): the synonyms of the ambiguous word added to its own gloss.

3) Gloss + Hypo (G+H): the hyponyms of the ambiguous word added to its own gloss.

4) Gloss + Syns + Hypo (G+S+H): The gloss of the word added to the hyponyms and synonyms.

V. Results and Discussion

Tables II, III, IV, and V show the results of different experiments, comparing our implementation of the simple Lesk algorithm (SL) against the proposed method (AM). The experiments were developed using two parameters as variables: context window and stemming. It is worth noting that the algorithms presented in these tables did not use a back-off strategy.

The context window is the number of sentences used to disambiguate a word. The possible values for this are: one sentence (which is where the ambiguous word is), and three sentences (the sentence where the ambiguous word is, the one after, and the one before). There are two special cases in context selection: a) when the ambiguous word is in the first sentence, and b) when it is in the last sentence. For both cases, only two sentences are considered: in the first sentence, the window is composed using the sentence with the ambiguous word and its following one. For the last sentence, the context window is the sentence with the ambiguous word and its preceding one. For each configuration, precision, recall, and F1-score were computed.

On the other hand, stemming represents the reduction of a word into a base form. This reduction could be applied (or not) to the context and ambiguous words before the disambiguation process.

Tables II and III show the experiments using the gloss (table II), and gloss and synonyms (table III), as the source of information to form the fundamental set of associative memories. Both tables show that in the precision metric the simple Lesk algorithm performs better than our proposal in each experiment; this means that the simple Lesk algorithm is more assertive when assigning a sense to a word. However our proposal assigns a sense to more words, according to the recall results. In addition, our proposal presents better results as tested using F1-score metric. We can thus conclude from these results that: a) considering the tradeoff between precision and recall, our proposal performs better, and b) our proposal is less dependent on the stemming process, given that the differences between F1-score with and without stemming are smaller than the ones reported from using the simple Lesk algorithm.

TABLE II
Results using the gloss of the ambiguous word

	Context window	Stemming	Precision	Recall	F1-Score
AM	1	Yes	42.11	17.56	24.78
SL	1	Yes	53.88	10.38	17.40
AM	1	No	49.18	15.47	23.53
SL	1	No	55.93	7.86	13.78
AM	3	Yes	47.86	25.34	33.13
SL	3	Yes	56.33	17.86	27.12
AM	3	No	53.07	22.91	32.00
SL	3	No	55.05	13.97	22.28

TABLE III
Results using the gloss and synonyms of the ambiguous word

	Context window	Stemming	Precision	Recall	F1-Score
AM	1	Yes	43.67	18.29	25.78
SL	1	Yes	54.97	10.64	17.82
AM	1	No	50.47	16.03	24.33
SL	1	No	56.20	8.33	14.50
AM	3	Yes	48.48	25.85	33.72
SL	3	Yes	57.29	18.46	27.92
AM	3	No	54.27	23.63	32.92
SL	3	No	56.73	14.96	23.67

Table IV reports the results of when the fundamental set was constructed using the gloss and hyponyms. It shows that each metric had a better performance compared with the ones presented in table II and III, maintaining observed patterns. This is, the simple Lesk outperforms our proposal in precision, but our proposal performs better in recall and F1-score. In addition, it is worth noting that the AM with a context window of three, without stemming, surpasses the SL in precision.

Table V presents the results of when the fundamental set was the compound of the gloss, synonyms, and hyponyms. As opposed to table III and IV, which present an increased performance when more information was included in the fundamental set, table V presents a decrease in performance

ISSN 2395-8618

of all simple Lesk experiments in relation to table IV, whereas just one AM experiment shows this performance decrement. Moreover, as is the same as table IV, the AM presents a better performance in all F1-scores and presents one case where the AM precision is better than the simple Lesk Algorithm.

TABLE IV
RESULTS USING THE GLOSS AND HYPONYMS OF THE AMBIGUOUS WORD

	Context window	Stemming	Precision	Recall	F1-Score
AM	1	Yes	45.07	19.15	26.87
SL	1	Yes	53.89	10.94	18.18
AM	1	No	52.34	17.65	26.39
SL	1	No	56.37	8.50	14.77
AM	3	Yes	51.49	28.08	36.34
SL	3	Yes	56.55	18.63	28.02
AM	3	No	57.12	25.90	35.63
SL	3	No	56.67	14.70	23.34

TABLE V
RESULTS USING THE GLOSS, SYNONYMS AND HYPONYMS OF THE AMBIGUOUS WORD

	Context window	Stemming	Precision	Recall	F1-Score
AM	1	Yes	46.21	19.79	27.71
SL	1	Yes	53.77	10.98	18.23
AM	1	No	53.55	18.03	26.97
SL	1	No	56.02	8.55	14.83
AM	3	Yes	51.41	28.03	36.27
SL	3	Yes	55.80	18.72	28.03
AM	3	No	57.70	26.11	35.95
SL	3	No	55.97	14.83	23.44

Meanwhile, Table VI shows the results of different experiments, comparing the SL algorithm against AM using random sense as a back-off strategy. The F1-score was computed for each information source (G, G+S, G+H, G+S+H), using one and three sentences as context window, and with or without stemming. These experiments exhibit that the AM algorithm does not outperform the SL algorithm for all cases but one, when the context window size is one sentence, without using stemming, and "G+H" as information source, being the F1-score of 45.98.

On the other hand, Table VII presents the results of the AM algorithm compared with two state-of-art algorithms, the random base line (RBL), and the simple Lesk algorithm. The state-of-art algorithms are: 1) a modified implementation of simple Lesk algorithm which instead of selecting the neighboring sentences as context window, it builds its own context by selecting the words that do overlap at least in one word with any gloss of the target word [19]; and 2) a word sense disambiguation algorithm based on Bayes' theorem which compute the a posteriori probabilities of the senses of a polysemous word, then, the sense selected for a given ambiguous word is that with the greater probability [20] (hereinafter Modified SLA and NaiveBayesSM, respectively). These results show that the AM performs better in three out of four algorithms presented, but it is below to Modified SLA which presents an F1-score of 47.8.

TABLE VI
RESULTS USING RANDOM SENSE AS A BACK-OFF STRATEGY

	Context window	Stemming	G	G+S	G+H	G+S+H
AM	1	Yes	44.06	43.46	43.97	44.02
SL	1	Yes	43.76	45.00	46.54	45.04
AM	1	No	45.30	44.57	45.98	45.90
SL	1	No	44.27	44.02	44.27	43.76
AM	3	Yes	42.78	43.42	44.49	43.29
SL	3	Yes	47.31	47.52	47.14	46.50
AM	3	No	44.15	43.80	44.49	45.00
SL	3	No	43.93	46.54	45.81	46.58

TABLE VII
STATE-OF-ART COMPARISON

	Context window	Stemming	F1-Score
Modified SLA	1	-	47.8
AM	1	No	45.98
SL	1	No	44.27
RBL	-	-	41.22
NaiveBayesSM	1	Yes	36.2

VI. CONCLUSIONS AND FUTURE WORK

Tables II to V present, among others metrics, the F1-scores computed for both the associative memory and the original simple Lesk approach. These show that the AM performs better than the simple Lesk algorithm in all cases. In respect to the precision metric, even when the simple Lesk algorithm performs better than our proposal, Tables IV and V show two cases where the associative memory approach outperforms it. These two cases share a context window size of three (the greatest size presented in this work), and the stemming process was not applied. From this, it may be aptly concluded that, in contrast to the simple Lesk algorithm, the associative memory approach is beneficial when more information is available. Furthermore, its performance is not severely reduced when stemming is not applied.

On the other hand, Tables IV and V present interesting outcomes: it seems that, the more data entered in the simple Lesk algorithm for the “bag of words”, the more its precision was decreased. If, for both tables, the experiments that correspond to equal size context window –with the same stemming option– are compared, we notice that the simple Lesk algorithm has a reduced precision, if the gloss, synonyms, and hyponyms conform to the bag of words.

Subsequently, Table VI show that when applying the random sense back-off strategy, the SL reports a greater F1-score except for one instance. It is important to note however,

ISSN 2395-8618

that most of the cases where the SL performs better (Table VI) are those where the SL without back-off strategy (Table V), presented a bigger F1-score difference between both algorithms. Therefore, it is possible to infer that when combining a back-off strategy with the SL algorithm, the smaller the F1-score, the fewer decisions are taken by it, then, the back-off randomly choose a sense, and, if the target word has a few senses, it is more likely select the correct one; improving the overall performance. The only instance where the AM comes out better is that where F1-score presents a shorter difference between AM and SL (Table V).

Finally, even when random sense back-off strategy is combined with AM, it does not succeed over Modified SLA. It may be because of the words with which the context are built, are those that appear, at least, one time in any gloss of the word to disambiguate; being more likely selecting the correct sense when the gloss shares one word than those that does not.

In future work, a search for different binary codifications will be made; then, their corresponding implementations will be tested to find the codification that best fit the disambiguation purposes. Another interesting approach to research involves changing the lexical resources (dictionaries), and performing a set of experiments to identify the advantages and disadvantages that are present in each of them. Also, it would be interesting to combine the context building strategy presented by Viveros-Jimenez et al. [19] with our proposal. Finally, in order to increase the response time of the algorithm, a CUDA implementation of our proposal will be made.

In addition, on our future work we plan to explore the role of our WSD method in important tasks where the meaning of ambiguous words plays an important role, such as sentiment analysis [21], [22], [23], sarcasm detection [24], and textual entailment [25].

REFERENCES

[1] G. Hirst, E. Hovy, and M. Johnson, "Theory and Applications of Natural Language Processing", 2013.

[2] R. Navigli, and N. Lapata, "An experimental study of graph connectivity for unsupervised word sense disambiguation", *IEEE transactions on pattern analysis and machine intelligence,* vol. 32, num. 4, pp. 678-692*,* 2010.

[3] P.P. Borah, G. Talukdar, and A. Baruah, "Approaches for Word Sense Disambiguation–A Survey", *International Journal of Recent Technology and Engineering*, vol. 3, no. 1, 35–38, 2014.

[4] R. Navigli, "A quick tour of word sense disambiguation, induction and related approaches", In *International Conference on Current Trends in Theory and Practice of Computer Science,* Springer, pp. 115-129, 2012.

[5] T. Kohonen, *"Self-organization and associative memory"*, Springer-Verlag, Berlin. 1989.

[6] M.E. Acevedo-Mosqueda, C. Yáñez-Márquez, and I. López-Yáñez, "Alpha–Beta bidirectional associative memories: theory and applications", *Neural Processing Letters*, vol. 26, no 1, p. 1-40, 2007.

[7] H. Jarollahi, N. Onizawa, V. Gripon, et al. "Algorithm and architecture of fully-parallel associative memories based on sparse clustered networks". *Journal of Signal Processing Systems*, vol. 76, no 3, p. 235-247, 2014.

[8] G. Palm, "Neural associative memories and sparse coding" *Neural Networks*, vol. 37, pp. 165-171*,* 2013.

[9] I. Román-Godínez, I. López-Yánez, and C. Yánez-Márquez. "Classifying patterns in bioinformatics databases by using Alpha-Beta associative memories." In *Biomedical Data and Applications*, Springer, pp. 187-210, 2009.

[10] I. Román-Godínez, "Identification of functional sequences using associative memories" *Revista Mexicana de Ingeniería Biomédica,* vol. 32, no. 2, pp. 109-118, December, 2011.

[11] A. Argüelles, C. Yáñez, I. López, and O. Camacho, "Prediction of CO and NOx Levels in Mexico City Using Associative Models." *Artificial Intelligence Applications and Innovations*, vol. 364, pp. 313-322, 2011.

[12] M. Lesk, "Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone", in *Proc. of the 5th annual international conference on Systems documentation*, pp. 24-26*,* 1986.

[13] A. Kilgarriff, and J. Rosenzweig, "Framework and results for English SENSEVAL", *Computers and the Humanities*, vol. 34, no. 1-2, pp. 15-48, 2000.

[14] C. Yánez, "Memorias Asociativas basadas en Relaciones de Orden y Operadores Binarios", Ph.D. thesis. CIC-IPN, Mexico, 2002.

[15] G. X. Ritter, P. Sussner, and J.L. Diaz-de-Leon, "Morphological associative memories", *IEEE Transactions on Neural Networks*, vol. 9, pp. 281-293, 1998.

[16] I. Román-Godínez, and C. Yáñez-Márquez, "Complete recall on Alpha-Beta heteroassociative memory". In *Proc. MICAI*, pp. 193-202*,* 2007.

[17] M. Palmer, C. Fellbaum, S. Cotton, L. Delfs, and H. T. Dang, "English tasks: All-words and verb lexical sample", in *Proc. SENSEVAL-2,* pp. 21-24*, 2001*.

[18] R. Navigli, "Word sense disambiguation: A survey", *ACM Computing Surveys (CSUR)*, vol. 41, no. 2*,* 2009.

[19] F. Viveros-Jiménez, A. Gelbukh, and G. Sidorov. "Simple window selection strategies for the simplified lesk algorithm for word sense disambiguation". In *Mexican International Conference on Artificial Intelligence,* Springer*,* pp. 217-227, 2013.

[20] T. Wang and G. Hirst. Applying a Naive Bayes Similarity Measure to Word Sense Disambiguation. In *ACL* (2), pp. 531-537, 2014.

[21] S. Poria, E. Cambria, A. Gelbukh, F. Bisio, and A. Hussain. "Sentiment data flow analysis by means of dynamic linguistic patterns", *IEEE Computational Intelligence Magazine*, vol. 10, no. 4, pp. 26-36, 2015.

[22] S. Poria, E. Cambria, and A. Gelbukh. "Aspect extraction for opinion mining with a deep convolutional neural network", *Knowledge-Based Systems*, vol. 108, pp. 42-49, 2016.

[23] E. Cambria, S. Poria, R. Bajpai, and B. Schuller. "SenticNet 4: A semantic resource for sentiment analysis based on conceptual primitives". In: *COLING 2016*, Osaka, pp. 2666-2677, 2016.

[24] S. Poria, E. Cambria, D. Hazarika, and P. Vij. "A deeper look into sarcastic tweets using deep convolutional neural networks". In: *COLING 2016*, Osaka, pp. 1601-1612, 2016.

[25] P. Pakray, S. Neogi, P. Bhaskar, S. Poria, S. Bandyopadhyay, and A. Gelbukh. "A textual entailment system using anaphora resolution". In: *System Report. Text analysis conference recognizing textual entailment track notebook*, 2011.

ISSN 2395-8618

A Method Based on Genetic Algorithms for Generating Assessment Tests Used for Learning

Doru Popescu Anastasiu, Nicolae Bold, and Daniel Nijloveanu

***Abstract*—Tests are used in a variety of contexts in the activity of everyday and everywhere learning. They are a specific method in the process of assessment (evaluation), which is an important part of the educational activity. Setting an optimized sequence of tests (SOT) originating from a group of tests which have the same subject, with certain restrictions corresponding to a certain wish of the evaluator can be a slowly time-consuming task, because the restriction can be various and the number of tests can be high. In this matter, this paper presents a method of generating optimized sequences of tests within a battery of tests using a genetic algorithm. We associate a number of representative keywords with a test. The user expresses the restriction by setting up a number of keywords which approximate best the subject wanted to be tested. The genetic algorithm helps in finding the optimized solutions and uses a less amount of hardware resources.**

***Index Terms*—Test, genetic algorithm, keyword, sequence, generation.**

I. INTRODUCTION

THE process of learning is very complex and has three major components: teaching, learning and evaluation. This paper focuses on the third component and, in particular, on tests and their usage based on a customization given by the user, taking into account keywords, which are part of natural lexicon. We can specify that the solution proposed in the paper is immediate and needful.

Firstly, the emergence and fast development of devices creates a learning current which have some particularities (fast learning, huge amounts of data obtained relatively fast, knowledge based on competences, less than on accumulation of information etc.). This solution adapts on this intrusion of technology in learning.

Moreover, the time and energy consumed for the manual trial of the tests is obvious. Finding a solution which decreases this time is a need in a climate where every minute counts. For example, for a usual exam, a teacher creates tests with 10 questions that form a battery of tests. These tests have various subjects. In the process of creation of tests, the teacher applies "labels" to each test. For a manual search of tests, we assume that the verification for one test consumes an average of 1,5 minutes. For a battery of 100 tests, the verification consumes a total of 150 minutes. Using this application, the generation and the identification of tests within a well-organized battery consumes maximum 10 minutes. Furthermore, the teacher must make an additional search for different labels, while the generation is made faster.

From the point of view for the student, the solution presented in this paper adapts to the adaptive learning style, presented in detail in section 2. In short, the student can organize the self-assessment process (e.g., learning for an exam) on subjects. Furthermore, if a battery of tests contains tests with various subjects and the student wishes to prepare only for certain subjects in a limited time, the problem escalates. For example, in a battery of 1000 tests regarding programming languages, containing tests about C++, Java, PHP and JavaScript, the student must verify knowledge about Java. The solution can be applied for solving this problem.

Regarding types of tests, they can be various (simple questions tests, multiple-choice tests etc.). But, generally, as human beings, we learn things every moment. This learning process, also called self-learning, includes a constant permanent self-evaluation, even if it is formal, informal or our teacher is ourselves. We all participated at least at one exam or test – this can be called a formal evaluation. Our knowledge is tested by creating things and dealing with situations in real life and this could be considered a less formal examination. The perception is different for students and tutors and, in this matter, a study in the paper [1] shows these differences in perception.

The evaluation is made through several methods, some traditional, some novel (a presentation of new methods in assessment can be found in the book [2] and the perspective of the students on the new methods can be found in [3]). One of the traditional ones is the test (even if it is a multiple-choice test, a question test or problem solve test). Tests can be categorized in a formal register, when the mark obtained by the learner is important for a legal purpose, but also in a less informal register, in case of self-learning and self-evaluation. The purpose is to generate optimized test sequences which

Manuscript received on February 24, 2016, accepted for publication on June 16, 2016, published on October 30, 2016.

Doru Popescu Anastasiu is with the University of Pitesti, Faculty of Mathematics and Computer Science, Romania (e-mail: dopopan@gmail.com).

Nicolae Bold is with the University of Agronomic Sciences and Veterinary Medicine Bucharest, Faculty of Management, Economic Engineering in Agriculture and Rural Development, Slatina Branch, Romania (e-mail: bold_nicolae@yahoo.com).

Daniel Nijloveanu is with the University of Agronomic Sciences and Veterinary Medicine Bucharest, Faculty of Management, Economic Engineering in Agriculture and Rural Development, Slatina Branch, Romania (e-mail: nijloveanu_daniel@yahoo.com).

ISSN 2395-8618

contain the maximum number of keywords from the keywords set by the user.

The learners deal with a high number of tests in their learning process. In some cases, the tests are grouped in batteries or clusters of tests. These tests are not related at all, as in the problem studied in paper [4], where an arborescent structure existed. The particular grouping of tests in clusters makes difficult the process of finding tests with a certain subject. Thus, this paper presents a solution of a fast finding of wanted tests with the desired subject. This is made by previously assigning keywords to tests, then giving some keywords as input data and finding the ones whose keywords match with the given keywords. Thus, the problem is solved using lexical resources. Moreover, the usage of keywords encapsulates the concept of summarization. The results are given in the form of optimized sequences of numbers which codify the tests.

Regarding the problem of summarization and lexical resources, we can say that keywords represent the words that are essential for defining a test. They summarize best what a test contains, turning into lexical resources which are key elements in the solution of the problem.

The optimality refers to the finding of the maximum number of tests that can be selected after the required conditions are respected. Thus, the fitness is a maximum function and the problem can be classified as an optimization problem.

Even if the issue does not appear to be immediate and needful, the trial of tests consumes time and energy. In this matter, an algorithmically solution of this problem would bring a plus of efficiency in the process of evaluation, as shown in the previous paragraphs. The genetic method for generating the sequences was chosen for its performance in the usage of fewer hardware resources and for its variability of output solutions.

Section 2 will contain a short description of testing method and some measurable data to show its efficiency in the evaluation process. In Section 3, we will present some notions regarding genetic algorithms and their role in solving problems. The algorithm description takes place in Section 4 and Section 5 will describe some results regarding the algorithm efficiency and the productivity of the algorithm in solving the problem. Section 6 will draw the lines of this paper and will present the future work which will be made in this matter.

II. On the Impact of Tests, Evaluation and Information and Communication Technologies (ICT) on Education

The main idea of this paper related to evaluation process, in particular, and education, in general, can be viewed from various angles and has more perspectives.

One of them is the personalization of the subjects desired to be learned, either by a teacher or by a student. Regarding the latter category, a specific type of learning style has been implemented in several universities or faculties worldwide, where the student chooses the subjects he wishes to use in the future and the education is made using computers as teaching devices. Roughly, this process can be called adaptive learning style. In a similar way, choosing a sequence which suits best to specific needs foe evaluation will maximize the effectiveness in learning. In a study made in the paper [5] a comparison between an adaptive learning style and a non-adaptive one is realized. This study, based on measuring some key characteristics that influence learning, revealed that a high percent of students (90.63%) think that learning styles are important for learning. Another conclusion of the study is focusing on the fact that students would prefer to have recommended paths for learning, but this should be chosen by the student. The fact of self-choices has several aspects and part of them have not positive effects over the learning status of the students. Thus, some major risks in this case are:

- faulty choices of what is needed for learning due to the immaturity of children or depending on the pupil/student personality;
- lack of human communication, which is partly substituted with modern communication technologies, and its further implications.

Another perspective of using Information and Communication Technologies (ICT) in learning and evaluation is the perception of the student on the teaching style, because of the modern characteristic of using technology [6]. Because they are more familiar with the technology, students perceive this step to modern techniques of teaching as receptivity to innovation from the teacher. From a statistical point of view, in the same study, made on a total of 226 students, it was shown the attitude of students towards the school activity has also shown to be improved in the studied group.

A very interesting perspective is the creation of an open learning environment [7], which appears in case of using technology in education. The development of technology and the equipment, which has more and more influence on the society, in general (Internet, educational platforms and software on one hand and gadgets such as tablets, smart phones, laptops, projectors on the other hand), brings into attention notions regarding a specific new environment, defined in this paper as a combination between social technological and pedagogical factors which influence each other and which influence educations.

Examples of open-learning environments are the e-learning platforms. Studies regarding the inclusion of ICT in education have led to the apparition of e-learning platforms. The usage of Internet in the process of education is more and more visible today. An example of a study and of a model for an e-learning platform, with its threats is detailed in papers [8] and [9].

ISSN 2395-8618

Regarding the particular process of assessment, in the context of the combination between the three processes of education, numerous results of studies and data from literature show the fact that ICT has proven its benefits to evaluation in the matter of online and technology-based assessments versus the traditional methods of student evaluation, as proved in [10], [11] or [12]. Despite these results, there is not known the effect on a long-term period, so we should monitor the effects on a longer period than the time accorded for learning for an exam, possibly showing their efficiency in more practical situations.

The perspectives on ICT influence on the educational process are numerous and have both positive and negative implications on the personality of the student. As the authors in the paper [13] show, there are many questions to be answered in the matter of efficiency of ICT based methods in teaching, learning and evaluation. Thus, we must find answers to questions such as "how can the problem of communication be solved?", "how can practical abilities (e.g., crafting) can be developed?", "how can we measure the efficiency and the added value of educational methods based explicitly on ICT?" or "how can we improve the security of assessment structures?".

III. GENETIC ALGORITHMS

Problems which appear in practice can be solved using different methods, algorithms and structures. Their variety is wide, starting with trees and graphs, continuing with backtracking and greedy algorithms and finishing with random or genetic methods of solving problems. Among them, genetic algorithms are used for problems which solve the optimization aspects. They are inspired by genetics and use notions such as chromosomes, genes, mutation and crossover. Next, we will present shortly some problems that can be solved using genetic algorithm, as well as some key characteristics of them.

Firstly, a genetic algorithm uses a lower amount of hardware resources (the runtime is lower). This is a major point in using genetic algorithms, because of its methods, being preferred to other optimization or heuristic algorithms (such as backtracking) in some cases (for a larger number of solutions, in case of large amounts of input data etc.)

Another characteristic of a genetic algorithm is that the problem solved through this method has to be or to be transformed into an optimization problem [14]. This means that genetic algorithms found the most optimized solution in case of minimum and maximum issues.

An eventual drawback could be the fact that genetic algorithms do not find the most accurate solution, but at least they generate solutions that can be found in its proximity. Genetic algorithms are used mostly in cases in which input data is in large quantities, which is the case of the most practical problems needed to be solved.

One of the areas in which genetic algorithm is used is designing constructions. In the paper [15] there is presented a solution for designing the optimized thermal and lightning conditions, construction materials etc. within a building, using genetic algorithms. GA are also used in issues related to domains such as mathematics, statistics, physics, engineering, transportation, pollution cases [16], chemistry [17], agriculture [18], web programming [19], web applications [20] and even fashion issues [21]. This is a very short list of the applications of genetic algorithms in specific domains.

Even with the drawbacks of this method (the found solutions are optimized, but not optimal), we chose genetic algorithms because they offer a variety of solutions with given restrictions, which is the case of our problem. The large number of tests within a battery and the correspondence of genetic structures with the ones of the studied problem are other reasons for choosing genetic algorithms, besides the ones presented in the introduction.

IV. ALGORITHM

As we said before, the algorithm uses genetic notions inspired by biology and in this way it generates the sequences we want to obtain. In another paper [1], the case in which the tests have a tree-designed relationship was studied. Here, we will show the algorithm when tests are not level-related.

The tests will be codified by numbers from 1 to n. The optimized sequence of tests (a chromosome) is an arrangement with k elements, m being given by the user (representing the number of tests within the test battery) of the set {1,2,…,n} and a test (a gene) within an optimized sequence of tests is represented by a number from the set {1,2,…,n}. The fitness for a sequence is represented by the maximum number of keywords within the sequence (the m-arrangement) which correspond with the keywords set by the user. The chromosomes will be ordered by the value of the fitness function. In Figure 1, an example of the structure of a chromosome with 6 genes which will be output as solution is presented.

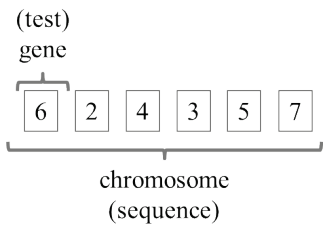

Fig. 1. An example of the general form of an optimized sequence of tests (chromosome) with 6 tests (genes)

Firstly, an array of arrays is initialized with 0 or 1 value (false or true). The purpose of this array is to verify if a keyword which characterizes a test is part of the keywords set

ISSN 2395-8618

by the user. Then, a number from 1 to n, representing a test, is randomly generated and verified if one of its keywords is part of the list of keywords set by the user. After this verification, when the optimized sequence of tests is generated completely, the fitness of this sequence is calculated. The optimized sequences of tests are ordered by the value of the fitness. During the next step, there are made operations such as mutation or crossover between chromosomes (sequences). Then, the resulted sequences are ordered again by the value of the fitness.

The variables used in the algorithm that we will need is presented next:

n: the number of tests within the battery;
m: the number of tests needed in the optimized sequence of tests;
no_generations: the number of generations used to generate the optimized sequences of tests
no_words: the number of keywords set by the user;
no_cuvT: the number of keywords of each test.

The structures (arrays) that will be needed within the algorithm are:

TG[no_words]: the array contains the keywords set by the user;
pop[no_generations][m]: the solution array;
T[n][no_words]: an array of arrays, which has the meaning:

$$T[i][k]=\begin{cases}1, \text{ if the } k^{th}\text{keyword from the } i^{th}\text{test}\\ \quad \text{is found among TG array}\\ 0, \text{otherwise}\end{cases};$$

$$i=\overline{1,n};\ k=\overline{1,no\_cuvT}$$

After presenting the input data, we can say that the fitness has the next form:

$$maxf = \sum_{\substack{1\le j\le m\\ 1\le k\le nr_{cuv}}} T\big[pop[i][j]\big][k];$$

$$i = \overline{1, no\_generations}$$

The input data will be formed from n, m, no_generations, no_words, no_cuvT and the keywords for each test. For avoiding comparing each time the keywords, the array T is built, so there are stored only the keywords which match with the ones from the array TG and the tests containing them. The output data will contain the first k solutions (optimized sequences of tests) from the array pop, where k is a value set by the user, and the number of keywords matching with the ones from TG for each sequence.

After we presented the input and output data needed for our algorithm, we shall present the steps of the algorithm.

Step 1. Input data (mentioned before) is read.

Step 2. The array T is initialized with 0 or 1 (false or true) values, according to the definition presented before.

Step 3. The chromosomes (optimized sequences of tests) are randomly generated, gene by gene. This will be the initial population.

Step 4. The fitness function is calculated for each chromosome. The fitness function is stored in the (m+1)th element of the solution array (pop).

$$pop[m+1] = \sum_{\substack{1\le j\le m\\ 1\le k\le nr\_cuv}} T\big[pop[i][j]\big][k];$$

$$i = \overline{1, no\_generations}$$

Step 5. Operations (mutation and crossover) are applied on the generated chromosomes. The fitness function is calculated for each chromosome in this step too. The fitness function is also stored in the (m+1)-th element of the solution array. Figure 2 presents an example of mutation within a chromosome.

Fig. 2. Mutation within a sequence (chromosome)

Figure 3 presents an instance of the crossover operation with one point between two chromosomes.

Fig. 3. Crossover with one point between two optimized sequences of tests (chromosomes)

Step 6. The chromosomes are ordered by the value of the fitness. At this step, method of order can be used. Steps 5 and 6 are repeated for no_generations times.

Step 7. The first desired solutions are output.

V. Results and Discussion

To show the efficiency in solving a practical problem, we shall take a short example. The battery of test will have as main subject programming languages. Tests have subjects represented by keywords such as software, editing, office, complier, pascal, Java, similarity, syntax, logical, expression, operation, type, protocol, Internet etc., in subdomains such as software in general, programming languages, syntax of languages, memory usage, programming methods, Internet, web programming or database concepts.

ISSN 2395-8618

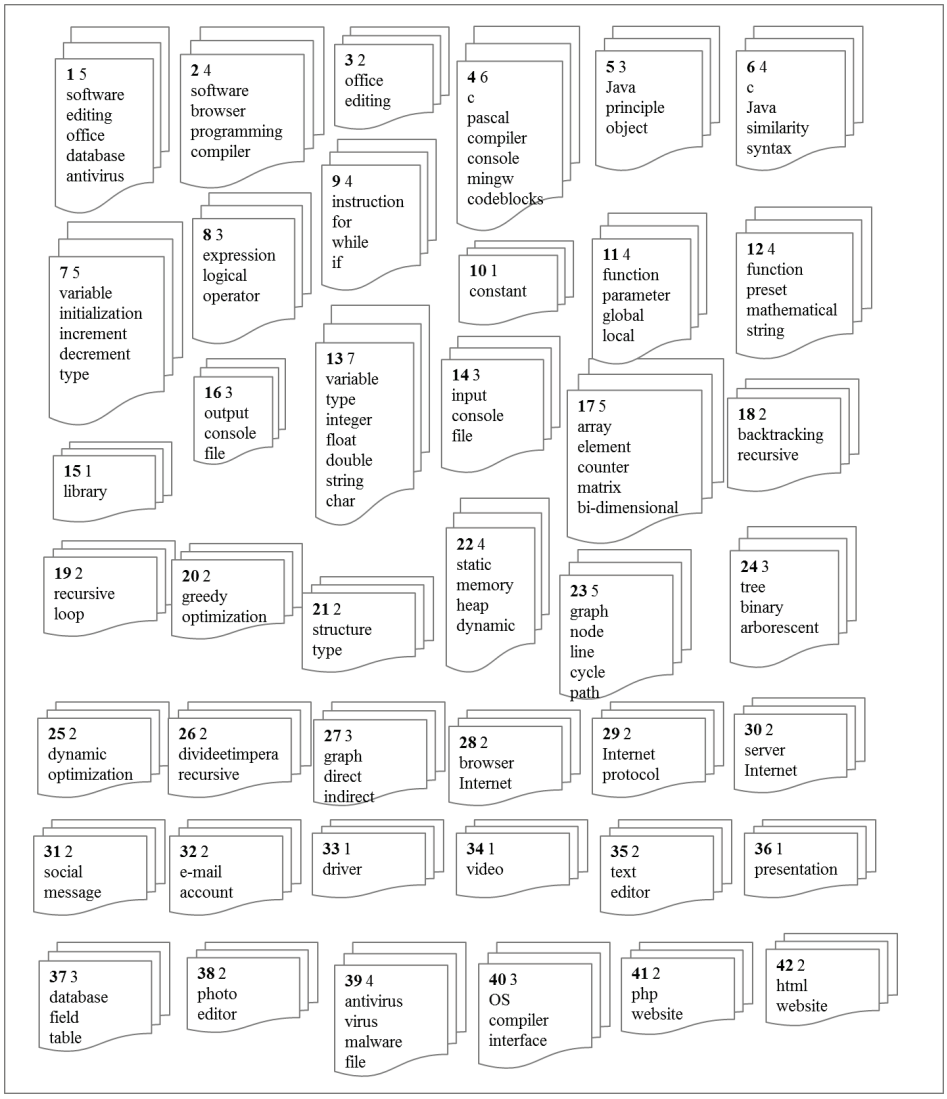

Fig. 4. Battery of 42 tests for our example

For our example, the variables have the next values: n=42, m=15, no_generation=600, no_words=15. The keywords set by the user are:

TG=(software, programming, c, Java, method, structure, variable, backtracking, tree, binary, compiler, instruction, recursive, console, Internet)

The battery of tests contains 42 tests and the keywords that characterize each test are presented in Figure 4. The number in bold characters is the integer assign for the test and the second number represents the number of keywords representative for each test.

The results are divided in two components:

– the output optimized sequences of tests;
– the runtime of the algorithm.

The output in our case is presented in Table II. In case of runtime, we present its dependence on parameters such as the number of tests (n), the number of generations and the number of tests in the optimized sequence of tests (m).

The values were obtained with a code written using the Java programming language, within a Java environment (NetBeans IDE 8.0.2).

The array T stores for each test the keywords that match the keywords from TG. In Table I and II the array T for our example is presented.

The number of keywords from TG which are found in the 42 tests is 22. The first 10 optimized sequences of tests resulted after the program runs are presented in Table III.

The runtime after running the program is 3.877521196 seconds, which shows that the program is efficient regarding the usage of resources. For supporting this affirmation, we present a graph which shows the runtime for different values of n. The values are resulted for m=15 and for no_generations (number of generations) equal to 700. Individual values are an average of 5 values. In the graphs there are made some

ISSN 2395-8618

TABLE I
The array T after running the algorithm

No.	TG 1 2 3 4 5 6 7 8 9 0 11 12 13 14 15
T1	1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 TG1 = software in test 1
T2	1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 TG1,2,11 = software, programming, compiler in test 2
T3	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 None of TG elements in test 3
T4	0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 TG3,11,14 = c, compiler, console in test 4
T5	0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 TG4 = Java in test 5
T6	0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 TG3,4 = c, Java in test 6
T7	0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 TG7 = variable in test 7
T8	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 None of TG elements in test 8
T9	0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 TG12 = instruction in test 9
T10	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 None of TG elements in test 10
T11	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 None of TG elements in test 11
T12	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 None of TG elements in test 12
T13	0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 TG7 = variable in test 13
T14	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 TG14 = console in test 14

No.	TG 1 2 3 4 5 6 7 8 9 0 11 12 13 14 15
T15	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 None of TG elements in test 15
T16	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 TG14 = console in test 16
T17	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 None of TG elements in test 17
T18	0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 TG8,13 = backtracking, recursive in test 18
T19	0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 TG13 = recursive in test 19
T20	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 None of TG elements in test 20
T21	0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 TG6 = structure in test 21
T22	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 None of TG elements in test 22
T23	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 None of TG elements in test 23
T24	0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 TG9,10 = tree, binary in test 24
T25	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 None of TG elements in test 25
T26	0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 TG13 = recursive in test 26
T27	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 None of TG elements in test 27
T28	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 TG15 = Internet in test 28

calculus to show the models of the functions. Figure 5 presents the model of the algorithm runtime for different values of n.

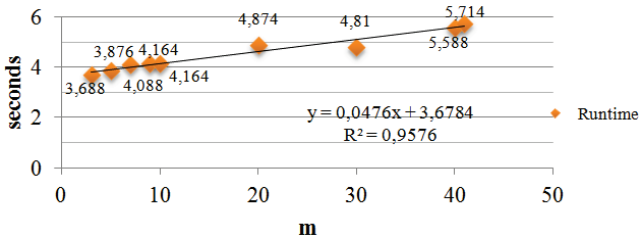

Fig. 5. Runtime depending on the value of m (n=42, no_generations=700)

Figure 6 presents the model of the algorithm runtime for different values of *number of generations*.

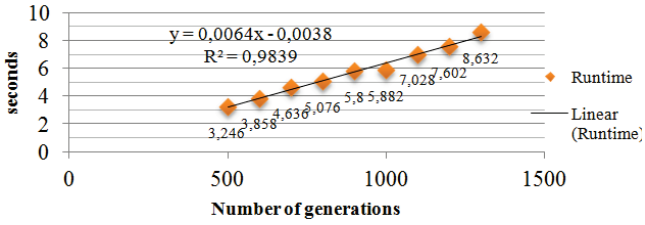

Fig. 6. Runtime depending on the value of no_generations (n=42, m=15)

Figure 7 presents for the model of the algorithm runtime for different values of *m*.

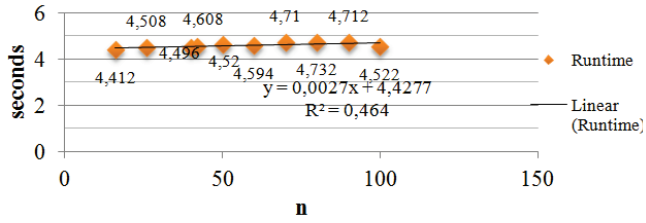

Fig. 7. Runtime depending on the value of n (m=15, no_generations=700)

It can be seen that the models have quite a major significance, depending on the coefficient of correlation R2, especially for models from Figures 6 and 7. All three models are based on a linear regression.

As we can see, the number of tests within the battery does not influence very much the runtime of the algorithm. The number of genes in the chromosome (m) influences in a minor way the runtime, the biggest increase being shown in case in which we want more accurate solutions (in this case, the number of generations increases). Despite this increase, the difference between the first measured element and the latter is 5.836 seconds (from 500 to 1300 generations). This is not an extremely significant difference, given the fact that the number of generations almost triples.

ISSN 2395-8618

TABLE II
The array T after running the algorithm

No.	TG 1 2 3 4 5 6 7 8 9 0 11 12 13 14 15
T29	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 TG15 = Internet in test 29
T30	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 TG15 = Internet in test 30
T31	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 None of TG elements in test 31
T32	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 None of TG elements in test 32
T33	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 None of TG elements in test 33
T34	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 None of TG elements in test 34
T35	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 None of TG elements in test 35
T36	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 None of TG elements in test 36
T37	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 None of TG elements in test 37
T38	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 None of TG elements in test 38
T39	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 None of TG elements in test 39
T40	0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 TG11 = compiler in test 28
T41	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 None of TG elements in test 41
T42	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 None of TG elements in test 42

TABLE III
Output optimized sequences of tests for the example

No.	Optimized sequence of tests
1.	14 6 30 2 18 5 19 4 26 40 24 29 16 9 7
2.	2 6 30 16 19 26 7 5 29 18 4 24 14 40 9
3.	2 40 19 30 7 14 26 16 18 5 6 9 4 24 29
4.	29 26 30 18 2 14 5 16 40 4 7 6 19 24 9
5.	2 5 40 14 4 16 19 30 9 24 26 6 7 18 29
6.	2 29 26 13 9 14 40 5 16 6 24 18 4 30
7.	18 6 26 5 7 14 19 29 4 2 24 30 16 40 9
8.	2 30 5 14 6 16 9 4 26 18 40 19 24 7 29
9.	2 26 30 13 6 14 24 4 18 7 5 40 9 29 16
10.	30 2 18 7 40 14 24 26 16 6 4 19 29 5 9

To summarize, the runtime obtained is far superior to a backtracking problem. Our problem is somehow similar to generating arrangements. In comparison, a backtracking algorithm would consume a large amount of time and resources or even stop with the current technology at a relatively low value of n (approximately 20-30 tests within the battery).

VI. Conclusion

This algorithm is useful in evaluation issues, but its applications could be extended in problems which can be structured in similar ways (e.g., in the agricultural domain, in mathematics etc.), because of the usage of combinatorics notions. Furthermore, the method can be extended for generating questions within a test (optimized sequences of questions), an issue presented in next papers. The method is useful for selecting specific tests conditioned by some constrains given by the user and the algorithm is designed to give solutions in a reasonable amount of time, within a reasonable precise range. A future work would be considered the implementation of a real-time web-based or offline application which can show in a graphical way the results of this algorithm.

References

[1] E. MacLellan, "Assessment for Learning: The differing perceptions of tutors and students," *Assessment & Evaluation in Higher Education*, vol. 26, no. 4, pp. 307-318, 2001.

[2] D. Boud and N. Falchikov, *Rethinking Assessment in Higher Education: Learning for the Longer Term*, Routledge Publishing, 2007.

[3] K. Struyven, F. Dochy, and S. Janssens, "Students' Perceptions about New Modes of Assessment in Higher Education: A Review," *Optimising New Modes of Assessment: In Search of Qualities and Standards*, vol. 1 of the series Innovation and Change in Professional Education, pp. 171-223, 2005.

[4] D. Nijloveanu, N. Bold, and A.-C. Bold, "A hierarchical model of test generation within a battery of tests," *International Conference on Virtual Learning*, pp. 147-153, 2015.

[5] E. Popescu, "Adaptation Provisioning with respect to Learning Styles in a Web-Based Educational System: An Experimental Study," *Journal of Computer Assisted Learning*, vol. 26, no. 4, pp. 243-257, 2010.

[6] V. Ştefănescu, C. Ştefănescu, and O. Roşu Stoican, "The influence of using ICT on the quality of learning", *International Conference on Virtual Learning, ICVL*, pp. 169-172, 2015.

[7] C. Holotescu, "A conceptual model for Open Learning Environments", International Conference on Virtual Learning – ICVL, pp. 54- 61, 2015

[8] C. Baron, A. Şerb, N. M. Iacob, and C. L. Defta, "IT Infrastructure Model Used for Implementing an E-learning Platform Based on Distributed Databases," *Quality-Access to Success Journal*, vol. 15, no. 140, pp. 195-201, 2014.

[9] C. L Defta, A. Şerb, N. M. Iacob, and C. Baron, "Threats analysis for E-learning platforms," *Knowledge Horizons - Economics*, vol. 6, no. 1, pp. 132–135, 2014.

[10] R. Clariana and P. Wallace, "Paper–based versus computer–based assessment: key factors associated with the test mode effect," *British Journal of Educational Technology*, vol. 33, no. 5, pp. 593–602, 2002.

[11] M. Graff, "Cognitive Style and Attitudes Towards Using Online Learning and Assessment Methods," *Electronic Journal of e-Learning*, vol. 1, no. 1, pp. 21-28, 2003.

[12] J. Gaytan and B. C. McEwen, "Effective Online Instructional and Assessment Strategies," *American Journal of Distance Education*, vol. 21, no. 3, pp. 117-132, 2007.

[13] M. J. Cox and G. Marshall, "Effects of ICT: Do we know what we should know?," *Education and Information Technologies*, vol. 12, no. 2, pp. 59-70, 2007.

[14] C. Groşan and M. Oltean, "Algoritmi Evolutivi," *Ginfo*, vol. 8, pp. 30-36, 2011.

[15] L. G. Caldas and L. K. Norford, "A design optimization tool based on a genetic algorithm," *Automation in Construction*, ACADIA 1999, vol. 11, no. 2, pp. 173–184, 2002.

[16] S. Rahmani, S. M. Mousavi, and M. J. Kamali, "Modeling of road-traffic noise with the use of genetic algorithm," *Applied Soft Computing*, vol. 11, no. 1, pp. 1008–1013, 2011.

[17] S. Darby, T. V. Mortimer-Jones, R. L. Johnston, and C. Roberts, "Theoretical study of Cu–Au nanoalloy clusters using a genetic algorithm," *Journal of Chemical Physics*, vol. 116, no. 4, 2002.

[18] D. Popescu Anastasiu and D. Radulescu, "Monitoring of irrigation systems using genetic algorithms," *ICMSAO*, pp. 1-4, 2015.

ISSN 2395-8618

[19] D. Popescu Anastasiu and D. Radulescu, "Approximately Similarity Measurement of Web Sites," *ICONIP*, *Neural Information Processing*, Proceedings, LNCS, Springer, 9-12, 2015.

[20] D. Popescu Anastasiu and I. A. Popescu, "Model of determination of coverings with web pages for a website, *International Conference on Virtual Learning*, pp. 279-283, 2015.

[21] H.-S. Kim and S.-B. Cho, "Application of interactive genetic algorithm to fashion design," *Engineering Applications of Artificial Intelligence*, vol. 13, no. 6, pp. 635–644, 2000.

ISSN 2395-8618

IN-DEDUCTIVE and DAG-Tree Approaches for Large-Scale Extreme Multi-label Hierarchical Text Classification

Mohammad Golam Sohrab, Makoto Miwa, and Yutaka Sasaki

***Abstract*—This paper presents a large-scale extreme multi-label hierarchical text classification method that employs a large-scale hierarchical inductive learning and deductive classification (IN-DEDUCTIVE) approach using different efficient classifiers, and a DAG-Tree that refines the given hierarchy by eliminating nodes and edges to generate a new hierarchy. We evaluate our method on the standard hierarchical text classification datasets prepared for the PASCAL Challenge on Large-Scale Hierarchical Text Classification (LSHTC). We compare several classification algorithms on LSHTC including DCD-SVM, SVM^{perf}, Pegasos, SGD-SVM, and Passive Aggressive, etc. Experimental results show that IN-DEDUCTIVE approach based systems with DCD-SVM, SGD-SVM, and Pegasos are promising and outperformed other learners as well as the top systems participated in the LSHTC3 challenge on Wikipedia medium dataset. Furthermore, DAG-Tree based hierarchy is effective especially for very large datasets since DAG-Tree exponentially reduce the amount of computation necessary for classification. Our system with IN-DEDUCIVE and DAG-Tree approaches outperformed the top systems participated in the LSHTC4 challenge on Wikipedia large dataset.**

***Index Terms*—Hierarchical text classification, multi-label learning, indexing, extreme classification, tree-structured class hierarchy, DAG- or DG-structured class hierarchy.**

I. Introduction

STATISTICAL Natural Language Processing (NLP) is now facing various "Big Data" challenges. In machine learning (ML)-based text classification (TC), the current front-line of "Big Data" deals with millions of training and test documents as well as hundreds of thousands, or even millions of labels. Although strong ML methods such as Support Vector Machines (SVMs) [1], [2], [3] have been successfully applied to TC, such large-scale datasets are often handled with a light-weight classifiers, such as k-nearest neighbors [4], or by information retrieval-based approaches [5].

In general, ML-based TC can be categorized into two classification tasks: a flat classification (FC) by referring to standard binary or multi-class classification problems and a hierarchical classification (HC)– typically a tree, a directed acyclic graph (DAG), or a directed graph (DG) is incorporated, where the classes to be predicted are organized into a class hierarchy. A very large amount of research in TC, data mining (DM), and related researches have focused on FC problems. In contrast, many important real-world classification problems are naturally cast as HC problems. In large-scale hierarchical text classification (LSHTC) tasks, the size of data is too large to analyze the suitable classifiers. Therefore, it is still an open and more challenging problem to design a model that classifies large-scale documents into large-scale hierarchically-structured categories that correspond to classes accurately and efficiently. The benefit of hierarchical text classification (HTC) approach is the efficiency. In the training stage, deeper in the hierarchy the category is located, less data need to be handled by classifiers on average. Because of this, total training time can be drastically reduced. In the classification stage, the complexity to decide a category as the assignment for a sample will be $O(\log n)$ with n leaf categories.

In this direction, this paper present an approach that refines the DG and generates a DAG-Tree hierarchy where DAG and tree are incorporated together, in order not only to drastically reduce training and test time but also to significantly improve the classification performance; especially when the hierarchy is too large and the training data for each class is sparse. We built an accurate and efficient LSHTC system and applied it to Wikipedia medium dataset (WMD) and Wikipedia large dataset (WLD), which treat a typical multi-label TC problem to automatically assigning Wikipedia hierarchical categories to a document. These tasks of assigning Wikipedia categories to documents are included in the third (LSHTC3[1]) and forth (LSHTC4[2]) editions of PASCAL challenge.

The remainder of the study is organized as follows. Section 2 describes the base ML algorithms. In Section 3, we elaborate the proposed inductive learning and deductive classification approaches for hierarchical learning. Section 4 presents a hierarchy refinement approach for very large-scale hierarchical classification. Section 5 shows the experiment results with discussions. Finally, we conclude the study in Section 6.

Manuscript received on February 09, 2016, accepted for publication on June 16, 2016, published on October 30, 2016.

Mohammad Golam Sohrab (corresponding author), Makoto Miwa, and Yutaka Sasaki are with the Faculty of Engineering, Toyota Technological Institute, 2-12-1 Hisakata Tempaku-ku Nagoya 468-8511, Japan (e-mails: {sohrab, makoto-miwa, yutaka.sasaki}@toyota-ti.ac.jp).

[1]http://lshtc.iit.demokritos.gr/LSHTC3_CALL
[2]http://lshtc.iit.demokritos.gr/

ISSN 2395-8618

II. Base ML Algorithms

The section briefly describes the efficient base learners: SGD-SVM, Pegasos, and DCD-SVM.

A. SGD-SVM

Stochastic Gradient Descent SVM (SGD-SVM) [6] is an incremental training algorithm for SVMs. It randomly selects a sample and adjusts the weight vector. Given a loss function $\ell(\vec{w};(x,y))$, SGD-SVM solves the following primary SVM optimization problem directly:

$$\min_{w} \frac{\lambda}{2}\|\vec{w}\| + \frac{1}{N}\sum_{(\vec{x},y)\in D} \ell(\vec{w};(\vec{x},y)), \tag{1}$$

where $\vec{w}$ is an weight vector, D is a set of N pairs of samples and their labels, and λ is a regularization parameter. The weight vector at a time step t is updated as:

$$\vec{w}_{t+1} = \vec{w}_t - \eta_t S^{-1}(\lambda\vec{w}_t + \ell'(\vec{w}_t;(\vec{x}_{t+1},y_{t+1}))\vec{x}_{t+1}) \tag{2}$$

where S is a symmetric positive definite matrix, regarded as a pre-conditioner.

B. Pegasos

Primal Estimated sub GrAdient SOlver for SVM (Pegasos) [7] is an efficient training algorithm for SVMs. Pegasos alternates stochastic gradient decent steps and projection steps. In the projection step, the current weight vector is re-scaled to fit the L_2-ball of radius $1/\sqrt{\lambda}$. For a randomly chosen sample $(\vec{x}_t, y_t)$ and a weight vector $\vec{w}_t$ at a time step t, if $y_t\vec{w}_t\cdot\vec{x}_t < 1$, Pegasos updates the weight vector as follows:

$$\vec{w}_{t+\frac{1}{2}} = (1-\eta_t\lambda)\vec{w}_t + \eta_t y_t \vec{x}_t. \tag{3}$$

$$\vec{w}_{t+1} = min\left(1, \frac{\frac{1}{\lambda}}{\|\vec{w}_{t+\frac{1}{2}}\|}\right)\vec{w}_{t+\frac{1}{2}}. \tag{4}$$

C. DCD-SVM

Dual coordinate decent support vector machine (DCD-SVM) [8] randomly selects a weight vector $\vec{w}$ and updates the weight vector as:

$$\vec{w} \leftarrow \vec{w} + (\alpha_i - \alpha_i')y_i\vec{x}_i, \tag{5}$$

where α_i' is the current value and α_i is the target value. The optimization process starts from an initial point $\vec{\alpha} \in \mathbb{R}^l$ and generates a sequence of vectors $\{\vec{\alpha}^k\}_k^\infty$. We refer to the process from $\vec{\alpha}^k$ to $\vec{\alpha}^{k+1}$ as an outer iteration. In each outer iteration, we have l inner iterations, so that sequentially updates $\alpha_1, \alpha_2, ..., \alpha_l$. In updating $\vec{\alpha}^{k,i}$ to $\vec{\alpha}^{k,i+1}$, the process must find the optimal solution as:

$$\alpha_i^{k,i+1} = min\left(max\left(\alpha_i^{k,i} - \frac{\nabla_i f\left(\vec{\alpha}^{k,i}\right)}{\vec{x}_i^T\cdot\vec{x}_i}, 0\right), C\right), \tag{6}$$

where $C > 0$ is a penalty parameter and set to 0.5 based on our pilot study. $\nabla_i f$ is the i-th component of the gradient ∇f, and $\nabla_i f\left(\vec{\alpha}^{k,i}\right)$ is set as:

$$\nabla_i f\left(\vec{\alpha}\right) = y_i\vec{w}^T\cdot\vec{x}_i - 1. \tag{7}$$

The process move to index $i+1$ with updating $\alpha_i^{k,i}$, if and only if the projected gradient $\nabla_i^P f\left(\vec{\alpha}^{k,i}\right) \neq 0$ and satisfy the following conditions,

$$\nabla_i^P f\left(\vec{\alpha}\right) = \begin{cases} \nabla_i f\left(\vec{\alpha}\right) & \text{if } 0 < \alpha_i < C, \\ \min\left(0, \nabla_i f\left(\vec{\alpha}\right)\right) & \text{if } \alpha_i = 0, \\ \max\left(0, \nabla_i f\left(\vec{\alpha}\right)\right) & \text{if } \alpha_i = C. \end{cases} \tag{8}$$

III. IN-DEDUCTIVE Approach

IN-DEDUCTIVE (inductive learning and deductive classification) is a hierarchical learning and classification approach for classifying large-scale documents into a large-scale hierarchically structured categories (category hierarchy) accurately and efficiently. The IN-DEDUCTIVE approach follows the bottom-up propagation with edge-based training, top-down classification approach with global adjustments, and global pruning.

A. Inductive Learning on Category Hierarchy

The inductive learning induces a set of observed instances or samples from specific bottom categories to general top categories in the category hierarchy.

1) Bottom-up Propagation: Since only leaf categories are assigned to data, first we propagate training samples from the leaf level to the root in the category hierarchy. Fig. 1 illustrates propagation of documents in a hierarchy consisting of six categories A-F. In this figure, sample x_1 is assigned to categories D and E, x_2 to D, and x_3 to F. Let us look at the case of x_1 assigned to E. x_1 of E is propagated to both categories B and C. Then, x_1 of B is propagated to A. When x_1 is propagated from C to A afterwards, to avoid redundant propagation, the propagation of x_1 (originally from E via C) terminates at A, even if A had a parent category. To perform the propagation of the sample, we employ a recursive algorithm. Steps 1-13 for bottom-up propagation in Algorithm 1 are described in pseudo-code. Samples are propagated in a DAG in the bottom-up manner.

2) Edge-based Training: Based on the propagation of training samples in the hierarchy, we train classifiers for each edge of the hierarchy to estimate how strong the samples of the parent node are related to the child nodes. Each edge is coupled with a binary classifier using the one-against-the-rest approach. In Fig. 2 at node B, x_1 and x_2 are assigned to node B during the bottom-up propagation. Since edge-based learning is in concern, the model M_{BD} is trained in the hierarchy as to classify both x_1 and x_2 to D; whereas the model M_{BE} is trained as to classify x_1 to E but not x_2 to E. M_{BD} is trained without negatives. Training models on local branches is beneficial in restricting the number of samples. It is also

ISSN 2395-8618

effective to reduce positive-negative data imbalance. Another benefit of this edge-oriented classification is that a classifier can capture local characteristics of data distribution. Naturally, x_1 classified into E from node B and x_1 from node C should need different considerations, since the former is classification based on x_1 and x_2 and the later is based on x_1 and x_3, and thus M_{BE} and M_{CE} will be different models. Edge-based training is described in the steps 14-34 in Algorithm 1.

B. Deductive Classification on Category Hierarchy

The deductive classification deduces a set of unlabeled samples from general top categories to more specific bottom categories in the hierarchy.

1) Efficient Top-down Classification: Fig. 3 illustrates top-down classification of a test sample $\vec{x}$. First, $\vec{x}$ is classified to B and C, based on the decision by $M_{AB}(\vec{x})$ and $M_{AC}(\vec{x})$, respectively. The decision is made by:

$$G_{pc}(\vec{x}) = \vec{w}_{pc} \cdot \vec{x} + b_{pc}. \tag{9}$$

To adjust the effect of positive-negative sample imbalance, we set a bias β. When $G_{pc}(\vec{x}) > \beta$, $\vec{x}$ is classified from parent category p to child category c. When both $G_{AB}(\vec{x}) > \beta$ and $G_{AC}(\vec{x}) > \beta$ are satisfied, $\vec{x}$ is classified into both B and C. Note that the standard bias term b_{pc} is automatically tuned for each edge in the training stage. Usually, the dot product can be calculated efficiently with the dot product of sparse vectors [9]. For the first calculation of the dot product in the classification stage, the dot product can be calculated as follows:

$$\vec{w} \cdot \vec{x} = \sum_{i:1..size[x_{index}]} w[x_{index}[i]] * x_{value}[i], \tag{10}$$

where w is an weight vector array and x_{index} and x_{value} represent a sparse vector of sample x. Top-down classification is described in the steps 35-53 Algorithm 1. The classification procedure is initiated with TOP-DOWN-CLASSIFY(x,root,1).

C. Global Pruning

After the classification of a test sample x throughout the hierarchy, we prune unlikely classes for x. We define a confidence score and set the global threshold θ for it. When x reaches a leaf node n, the confidence score $c_\alpha(x, n)$ is calculated as follows:

$$c_\alpha(x, n) = \prod_{(n_1,n_2)\in E} \sigma_\alpha(G_{n_1 n_2}(x))), \tag{11}$$

where E is a set of edges that x has followed in the path from the root to the leaf n. The output value of a classifier is converted to $[0, 1]$ range by:

$$\sigma_\alpha(x) = \frac{1}{1 + \exp(-\alpha x)}, \tag{12}$$

α is set to 2 from our pilot study. When there are multiple nodes assigned to x, if $c(x, n) < \theta$, the assignment of x to n is removed. Fig. 4 illustrates the global pruning. In Fig. 4, x is classified into D and F. Here the confidence scores of x in D and F is 0.21 and 0.09 respectively. When θ = 0.3, both of the confidence scores are below the threshold. However, we need at least one category for every test sample. Therefore, only assignment of x to F is removed.

IV. Hierarchy Refinement: DAG-Tree

Since IN-DEDUCTIVE is a edge-oriented approach that increases the computational cost of training and test by building hundreds of thousands, or even millions of classification models. To reduce the computational cost in training and test as well as to improve the classification performances, we introduce a hierarchy refinement approach and generate a DAG- and tree-based hierarchy; especially from very large directed graph which contains cycles and multiples roots in the hierarchy.

DAG-Tree incorporates DAG and tree substructures together into a same graph to refine the given hierarchy. In the DAG-Tree approach, lead nodes contain multiple parents where from intermediate nodes to root nodes in the hierarchy contain single parent in the hierarchy. To generate the DAG-Tree hierarchy, we split the given hierarchy into three indexing system. In the LEAF-INDEX, all the leaf categories (leaf contains only parent) in the hierarchy are associated. In the INTER-INDEX, all the intermediate categories (Intermediate nodes contain both parent and child) in the hierarchy are associated. Finally, in the ROOT-INDEX, root categories (Root contains only children) in the hierarchy are associated. From leaf level to intermediate level in the DAG-Tree hierarchy, a leaf node can have more than one parents. From intermediate levels to top level, a node contains only one parent in the DAG-Tree hierarchy.

To handle large data using hierarchy, we introduce a bottom-up based DAG-Tree. The primary motivation of creating DAG-Tree is to remove all cycles by ignoring nodes and edges that have been already visited in the paths from leaf to root in the intermediate levels. Any parent-child relations that would lead to cycle are omitted. Fig. 6 shows an example DAG-Tree categories hierarchy.

DAG-Tree is a visited-paths based approach. In the visited-paths, DAG-Tree keeps record all the parents list of a certain leaf or intermediate node. If any parent of a certain leaf or intermediate node appears in the visited-paths list, DAG-Tree immediately stores the last parent record by terminating the process and accesses the next parent of a certain leaf in the hierarchy. In Figures 5 and 6, the bottom and top gray circles indicate the leaf and root nodes respectively; as well as circles in the dotted rectangle indicate the intermediate nodes. First we generate a leaf to root paths using bottom-up manner and invoke the paths as a reference-path for every parents of a certain leaf to reach root node. The procedures in Fig. 7 show a DAG-Tree generation from the hierarchy in Fig. 5.

Algorithm 2 shows the visited-paths based DAG-Tree generation algorithm. Compare to Fig. 5, Fig. 6 shows many nodes and edges, including intermediate node G and root node

ISSN 2395-8618

Algorithm 1. IN–DEDUCTIVE Approach for LSHTC

procedure BOTTOM-UP-PROPAGATE(NODE, SAMPLES)
  if sample is already assigned to node **then**
    return;
  else
    Assign sample to node;
  end if
  if node is the root **then**
    return;
  end if
  for all p in the parents of node **do**
    BOTTOM-UP-PROPAGATE(p, sample)
  end for
end procedure
procedure TRAIN(NODE)
  if sample is already explored **then**
    return;
  end if
  if node is a leaf **then**
    return;
  end if
  Let X be the set of samples assigned to node;
  Let Y be the set of labels for X
  for all n in the children of node **do**
    for all x_i in X **do**
      if x_i is assigned to n **then**
        $y_i \leftarrow +1$;
      else
        $y_i \leftarrow -1$;
      end if
    end for
    Train classification model $M_{node,n}$ on (X,Y)
    TRAIN(n)
  end for
end procedure
procedure TOP-DOWN-CLASSIFY(x, $node$, $conf$)
  if $node$ is a leaf node **then**
    assign x to node;
  end if
  if node is a leaf **then**
    c_α(x,$node$) $\leftarrow$ max($conf$,old(c_α(x,$node$));
  else
    c_α(x,$node$) $\leftarrow$ $conf$;
  end if
  for all n in the children of $node$ **do**
    if $G_{node,n} > \beta$ **then**
      TOP-DOWN-CLASSIFY($x, n, conf * \sigma(G_{node,n}(x))$);
    end if
  end for
  if None of child nodes satisfies $M_{node}, n(x) > \beta$ **then**
    n' = $\underset{n}{\text{argmax}} G_{node,n}(x)$;
    TOP-DOWN-CLASSIFY($x, n', conf * \sigma(G_{node,n'}(x))$);
  end if
end procedure

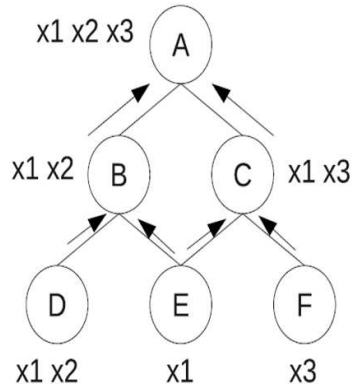

Fig. 1. Bottom–up propagation of training data.

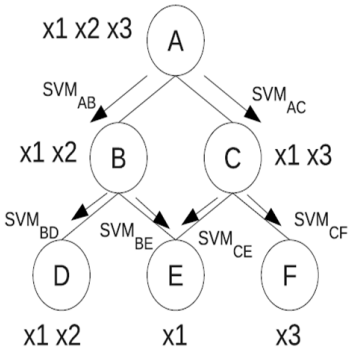

Fig. 2. Edge–based training.

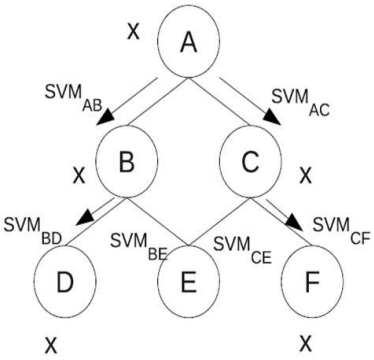

Fig. 3. Top–down classification.

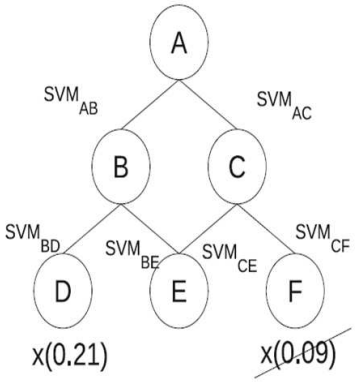

Fig. 4. Global pruning.

ISSN 2395-8618

C, are eliminated during the DAG-Tree generation process. We then extend the intermediate nodes by including the root nodes and connect all the root nodes to a single root Z.

V. Experimental Settings

In this section, we provide empirical evidence for the effectiveness of our proposed IN-DEDUCTIVE and DAG-Tree approaches.

A. Base ML Algorithms

We employ sofia-ml[3] for the experiments with Pagasos, SGD-SVM, Passive Aggressive (PA) [10], Relaxed Online Margin Algorithm (ROMMA) [11], and Logistic regression with Pegasos projection (logreg-pegasos). For Pegasos and SGD-SVM, we use the SVM regularization parameter C where $\lambda = \frac{1}{CN}$ and the number of iteration is set to $\max(10000, 100N)$. Moreover, SVM^{perf} [12], [13] is adopted, and the SVM parameters C for SVM^{perf} is given as $C^{perf} = NC/100$, where N is the number of samples of each edge. Note that C^{perf} varies in each node during the top-down training since C^{perf} depends on N.

B. Term Weighting Approaches

We compare several term weighting methods: term frequency (TF), TF.IDF, and four class-oriented indexing-based weighting methods.

TF is a very simple, conventional, and baseline term weighting method in the information retrieval task and it is defined as:

$$W_{TF}(t_i, d) = tf_{(t_i,d)}, \tag{13}$$

where $tf(t_i, d)$ is the number of occurrences of term t_i in document d.

The common document-indexing-based TF.IDF is defined as:

$$W_{TF.IDF}(t_i, d) = tf_{(t_i,d)} \times \left(1 + \log \frac{D}{\#(t_i)}\right), \tag{14}$$

where D denotes the total number of documents in the training corpus, $\#(t_i)$ is the number of documents in the training corpus in which term t_i occurs at least once, $D/\#(t_i)$ is the inverse document frequency (IDF) of term t_i.

As for the class-oriented indexing-based methods [14], [15], we employed four following methods defined as:

$$W_{TF.ICF}(t_i, d, c_k) = tf_{(t_i,d)} \times \left(1 + \log \frac{C}{c(t_i)}\right), \tag{15}$$

$$W_{TF.ICS_\delta F}(t_i, d, c_k) = tf_{(t_i,d)} \times \left(1 + \log \frac{C}{CS_\delta(t_i)}\right), \tag{16}$$

$$W_{TF.IDF.ICF}(t_i, d, c_k) = tf_{(t_i,d)} \times \left(1 + \log \frac{D}{\#(t_i)}\right) \times \left(1 + \log \frac{C}{c(t_i)}\right), and \tag{17}$$

$$W_{TF.IDF.ICS_\delta F}(t_i, d, c_k) = tf_{(t_i,d)} \times \left(1 + \log \frac{D}{\#(t_i)}\right) \times \left(1 + \log \frac{C}{CS_\delta(t_i)}\right), \tag{18}$$

where C denotes the total number of predefined categories in the training corpus, $c(t_i)$ is the number of categories in the training corpus in which term t_i occurs at least once, $\frac{C}{c(t_i)}$ is the ICF of the term t_i, and $\frac{C}{CS_\delta(t_i)}$ is the (ICS$_\delta$F) of term t_i. Please refer to [14], [16] for more details.

C. Datasets

To evaluate the performance of our proposed IN-DEDUCTIVE and DAG-Tree approaches, we compare our results with WMD and WLD which considering two standard datasets for LSHTC.

1) Wikipedia Medium Dataset: The Dataset[4] consists of 456,866 training documents with 346,299 distinct features and 81,262 test documents with 132,296 distinct features. It contains 36,504 leaf categories and 50,312 categories in the hierarchy with maximum depth 12. The number of edges in the hierarchy are 65,333. The category hierarchies of WMD is in the form of DAG.

2) Wikipedia Large Dataset: The Dataset[5] consists of 2,365,436 training data with 1,617,899 distinct features and 452,167 test data with 627,935 distinct features. It contains 325,055 leaf categories and 478,020 categories in the hierarchy with maximum depth 15. The number of edges in the hierarchy are 863,261. The category hierarchies of WLD are in the form of DG.

D. Performance Measures

We employ official LSHTC3 and LSHTC4 evaluation metrics [17]. Given documents D, correct labels Y_i, and predicted labels Z_i, the metrics are as follows:

- Accuracy(Acc): $1/|D| \sum_{i \in D} |Y_i \cap Z_i| / (|Y_i \cup Z_i|)$
- Example-based F1 measure (EBF): $1/|D| \sum_{i \in D} 2|Y_i \cap Z_i| / (|Y_i| + |Z_i|)$
- Label-based Macro-average F1 measure (LBMaF): Standard multi-label Macro-F1 score
- Label-based Micro-average F1 measure (LBMiF): Standard multi-label Micro-F1 score
- Hierarchical F1 measure (HF): The example-based F1-measure counting ancestors of true and predicted categories

We evaluated our systems on LSHTC evaluation site[6] because the gold standard labels for the test data of WMD and WLD are not publicly available.

[3] http://code.google.com/p/sofia-ml/

[4] http://lshtc.iit.demokritos.gr/LSHTC3_DATASETS

[5] http://lshtc.iit.demokritos.gr/LSHTC4_GUIDELINES

[6] http://lshtc.iit.demokritos.gr/

ISSN 2395-8618

Algorithm 2. Bottom–up DG to DAG-Tree Generation

```
1: procedure DAG–TREE(LEAF-INDEX, INTER-INDEX, ROOT-INDEX)
2:     while LEAF-INDEX.leaf is not empty do
3:         parentLeaf ← leaf.parentList
4:         LEAF–ROOT(parentLeaf)
5:     end while
6:     return path
7: end procedure
8: procedure LEAF–ROOT(PINDEX)
9:     while pIndex.curParent is not empty do
10:        if curParent is in ROOT-INDEX then
11:            VISITED–PATH(leaf, curParent)
12:        else
13:            childInter ← curParent
14:            INTER–NODES(childInter)
15:        end if
16:    end while
17: end procedure
18: procedure VISITED–PATH(leaf, p)
19:    path.leaf.pathList ← p
20:    if all parents of a leaf are explored then
21:        goto step2: for next leaf
22:    else
23:        explore parent of LEAF- or INTER-INDEX
24:    end if
25: end procedure
26: procedure INTER–NODES(inter)
27:    while INTER-INDEX.inter is not empty do
28:        parentInter ← inter.parentList
29:        LEAF–ROOT(parentInter)
30:    end while
31: end procedure
```

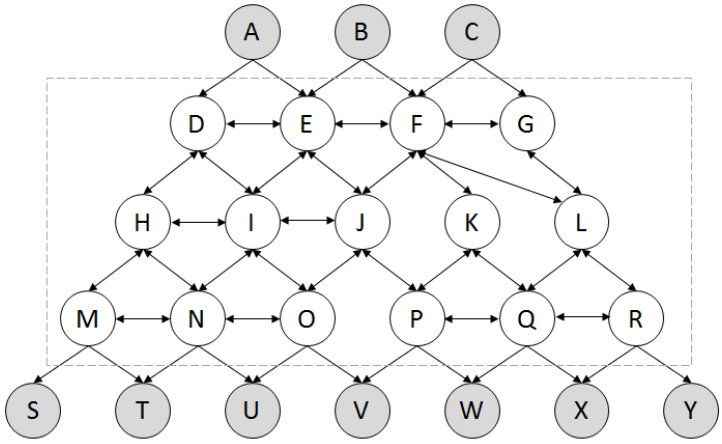

Fig. 5. Directed Graph (DG).

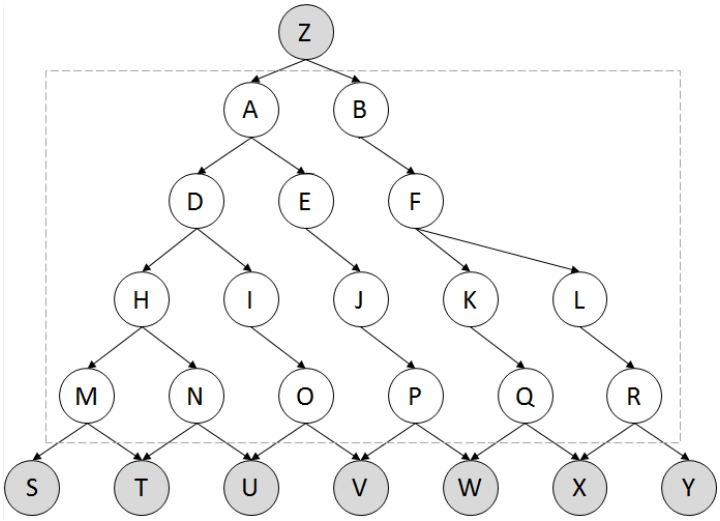

Fig. 6. DAG-Tree.

E. Experimental Environments

We assessed the training and classification time using a single Xeon 3.0GHz core with 96GB memory for WMD and 396GB memory for WLD. The feature values in the LSHTC3 and LSHTC4 datasets represent the number of occurrences of each unigram. From Eqns. 13–18, we scaled the feature value with the function $v/(v+1)$ where v is the original feature value.

VI. Results and Discussion

In this section, we provide empirical evidence of the INDEDUCTIVE approach and the effectiveness of the DAG-Tree approach in very large-scale datasets.

A. DG to DAG-Tree using Wikipedia Large Dataset

The hierarchy of WLD is a directed graph (DG), where a node can have more than one parents. It contains cycles, where a set of nodes are connected by edges and the cycles only appear in the intermediate or hidden nodes. Fig. 5 shows an example of DG, where the double arrow between two nodes, is the parent of one another. In the WLD, 10,246 cyclic nodes appear in the intermediate levels with a maximum depth of 13. It contains multiple roots with 11,405 in the hierarchy.

Using this DAG-Tree approach, we refine the hierarchy by reducing the edges, intermediate nodes, root nodes, and cyclic nodes from 863,261 to 683,359, 141,559 to 140,703, 11,405 to 10,902, and 10,246 to 0 respectively.

B. LSHTC Evaluation

Table I shows the results with Pegasos on WMD. We showed the results with $\beta \in (0.0, -0.5)$ and varied $\theta \in (0.00, 0.25, 0.27, 0.32)$. $\beta = -0.5$ means that data classified into negative side to some extent are passed to the child node. This means that some incorrect assignments are kept in the candidate sets. However, most of the incorrect classification are removed after-ward during the global pruning stage. SVM hyper-parameter C has been set to 0.5 based on our pilot study. Table II shows the scores of different weighting approaches with DCD-SVM. As for DCD-SVM results on WMD in

ISSN 2395-8618

1: S→M→H→D→A ▷ First Reference Visited-Paths from leaf to root
2: T→M ▷ M parent of T, is in the Visited-paths list in step 1:
3: T→N→H ▷ H parent of N, is in the Visited-paths list in step 1:
4: U→N ▷ N parent of U, is in the Visited-paths list in step 3:
5: U→O→I→D ▷ D parent of I, is in the Visited-paths list in step 1:
6: V→O ▷ O parent of V, is in the Visited-paths list in step 5:
7: V→P→J→E→A ▷ A parent of E, is in the Visited-paths list in step 1:
7: W→P ▷ P parent of W, is in the Visited-paths list in step 7:
8: W→Q→K→F→B ▷ B parent of F, is a root
9: X→Q ▷ Q parent of X, is in the Visited-paths list in step 8:
10: X→R→L→F ▷ F parent of L, is in the Visited-paths list in step 8:
11: Y→R ▷ R parent of Y, is in the Visited-paths list in step 10:

Fig. 7. Procedures to generate DAG-Tree hierarchy with a visited-paths list

TABLE I
EXPERIMENTAL RESULTS WITH PEGASOS ON WMD

C	α	β	θ	Acc	EBF	LBMaF	LBMiF	HF
0.5	2	0.0	0.00	0.3955	0.4585	0.2753	0.4393	0.6633
0.5	2	0.0	0.25	0.4334	0.4840	0.2666	0.4870	0.7015
0.5	2	0.0	0.27	0.4335	0.4834	0.2633	0.4870	0.7028
0.5	2	0.0	0.32	0.4328	0.4816	0.2552	0.4853	0.7014
0.5	2	–0.5	0.00	0.2966	0.3749	0.2542	0.2772	0.5469
0.5	2	–0.5	0.25	0.4388	**0.4958**	**0.2832**	0.4951	0.7058
0.5	2	–0.5	0.27	0.4406	**0.4958**	0.2773	0.4966	0.7069
0.5	2	–0.5	0.32	**0.4423**	0.4948	0.2669	0.4966	**0.7076**

TABLE II
DIFFERENT WEIGHTING APPROACHES WITH DCD-SVM ON WMD

Weighting Approach	β	θ	Acc	EBF	LBMaF	LBMiF	HF
IN-DEDUCTIVE + TF	–0.5	0.39	**0.4452**	**0.4968**	**0.2664**	**0.4978**	**0.7086**
IN-DEDUCTIVE + TF.IDF	–0.5	0.42	0.4284	0.4764	0.2537	0.4800	0.6943
IN-DEDUCTIVE + TF.ICF	–0.5	0.42	0.4346	0.4831	0.2592	0.4863	0.6984
IN-DEDUCTIVE + TF.IDF.ICF	–0.5	0.42	0.4219	0.4695	0.2481	0.4733	0.6900
IN-DEDUCTIVE + TF.ICS$_\delta$F	–0.5	0.42	0.4297	0.4779	0.2544	0.4812	0.6953
IN-DEDUCTIVE + TF.IDF.ICS$_\delta$F	–0.5	0.42	0.4221	0.4697	0.2481	0.4735	0.6899

Table III, when β = -0.5 and θ = 0.39, we obtained the best accuracy 0.4452. Since the local weight TF in Table II outperformed other weighting approaches, only this weighting approach is taken into account for WLD. In addition, the parameter β = -0.5 performed consistently better in different learning algorithms, thus β = -0.5 is taken into account for the WLD. Table V shows the results with Pegasos on the WLD, where we obtained the best result 0.3496 using our system with DAG-Tree hierarchy. The result shows that our system using DAG-Tree hierarchy outperformed with the given hierarchy.

Tables IV and VI, summarizes our results with compare to the top four systems using WMD and WLD respectively. The result shows that the IN-DEDUCTIVE approach based system outperformed the other systems participated in the LSHTC3 challenge as well as in LSHTC4 challenge. Table VII illustrates the training and test time spent for the WMD and WLD. In Table II, Table III, Table V, Table VI, and Table VII we use $C = 0.5$ and $\alpha = 2$.

C. Discussion

Ioannou [20] summarizes thresholding methods in multi-label classification. Basically bias β is a threshold that adjusts PCut [21]. Note that the training phase automatically set bias b of the decision function $G_{pc}(x) = \vec{w}_{pc}\vec{x} + b_{pc}$. Setting β means the classification threshold adjustment, $i.e., G_{pc}(x) = \vec{w}_{pc}\vec{x} + b_{pc} > \beta$, where p and c are parent and child categories, respectively. It is noticeable in Table II that the local term weighting approach TF outperformed other weighting approaches that incorporate global weightings into local weights for LSHTC.

ISSN 2395-8618

TABLE III
COMPARISON OF EFFICIENT ML METHODS ON WMD

Learning Algorithm	β	θ	Acc	EBF	LBMaF	LBMiF	HF
IN-DEDUCTIVE + DCD-SVM	–0.5	0.39	**0.4452**	**0.4968**	0.2664	**0.4978**	**0.7086**
IN-DEDUCTIVE + Pegasos	–0.5	0.32	0.4423	0.4948	**0.2669**	0.4966	0.7076
IN-DEDUCTIVE + SGD-SVM	–0.5	0.32	0.4419	0.4938	0.2641	0.4957	0.7072
IN-DEDUCTIVE + SVM^{perf}	–0.5	0.32	0.4405	0.4919	0.2623	0.4947	0.7071
IN-DEDUCTIVE + PA	–0.5	0.49	0.4005	0.4512	0.2550	0.4527	0.6673
IN-DEDUCTIVE + ROMMA	–0.5	0.15	0.3827	0.4324	0.2296	0.4362	0.5610
IN-DEDUCTIVE + logreg	–0.3	0.14	0.3690	0.4235	0.1544	0.4271	0.6688
IN-DEDUCTIVE + logreg-pegasos	–0.5	0.14	0.3689	0.4255	0.1644	0.4296	0.6682

TABLE IV
EXPERIMENTAL RESULTS WITH PEGASOS ON WLD

Name	C	α	β	θ	Acc	EBF	LBMaF	LBMiF	HF
IN-DEDUCTIVE + DG	0.5	2	−0.5	0.37	0.3183	0.3861	0.1918	0.3641	0.4291
IN-DEDUCTIVE + DAG-Tree	0.5	2	−0.5	0.38	0.3495	**0.4239**	**0.1968**	0.3946	**0.4790**
	0.5	2	−0.5	0.39	**0.3496**	0.4235	0.1937	0.3951	0.4773
	0.5	2	−0.5	0.40	0.3494	0.4229	0.1907	**0.3952**	0.4751

TABLE V
COMPARISON WITH TOP FOUR LSHTC3 PARTICIPANTS ON WMD

Name	Acc	EBF	LBMaF	LBMiF	HF
IN-DEDUCTIVE + DCD-SVM	**0.4452**	**0.4968**	0.2664	**0.4978**	**0.7086**
IN-DEDUCTIVE + Pegasos	0.4423	0.4948	0.2669	0.4966	0.7076
IN-DEDUCTIVE + SGD-SVM	0.4419	0.4938	0.2641	0.4957	0.7072
IN-DEDUCTIVE + SVM^{perf}	0.4405	0.4919	0.2623	0.4947	0.7071
arthur (Ist)	0.4382	0.4937	0.2674	0.4939	0.7092
coolvegpuff (2nd)	0.4291	0.4824	0.2507	0.4779	0.6892
TTI (3rd)	0.4200	0.4771	**0.2835**	0.4725	0.6922
chrishan (4th)	0.4117	0.4768	0.2454	0.4187	0.6766

TABLE VI
COMPARISON WITH TOP FOUR LSHTC4 PARTICIPANTS ON WLD

Name	Acc	EBF	LBMaF	LBMiF	HF
IN-DEDUCTIVE + Pegasos + DAG-Tree	**0.3495**	**0.4239**	**0.1968**	**0.3946**	**0.4790**
TTI (1st)	0.3185	0.3866	0.1920	0.3644	0.4295
anttip (2nd)	0.3152	0.3682	0.1919	0.3038	0.4546
knn baseline (3rd)	0.2724	0.3472	0.1486	0.3016	0.4616
kensk8er (4th)	0.2714	0.3462	0.1519	0.3002	0.4594

TABLE VII
EFFICIENCY WITH IN-DEDUCTIVE APPROACH

Data	Training	CPU time	Test	CPU time
WMD	65,333 models		36,506 leaf categories	
	SVM^{perf}	13.85 hrs	SVM^{perf}	7.9m
	Pegasos	3.8 hrs	Pegasos	10.2m
	TTI [18]	16 hrs	dhlee [5]	12.5m
	anttip [19]	15 days	TTI [18]	10.2m
WLD	863,261 models		478,020 leaf categories	
	Pegasos + DG	16 days	Pegasos + DG	2 days
	683,359 models		478,020 leaf categories	
	Pegasos + DAG-Tree	2 days	Pegasos + DAG-Tree	18 hrs

 https://doi.org/10.17562/PB-54-8

ISSN 2395-8618

It is also noticeable that the DAG-Tree approach not only drastically decrease the computational cost but also can significantly improve the system performances. It decreases the computational cost by reducing the given hierarchy to a new one. Even though we get less data from intermediate to top nodes but the IN-DEDUCTIVE approach based system gets higher accuracy using the proposed DAG-Tree. Since we first perform the DAG-based approach to train each edge from leaf to immediate intermediate nodes in the hierarchy, the original training information remains. Moreover, from bottom to top intermediate levels we perform the Tree-based approach to train each edge in the hierarchy based on number of descendants are associate with a certain parent. Since large-scale data-set is in concern and for each outer iteration, we randomly select samples and update weight vectors from block size– referring to inner iteration, even a less information in the intermediate nodes can significantly improve the system performance. Thus, the DAG-Tree is useful to enhance the HTC for very large-scale hierarchy.

VII. Related Work

TC is a typical multi-class single- and multi-label classification problem. To efficiently solve, Platt [9] proposed a faster training of SVM using sequential minimal optimization (SMO) that breaking a very large quadratic programming optimization problem into a series of smallest possible problems as an inner loop in each outer iteration. The approach is generally 1200 and 15 times faster for linear and non-linear SVMs respectively. Studies to solve multi-class multi-label classification have been summarized in [22], in three smaller data sets with maximum labels of 27 in compare to current front-line of multi-label classification task.

There have been many studies that use local context in HTC [23], [24], [25]. Chakrabarti et al. [23] proposed a Naive-Bayes document classification system that follows hierarchy edges from the root node. Koller et al. [24] applied Bayesian networks to a hierarchical document classification. In our approach, we applied efficient learners which have shown good classification performance with fast training as well as improved the pruning method.

In LSHTC3, the arthur system [26] successfully applied meta-classifiers to the LSHTC task. Meta-classifiers can also be regarded as a sort of pruning. The system employed Liblinear, Max Entropy classifier, and SVM^{light}. The meta-classifier with SVM^{light} achieved 0.4381 on the aspect of accuracy; however relatively slow in compare to Liblinear and Max Entropy on the aspect of efficiency.

The anttip system [19] employed the ensemble of different classifiers by introducing Feature-Weighted Linear Regression. The system also used greedy pruning of the base-classifiers by ranking hypothesis that iteratively removing a classier from the ensemble in the development stage. In LSHTC, the system achieved 0.3152 over the WLD on the aspect of accuracy.

Lee [5] proposed a Multi-Stage Rocchio classification (MSRC) based on similarity between test documents and label's centroids for large-scale datasets. The system used greedy search algorithm in the predicted label set and then compare similarities between test documents and two centroid to check whether more labels are needed or not. The MSRC achieved 0.3974, 0.4326, and 0.6783 in terms of accuracy, LBMiF, and HF respectively for WMD. On the aspect of efficiency the system is much faster than baseline such as K-Nearest Neighbor when the expected number of labels per document are less.

VIII. Conclusions

The IN-DEDUCTIVE approach based system outperformed top-group systems in LSHTC3, w.r.t the most of evaluation metrics in different learning algorithms. This can be attributed to the bias adjustment $\beta = -0.5$ and post pruning. Moreover, the SVM-based systems with the IN-DEDUCTIVE and the DAG-Tree approaches also outperformed the LSHTC4's top-group systems. We believe that, to handle extreme multi-label LSHTC problems, the results will make a useful contribution as an useful performance reference. Our future work includes the development of much more efficient algorithms for large-scale IN-DEDUCTIVE approach based system in HTC.

Acknowledgment

This work has been partially supported by JSPS KAKENHI Grant Number 449001.

References

[1] C. Cortes and V. Vapnik, "Support vector networks," *Journal of Machine Learning*, vol. 20, pp. 273–297, 1995.

[2] S. Dumais, J. Platt, and D. Heckerman, "Inductive learning algorithms and representations for text categorization," in *Proceedings of the 1998 CIKM*, 1998, pp. 148–155.

[3] V. Vapnik, *The Nature of Statistical learning Theory*. Springer, 1995.

[4] X. Han, S. Li, and Z. Shen, "k-NN method for large scale hierarchical text classification for LSHTC3," in *Proceedings of the 2012 ECML/PKDD Discovery Challenge Workshop on Large-Scale Hierarchical Text Classification*, Bristol, 2012.

[5] D. H. Lee, "Multi-stage rocchio classification for large-scale multi-labeled text data," in *Proceedings of the 2012 ECML/PKDD Discovery Challenge Workshop on Large-Scale Hierarchical Text Classification*, Bristol, 2012.

[6] T. Zhang, "Solving large scale linear prediction problems using stochastic gradient decent algorithms," in *Proceedings of the the 21th Conference on Machine Learning*, 2004.

[7] S. Shlev-Shwartz, Y. Singer, and N. Srebro, "Primal estimated sub-gradient solver for SVM," in *Proceedings of the 24th International Conference on machine Learning*, 2007.

[8] C. Hsieh, K. Chang, C. Lin, S. S. Keerthi, and S. Sundarara-jan, "A dual coordinate descent method for large-scale linear SVM," in *Proceedings of the ICML-08*, 2008, pp. 408–415.

[9] J. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods: Support Vector Learning*. MIT Press, 1998.

[10] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithm," *Journal of Machine Learning Research*, vol. 7, pp. 551–585, 2006.

[11] Y. Li and P. Long, "The relaxed online maximum margin algorithm," *Journal of Machine Learning*, vol. 46, pp. 1–3, 2002.

[12] T. Joachims, "A support vector method for multivariate performance measures," in *Proceedings of the 2005 International Conference on Machine Learning*, 2005, pp. 377–384.

ISSN 2395-8618

[13] ——, "Training linear SVMs in linear time," in *Proceedings of the ACM conference on Knowledge Discovery and Data Mining*, 2006, pp. 217–226.

[14] F. Ren and M. G. Sohrab, "Class-indexing-based term weighting for automatic text classification," *Information Sciences*, vol. 236, pp. 109–125, 2013.

[15] M. G. Sohrab and F. Ren, "The effectiveness of class-space-density in high and low-dimensional vector space for text classification," in *Proceedings of the 2nd IEEE International Conference of CCIS*, China, 2012, pp. 2034–2042.

[16] M. G. Sohrab, M. Miwa, and Y. Sasaki, "Centroid-means-embedding: An approach to infusing word embeddings into features for text classification," in *Proceedings of Advance in Knowledge Discovery and Data Mining, LNCS*, vol. 9077, 2015, pp. 289–300.

[17] G. Tsoumakes, I. Katakis, and I. Vlahavas, "Random k-labelsets for multi-label classification," in *Proceeding of the Knowledge Discovery and Data Engineering*, 2010.

[18] Y. Sasaki and D. Weissenbacher, "TTI'S system for the LSHTC3 challenge," in *Proceedings of the 2012 ECML/PKDD Discovery Challenge Workshop on Large-Scale Hierarchical Text Classification*, Bristol, 2012.

[19] A. Puurula and A. Bifet, "Ensembles of sparse multinomial classifiers for scalable text classification," in *Proceedings of the 2012 ECML/PKDD Discovery Challenge Workshop on Large-Scale Hierarchical Text Classification*, Bristol, 2012.

[20] M. Ioannou, G. Sakkas, G. Tsoumakas, and L. Vlahavas, "A dual coordinate descent method for large-scale linear SVM," in *Proceedings of the 2010 IEEE International Conference on Tools with artificial Intelligence*, 2010, pp. 409–419.

[21] Y. Yang, "A study on threshold strategies for text classification," in *Proceedings of the 24th annual international ACM SIGIR conference on research and development in Information Retrieval*, NY, 2001, pp. 137–145.

[22] G. Tsoumakes and I. Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining*, vol. 3, no. 3, pp. 1–13, 2007.

[23] S. Chakrabarti, B. Dom, R. Agrawal, and P. Raghavan, "Scalable feature selection, classification and signature generation for organizing large text databases into hierarchical topic taxonomies," *International Journal on Very Large data Bases*, vol. 7, no. 3, pp. 163–178, 1998.

[24] D. Koller and M. Sahami, "Hierarchically classifying documents using very few words," in *Proceedings of the 14th Conference on Machine Learning*, 1997, pp. 170–178.

[25] A. McCallum, R. Rosenfeld, T. Mitchell, and A. Ng, "Improving text classification by shrinkage in a hierarchy of classes," in *Proceedings of the Fifteenth International Conference on Machine Learning*, 1998, pp. 359–367.

[26] K. L. Wang, H. Zhao, and B. L. Lu, "A meta-top down method for large scale hierarchical classification," in *Proceedings of the 2012 ECML/PKDD Discovery Challenge Workshop on Large-Scale Hierarchical Text Classification*, Bristol, 2012.

ISSN 2395-8618

Instance Selection to Improve Gamma Classifier

Jarvin A. Antón-Vargas, Yenny Villuendas-Rey, and Itzamá López-Yáñez

***Abstract*—Pre-processing the dataset is an important stage in the Knowledge Discovery in Datasets (KDD) process. Filtering noise through instance selection is a necessary task. With this, the risk to use misclassified and non-representative instances to train supervised classifiers is reduced. This study aims at improving the performance of the Gamma associative classifier, by introducing a novel similarity function to guide instance selection. The experimental results, over 15 datasets, include several instance selection methods, and their influence in the performance of Gamma classifier is analyzed. The effectiveness of the proposed similarity function is tested, obtaining good results according to classifier accuracy and instance retention ratio.**

***Index Terms*—Gamma classifier, instance selection, data pre-processing, similarity functions.**

I. Introduction

The training dataset plays a key role for supervised classification. Training data allows building classifiers able to estimate the label or class of a new unseeing instance. Several researchers have pointed out that if the dataset has high quality instances, the classifier can produce predictions that are more accurate [1]. However, in several real-world applications, it is not possible to obtain a training set without noisy and mislabeled instances. To overcome this problem, several algorithms for instance selection and construction have been proposed [1] [2].

The Gamma classifier [3] [4] is a recently proposed supervised classifier, and it has been applied successfully to several prediction tasks, such as air quality monitoring [5], pollutant prediction [6] and development effort prediction of software projects [7]. Despite of the excellent performance of the Gamma classifier, it is noted that it is affected by noisy or mislabeled instances.

Most of the instance selection algorithms are designed for the Nearest Neighbor (NN) classifier [8]. Little work has been done for selecting instance for other supervised classifiers, such as ALVOT [9] [10] and Neural Networks [11], and these proposals are no directly applicable to the Gamma classifier.

Manuscript received on December 14, 2015, accepted for publication on April 23, 2016, published on October 30, 2016.

J. A. Antón-Vargas is with the Computer Science Department of the University of Ciego de Ávila, Cuba. (e-mail: janton@ unica.cu).

Y. Villuendas-Rey (corresponding author) is with the CIDETEC, Instituto Politécnico Nacional, Mexico City, Mexico (e-mail: yvilluendasr@ipn.mx).

I. López-Yáñez is with the CIDETEC, Instituto Politécnico Nacional, Mexico City, Mexico (e-mail: ilopezy@ipn.mx).

This paper proposes a novel similarity function based on the Gamma operator of the Gamma classifier, and use it for similarity comparisons in NN instance selection algorithms. The thorough experimental study carried out shows the significant performance gains of the proposed approach.

With the advance of digital technology, the technological advances of computers, the continued growth of the computerization of society, and the development of the web it has facilitated the easy accumulation of data (business data, websites, data warehouses, etc.) and information [12]. This phenomenon has referred by some authors as “drowning in information” [13], because working with them is tedious and involves a high computational cost [14]. As example the data collected by institutions such as the particle collider in Switzerland CERN or data obtained in sciences like astronomy and biology in studying the human genome and protein sequencing [15] [16]. Any of these fields of study can work with data in the order of petabytes. Researchers daily have to face this problem, mainly to the analysis of databases with a large dimensionality, we must understand large dimensionality as numerous features and a high number of instances.

It is common for researchers when they use values from real problems, have to deal in many cases with data represented by many characteristics and only a few of them are directly related to the objective of the problem. Redundancy may exist where several features can have a high correlation, which makes it not necessary to include them all in the final model. Find interdependence also applies, so that two or more features contains relevant information, and if is excluded any of them, it can make this information useless [17].

Another important problem arises when the training set is excessively large relative to the number of instances, making impracticable the supervised learning. By other hand if the classification is practicable, to cite one example; when it contains class imbalance problems, in most of the time the algorithms opt for the majority class and include in that category objects of minority classes.

It is normal that there are also instances that do not contain relevant information or are not significant for the classification of the problem in question. Once the preprocessing is performed through the selection of instances, the model could predict on the basis of a training set, adjusted to the most representative elements of the problem in question and in turn reducing the time execution, this are essential elements to arrive at a desirable outcome [18].

ISSN 2395-8618

II. Previous works

As is known in the supervised classification process, the training is an important phase. Such learning is guided by datasets containing training cases. It is usual in real life problems, that this training sets contain vain information for the classification process; understood by this superfluous cases, which may contain noise or may be redundant [18]. That is why removing these cases from the initial training set is needed.

Given a training set, the aim of the instance selection methods is to obtain a subset that does not contain superfluous instances, so that the accuracy of the classification obtained using the resulting subset of instances is not degraded. These methods can generate subsets incrementally, adding instances as the knowledge space is explored. Another alternative is to start from the initial set of instances and thus eliminating instances to find the optimal subset according to the algorithm used. Through the selection of instances, the training set is reduced, which could be useful in reducing the time during the learning process, particularly in based instance models where the classification of a new instance uses the entire training set.

Several models have been proposed in the literature to address this task, obtaining good results considering the main objective of this preprocessing technique. In the next sections, we will discuss different models, observing the diversity of approaches and the elements considered by the researchers in order to improve the supervised classification model.

A. Instance selection for Nearest Neighbor classifier

Most of the instance selection methods proposed are based on the NN classifier. CNN (Condensed Nearest Neighbor) [19] has been one of the first, this method follows the incremental model and its initial routine randomly include in the result set S, an instance belonging to each of the class problem. Then each instance of the original set is classified using as training set S; if the instance is classified incorrectly, then it is included in the set S pursuing the idea that if there is another instance like this then be classified correctly. One drawback of this model is that it could hold instances that constitute noise in the result set.

Since this method is derived a set of methods among which are the SNN (Selective Nearest Neighbor) [20], which generates the set S following the criterion that each member of the original set is closer to a member of the result set S than any other, this could be understood as each instance would be correctly classified by the NN classifier using as training set to S. Another variation within this group is the GCNN (Generalized Condensed Nearest Neighbor) [21], its operation is identical to CNN, and this just includes an absorption criterion according to a threshold. This means that for each instance, the absorption is calculated in terms of the nearest neighbors and closest enemies (those closest instances to a member of a class but belonging to another class).

Another method for selecting instances is the ENN (Nearest Neighbor Edited) [22] that focuses on discard the noisy instances present in the training set. This method discards those instances when the class is different from the majority class of their closest k neighbors (ENN generally used k = 3). An extension of the ENN is the RENN (Repeated ENN), this method applies ENN repeatedly until all instances present in the resulting set S belong to the same class as the class that belongs the majority of their k nearest neighbors. Another variant is the All-KNN [23], this method works iterating the routine k-NN algorithm k times, labeling the instances that are misclassified. Once the iterations are stopped, all labeled instances are discarded from the training set.

B. Instance selection for Artificial Neural Networks

It is well known that Artificial Neural Networks (ANNs) can produce robust performance when a large amount of data is available. However, the noisy data may produce inconsistent and unpredictable performance in ANN's classification. In addition, it may not be possible to train ANN or the training task cannot be effectively carried out without data reduction when a data set is too huge.

In the literature, we can find many researches trying to obtain the best training set for this powerful technique. In [24] the authors propose a new hybrid model of ANN and Genetic Algorithms (GAs) for instance selection. An evolutionary instance selection algorithm reduces the dimensionality of data and eliminate noisy and irrelevant instances. In addition, this study simultaneously searches the connection weights between layers in ANN through an evolutionary search. By the other hand the genetically evolved connection weights mitigate the well-known limitations of gradient descent algorithm.

In the same way to simplify the space dimension of input information and reduce the complexity of network structure, the information entropy reduction theory is brought in. Trying to aim at the main shortage of ANN (the converging speed is often slow and the network is easily involved in local optimum), is introduced the Particle Swarm Optimization (PSO) [25].

C. Instance selection for other classifiers

In the Logical Combinatorial approach to Pattern Recognition (LCPR), ALVOT is a model for supervised classification. This model was inspired in the works by Zhuravlev, and it is based on partial precedence. Let understand as partial precedence, the principle of calculating the similarity between objects using comparisons between their partial descriptions. A partial description is a combination of features. This is the way that many scientists such as physicians, and other natural scientists, establish comparisons among objects in real world problems [26].

When a new instance is classified, many partial comparisons with all the objects in the training set have to be calculated. This can be very time consuming, while the cardinality of the set increases. That is why an instance selection method for ALVOT was introduced in [27] [28] with good results. In both

ISSN 2395-8618

algorithms, the authors introduce a voting strategy to select the most relevant instances.

In addition, Genetic Algorithms have been used for instance selection in the context of ALVOT classifier [9]. The algorithm presented in [9] start generating randomly the initial population. The input parameters for the algorithm are the population size and iteration number. Then the population's individuals are sorted according to their fitness. The first and last individuals are crossed, the second is crossed with the penultimate and this process is repeated until finishing the population. They are crossed using a 1-point crossover operator in the middle of the individual. The fitness function is the ratio of well classified objects. The mutation operator is evaluated for each individual in the population changing randomly the values of an individual's gene. Then the fitness is evaluated for this new population. The original individuals together with those obtained by crossing and mutation are sorted in descending order according to their fitness and those with highest fitness are chosen (taking into account the population size). The new population is used in the next iteration of the algorithm.

To this classifier is possible apply others instances selections methods, such as the classical models based on NN rule. An analogue solution was reported by Decaestecker [29] and Konig et al. [30], in which the training set is edited for a Radial Based Function network, using a procedure originally designed for NN.

Because of the importance of data preprocessing for any classifier, it is interesting to note that for associative classifiers, such as Gamma, to the best of our knowledge, there are no analysis of the impact of instance selection in classifiers performance. Considering that this approach generates a memory of fundamental patterns, and associates instances with their respective classes, we hypothesize that this association process may provide better results if this memory is created from refined and representative instances of the problem to solve.

III. Gamma Classifier

The Gamma associative classifier belongs to the associative approach of Pattern Recognition, created in the National Polytechnic Institute of Mexico [31]. The Gamma classifier is based on two operators named Alpha and Beta, which are the foundation of the Alpha-Beta associative memories [32]. The Alpha and Beta operators are defined in a tabular form considering the sets $A = \{0, 1\}$ and $B = \{0, 1, 2\}$, as shown in Fig. 1.

In addition to the Alpha and Beta operator, the Gamma classifier also uses two other operators: the u_β operator and the generalized gamma similarity operator, γ_g. The unary operator u_β receives as an input a binary n-dimensional vector, and returns a number $p \in \mathbb{Z}^+$ according to the following expression:

$$u_\beta = \sum_{i=1}^{n} \beta\,(x_i, x_i) \qquad (1)$$

The generalized gamma similarity operator receives as input two binary vectors $x \in A^n$ and $y \in A^m$ with $n, m \in \mathbb{Z}^+, n \leq m$, and also a non-negative integer θ, and returns a binary digit, as follows:

$$\gamma_g(x, y, \theta) = \begin{cases} 1 & if\ m - u_\beta[\alpha(x, y) mod 2] \leq \theta \\ 0 & otherwise \end{cases} \qquad (2)$$

That is, the γ_g operator returns 1 if the input vectors differentiate at most in θ bits, and returns zero otherwise.

The Gamma classifier is designed for numeric patterns, and assumes that each pattern belongs to a single class. However, as the generalized gamma similarity operator receives as input two binary vectors, the Gamma classifier codifies numeric instances using a modified Johnson-Möbius code [3]. In Figure 2 we show a simplified schema of the Gamma classifier.

$\alpha : A \times A \rightarrow B$

x	y	$\alpha(x, y)$
0	0	1
0	1	0
1	0	2
1	1	1

$\beta : B \times A \rightarrow A$

x	y	$\beta(x, y)$
0	0	0
0	1	0
1	0	0
1	1	1
2	0	1
2	1	1

Fig. 1. Tabular definition of Alpha and Beta operators.

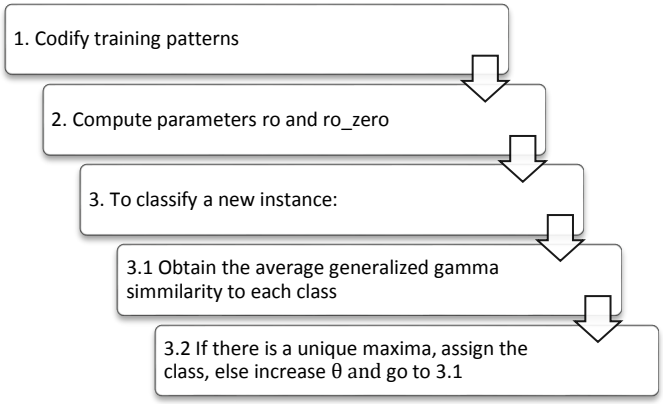

Fig. 2. Simplified schema of the classification process with the Gamma classifier.

The training process of Gamma uses the modified Johnson Möbius code to codify the instances in sets of bits, this allows to obtain the values of ρ and ρ_0, necessaries parameters to the next phase. This values are determined finding the lowest of all e_m values which are the greatest of all values for each feature, as it shows in the next formulas:

ISSN 2395-8618

$$\rho = \bigwedge_{i=1}^{p} e_m(j) \tag{3}$$

$$e_m(j) = \bigvee_{i=1}^{p} x_i^j \tag{4}$$

In the classification phase, the new instance is codified with the same Johnson-Möbius code, thus a θ parameter its initialized with zero value, then all class in the training dataset are grouped by class and its calculated the $\gamma_g(x, y, \theta)$ between the new instance and all instances for each kind of class. The final classification is assigned from the class with greatest similarity value calculated by:

$$C_i = \frac{\sum_{i=1}^{n} \gamma_g(x, y, \theta)}{n} \tag{5}$$

If the value of C_i is not unique, then θ increment its value relaxing the similarities between the instances and all the process is done again, until the θ gets a value equal to ρ. In the case that at the end of all iterations there's not a unique maximum for a class, then it is assigned the first class with maximum similarity.

A detailed characterization of the Gamma classifier can be found in [33]. Its use has been extended into datasets with different characteristics and with different objectives; mainly in classification tasks, for which it was designed. It also has been used in interpolation tasks and functions exploration, these last ones for which it was not designed. It was determined that good results can be expected when data induce a function. In other words, if each input element has a single output pattern or class, then the classifier is competitive and even superior to other algorithms. Otherwise, when an input pattern has various output patterns then the algorithm is in a situation not expected by the original algorithm.

The algorithm has a competitive performance also in the case of a known sequence is present, which output will be the known value that best matches with this sequence. However, it is clear to suppose that it may happen that this sequence is not exactly known, but it is near the border between two or more known sequences. Then, an output near the border between the known corresponding outputs is obtained.

IV. Gamma based similarity

According to the classification strategy of the Gamma classifier, we propose a similarity function to compare pairs of instances, regarding the θ parameter. This allows us to detect noisy or mislabeled instances.

The proposed Gamma Based Similarity (GBS) uses the generalized gamma operator, but it considers the standard deviation of the feature instead of the θ parameter. Let be X and Y to instances, the Gamma based similarity between them is computed as:

$$GBS(X, Y) = \sum_{i=1}^{p} \gamma_g(x_i, y_i, \sigma_i) \tag{3}$$

where p is the number of features describing the instances, σ_i is the standard deviation of the i-th feature, and x_i and y_i are the binary vectors associated with the i-th feature in instances X and Y, respectively.

Considering this novel similarity, we are able to apply several instance selection algorithms which were designed for the NN classifier, and test their performance in the filtering of noisy and mislabeled instances for the Gamma classifier.

As shown, in the instance selection methods described in previous sections, the similarity between instances is critical to the operation of any method that seeks to select or discard instances of a training set. In the case of Gamma associative classifier, a new similarity function it is proposed, based mainly on the criteria to take a decision over the values of the features of each instance. The original similarity, works in dependence on a θ, value that is dynamically updated, this dynamic value allows a relaxation of the classifier which is not a good criterion for selection instances model.

That is why the use of standard deviation for the θ value is proposed. The criteria taken into account for the adoption of this variant are the benefits that has the standard deviation over a set of values, in this case would be the values of each feature. First, keep in mind that the standard deviation is by far the measure generally used to analyze the variation in a group of values. It is a measure of absolute variation [34] that calculate the real amount of variation present in a dataset. This allows to know more about the dataset of interest. It is not enough to know the measures of central tendency, we also need to know the deviation present in the data with regard to the average of these values. Its calculation is determined by the following formula:

$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \tilde{x})^2}{n-1}} \tag{6}$$

where $\tilde{x}$ is the arithmetic mean.

This allow us to know for each feature its dispersion and get a better fit of the decision criteria that define if two values are similar or not.

V. Experimental results

To test the influence of instance selection algorithms in the performance of the Gamma classifier, we use some of the most representative instance selection algorithms reported in the literature, and apply them over well-known datasets from the Machine Learning repository of the University of California at Irvine [13]. Table 1 shows the characteristics of the selected datasets.

ISSN 2395-8618

TABLE I
DATASETS USED IN THE NUMERICAL EXPERIMENTS

Datasets	Instances	Attributes	Classes
balance-scale	625	4	3
diabetes	768	8	2
ecoli	336	7	8
hayes-roth	160	4	3
heart-statlog	270	13	2
ionosphere	351	34	2
iris	150	4	3
liver-disorders	345	6	2
mfeat-morphological	2000	6	10
new-thyroid	215	5	3
page-blocks	5473	10	5
pendigits	10992	16	10
spambase	4601	57	2
vehicle	846	18	4
wine	154	13	3

We selected error-based editing methods due to their ability of smoothing decision boundaries and to improve classifier accuracy. The selected methods are the Edited Nearest Neighbor (ENN) proposed by Wilson [14], the Gabriel Graph Editing method (GGE) and Relative Neighborhood Graphs (RNGE) proposed by Toussaint [15] and the MSEditB method, proposed by García-Borroto et al. [16].

The ENN algorithm (Edited Nearest Neighbor) is the first error-based editing method reported [14]. It was proposed by Wilson in 1972 and it consist on the elimination of the objects misclassified by a 3-NN classifier. The ENN works by lots, because it flags the misclassified instances and then simultaneously deletes them all, which guaranteed order independence. The ENN has been extensively used in experimental comparisons, showing very good performance [1].

The GGE algorithm is based on the construction of a Gabriel graph. A Gabriel graph is a directed graph such that two instances $x \in U$ and $y \in U$ form an arc if and only if $\forall z \in U\ (d((x+y/2),z) > d(x,y)/2)$, where d is a dissimilarity function. That is, two instances x and y are related in a Gabriel graph if there is no object in the hypersphere centered in the middle point of x and y, and with radius the distance between x and y.

The GGE algorithm consists in deleting those instances connected to others of different class labels. It deletes borderline instances, and keep class representative ones.

Similar to GGE, the RNGE [15] uses a Relative Neighborhood graph to determine which instance delete. A Relative Neighborhood graph is a directed graph such that two instances $x \in U$ and $y \in U$ form an arc if and only if $\forall z \in U\ (d(x,y) < max\{d(x,z), d(y,z)\})$, where d is a dissimilarity function.

The MSEditB algorithm [16] uses a Maximum similarity graph to select the objects to delete. A Maximum similarity graph is a directed graph such that each instance is connected to its most similar instances. Formally, let be S a similarity function, an instance $x \in U$ form an arc in a Maximum similarity graph with an instance $y \in U$ if and only if $d(x,y) = \max_{z \in U} d(x,z)$.

The MSEditB algorithm deletes an instance if it has a majority of its predecessors and successors instances not of its class.

All algorithms were implemented in C# language, and the experiments were carried out in a laptop with 3.0GB of RAM and Intel Core i5 processor with 2.67HZ. We cannot evaluate the computational time of the algorithms, because the computer was not exclusively dedicated to the execution of the experiments.

To compare the performance of the algorithms, it was used the classifier accuracy. The classifier accuracy is measure as the percent of correctly classified instances. Let be X the testing set, $l(x)$ the true label of instance $x \in X$, and $d(x)$ the decision made by the classifier. The classifier accuracy is defined as:

$$cc(X) = \frac{|\{x \in X: l(x) = d(x)\}|}{|X|} * 100 \qquad (7)$$

It was also computed the Instance Retention Ratio (IRR) for every algorithm, in order to determine the number of selected instances. The IRR is measured as the ratio of instances that are kept by the instance selection algorithm. Let be T the set of training instances, and $E \subseteq T$ the set of instances selected. The IRR is computed as:

$$IRR = \frac{|E|}{|T|} \qquad (8)$$

In Table 2, we show the accuracy of the Gamma classifier without selecting instances (Gamma) and the accuracy of the Gamma classifier trained using the instances selected by ENN, GGE, RNGE and MSEditB, respectively. Results corresponding to accuracy improvements of Gamma classifier are highlighted in bold.

TABLE II
CLASSIFIER ACCURACY AFTER SELECTING INSTANCES

Datasets	Gamma	ENN	GGE	RNGE	MSEditB
balance-scale	83.845	**74.071**	83.372	**90.719**	**90.241**
diabetes	59.516	**61.471**	58.083	**61.470**	**60.947**
ecoli	50.980	**52.433**	46.827	50.089	48.966
hayes-roth	74.375	71.250	65.625	68.750	**79.375**
heart-statlog	81.852	81.852	**82.593**	**82.222**	**82.963**
ionosphere	**74.373**	64.111	35.889	*	*
iris	88.667	**90.000**	**90.000**	**90.000**	**90.000**
liver-disorders	**57.697**	55.059	56.546	54.504	55.387
mfeat-morphological	46.000	43.500	**46.650**	41.850	41.650
new-thyroid	80.476	**81.861**	80.931	**81.407**	**82.316**
page-blocks	77.160	**77.873**	77.105	76.557	**77.471**
pendigits	64.456	**64.483**	64.465	**64.483**	**64.547**
spambase	71.310	70.441	**75.287**	73.918	70.202
vehicle	**59.592**	58.641	56.146	57.817	58.053
wine	72.451	71.863	70.294	**73.007**	72.451
Increases of classifier accuracy		7	4	**7**	8

* The RNGE and MSEditB algorithms select no instance.

ISSN 2395-8618

In 12 of the tested datasets, the instance selection algorithms were able to improve the accuracy of the Gamma classifier. However, in datasets ionosphere, liver-disorders and vehicle no improvement was achieved. In particular, the ionosphere dataset shows a high degree of class overlapping, such that both RNGE and MSEditB algorithms do not kept any instance.

Despite this pathological behavior, the instance selection algorithms exhibit a very good performance, with several improvements in classifier accuracy.

In Table 3, we show the Instance Retention Rate (IRR) obtained by ENN, GGE, RNGE and MSEditB, respectively. Best results are highlighted in bold.

TABLE III
INSTANCE RETENTION RATIO OBTAINED BY THE ALGORITHMS

DATASETS	ENN	GGE	RNGE	MSEDITB
balance-scale	0.912	0.847	0.895	**0.777**
diabetes	0.847	**0.669**	0.770	**0.669**
ecoli	0.844	0.888	0.754	**0.675**
hayes-roth	0.866	0.647	0.278	**0.645**
heart-statlog	0.887	0.772	0.852	**0.763**
ionosphere	0.641	**0.359**	*	*
iris	0.955	0.973	0.943	**0.934**
liver-disorders	0.838	**0.537**	0.739	0.580
mfeat-morphological	0.826	0.936	0.715	**0.639**
new-thyroid	0.982	0.967	0.980	**0.957**
page-blocks	0.979	**0.958**	0.963	**0.958**
pendigits	0.997	0.997	0.995	**0.992**
spambase	0.952	**0.856**	0.870	0.902
vehicle	0.838	0.851	0.777	**0.675**
wine	0.986	**0.889**	0.988	0.951

* The RNGE and MSEditB algorithms select no instance.

Both GGE and MSEDitB were the algorithms with best results according to IRR. GGE obtained IRR varying from 0.35 to 0.95, and MSEditB from 0.63 to 0.992. ENN and RNGE obtained inferior results.

However, to determine the existence or not of significant differences in algorithm´s performance it was used the Wilcoxon test [17].

The Wilcoxon test is a non-parametric test recommended to statistically compare the performance of supervised classifiers. In the test, we set as null hypothesis that there is no difference in performance between the gamma classifier without instance selection (Gamma) and the gamma classifier after instance selection, and as alternative hypothesis that instance selection algorithms lead to better performance. We set a significant value of 0.05, for a 95% of confidence.

Tables 4 and 5 summarize the results of the Wilcoxon test, according to classifier accuracy and instance retention rate, respectively.

The Wilcoxon test obtains probability values greater than the significance level, and thus, we do not reject the null hypothesis. These results confirm that instance selection algorithms using the proposed similarity function are able to preserve classifier accuracy, using a small number of instances.

TABLE IV
WILCOXON'S TEST COMPARING CLASSIFIER ACCURACY

Gamma vs.	w-l-t	Z	Probability	Decision
ENN	6-8-1	-1.420	0.245	No reject
GGE	6-9-0	-1.420	0.156	No reject
RNGE	8-7-0	-0.454	0.650	No reject
MSEditB	8-6-1	-0.094	0.925	No reject

TABLE V
WILCOXON'S TEST COMPARING CLASSIFIER ACCURACY

Gamma vs.	w-l-t	Z	Probability	Decision
ENN	15-0-0	-3.408	0.001	Reject
GGE	15-0-0	-3.408	0.001	Reject
RNGE	15-0-0	-3.408	0.001	Reject
MSEditB	15-0-0	-3.408	0.001	Reject

According to instance retention ratio, the Wilcoxon test rejects the null hypothesis in all cases. That is, the number of selected objects using ENN, GGE, RNGE and MSEditB with the proposed gamma based similarity function, was significantly lower than the original number of instances in the training set.

The experimental results carried out show that selecting instances by using a similarity function based on the Gamma operator maintains classifier accuracy, and also reduces the cardinality of the training sets, diminishing the computational cost of the Gamma classifier.

VI. CONCLUSION

We considered that instance selection process based on the proposed similarity function contributes to the improvement of the Gamma associative classifier by maintaining its performance with low computational complexity. As future work, we plan to experiment with the feature weight assignment process, in order to further improve the Gamma classifier.

REFERENCES

[1] S. García, J. Derrac, J. R. Cano and F. Herrera, "Prototype Selection for Nearest Neighbor Classification: Taxonomy and Empirical Study," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 34, pp. 417-435, 2012.

[2] I. Triguero, J. Derrac, S. Garcia and F. Herrera, "A taxonomy and experimental study on prototype generation for nearest neighbor classification," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions,* vol. 42, pp. 86-100, 2012.

[3] I. López-Yáñez, "Clasificador automático de alto desempeño (MS dissertation)," 2007.

[4] I. López-Yáñez, L. Sheremetov and C. Yáñez-Márquez, "A novel associative model for time series data mining," *Pattern recognition Letters,* vol. 41, pp. 23-33, 2014.

[5] C. Yánez-Márquez, I. López-Yánez and G. Morales, "Analysis and prediction of air quality data with the gamma classifier," *Progress in Pattern Recognition, Image Analysis and Applications,* pp. 651-658, 2008.

[6] I. Lopez-Yanez, A. J. Argüelles-Cruz, O. Camacho-Nieto and C. Yanez-Marquez, "Pollutants time-series prediction using the Gamma classifier,"

ISSN 2395-8618

International Journal of Computational Intelligence Systems, nº 4, pp. 680-711, 2012.

[7] C. López-Martın, I. López-Yánez and C. Yánez-Márquez, "Application of Gamma Classifier to Development Effort Prediction of Software Projects," *Appl. Math,* vol. 6, nº 3, pp. 411-418, 2012.

[8] T. M. Hart and P. E. Cover, "Nearest Neighbor pattern classification," *IEEE Transactions on Information Theory,* vol. 13, pp. 21-27, 1967.

[9] M. A. Medina-Pérez, M. García-Borroto, Y. Villuendas-Rey and J. Ruiz-Shulcloper, "Selecting objects for ALVOT," *Progress in Pattern Recognition, Image Analysis and Applications,* pp. 606-613, 2006.

[10] M. A. Medina-Pérez, M. García-Borroto and J. Ruiz-Shulcloper, "Object selection based on subclass error correcting for ALVOT," *Progress in Pattern Recognition, Image Analysis and Applications,* pp. 496-505, 2007.

[11] H. Ishibuchi, T. Nakashima and M. Nii, "Learning of neural networks with GA-based instance selection," *IFSA World Congress and 20th NAFIPS International Conference,* vol. 4, pp. 2102-2107, 2001.

[12] H. M. HUAN-LIU, "On Issues of Instance Selection," *Data Mining and Knowledge Discovery,* vol. 6, pp. 115-130, 2002.

[13] A. Szalay, "Drowning in data," *Scientific American.,* 1999.

[14] B. Z. Jun-Yan, "Effective and Efficient Dimensionality Reduction for Large-Scale and Streaming Data Preprocessing," *IEEE Transactions on Knowledge and Data Engineering,* vol. 18, 2006.

[15] L. Y. Liu, "Efficient feature selection via analysis of relevance and redundancy," *Journal Machine Learning Research,* vol. 5, pp. 1205-1224, 2004.

[16] J. A. Antón-Vargas and C. Santiesteban-Toca, "Selección de algoritmos para la predicción de contactos interresiduales de proteínas," *9no Congreso Internacional Biotecnología Vegetal,* 2013.

[17] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *JMLR Special Issue on Variable and Feature Selection,* pp. 1157-1182, 2003.

[18] J. Olvera-López, J. Carrasco-Ochoa, J. Martínez-Trinidad and J. Kittler, "A review of instance selection methods," *Artif Intell Rev,* vol. 34, pp. 133-143, 2010.

[19] P. E. Hart, "The condensed nearest neighbor rule," *IEEE Trans Inf Theory,* vol. 14, pp. 515-516, 1968.

[20] G. L. Ritter, H. Woodruff, S. R. Lowry and T. L. Isenhour, "An algorithm for a selective nearest neighbor decision rule," *IEEE Trans Inf,* vol. 21, pp. 665-669, 1975.

[21] C. Chien-Hsing, K. B.H. and C. Fu, "The generalized condensed nearest neighbor rule as a data reduction method," *in 18th international conference on pattern recognition,* pp. 556-559, 2006.

[22] D. L. Wilson, "Asymptotic properties of nearest neighbor rules using edited data," *IEEE Transactions on Systems, Man and Cybernetics,* vol. 1, pp. 408-421, 1972.

[23] I. Tomek, "An experiment with the edited nearest-neighbor rule," *IEEE Trans Syst Man Cybern,* vol. 6, pp. 448-452, 1976.

[24] K. Kyoung-jae, "Artificial neural networks with evolutionary instance selection for financial forecasting," *Expert Systems with Applications,* vol. 30, pp. 519-526, 2006.

[25] Z. Li-Ning, Z. Qi, L. Da-Chao and A. Jing, "Research on the Pre-warning Model of Enterprise Financial Crisis Based on the Information Entropy and PSO-ANN," *International Conference on E-Business and E-Government (ICEE),* pp. 1567-1570, 2010.

[26] J. Ruiz-Shulcloper and M. Abidi, "Logical Combinatorial Pattern Recognition: A Review," de *Recent Research Developments in Pattern Recognition. Transword Research Networks*, 2002, pp. 133-176.

[27] J. A. Carrasco-Ochoa and J. F. Martínez-Trinidad, "Editing and Training for ALVOT, an Evolutionary Approach," *Lecture Notes in Computer Science,* vol. 2690, pp. 452-456, 2003.

[28] J. A. Carrasco-Ochoa and J. F. Martínez-Trinidad, "Combining Evolutionary Techniques to Improve ALVOT Efficiency," *WSEAS Transactions on Systems,* vol. 2, pp. 1073-1078, 2003.

[29] C. Decaestecker, "NNP: A neural net classifier using prototype," *International Conference on Neural Networks,* pp. 822-824, 1993.

[30] A. König, R. J. Rashhofer and M. Glesner, "A novel method for the design of radial-basisfunction networks and its implication for knowledge extraction," *International Conference on Neural Networks,* pp. 1804-1809, 1994.

[31] I. López-Yáñez, "Clasificador automático de alto desempeño," Mexico, D.F., 2007.

[32] L. C. Yáñez-Márquez and J. L. Díaz, "Memorias Asociativas basadas en relaciones de orden y operaciones binarias," *Computación y Sistemas,* vol. 6, nº 4, pp. 300-311.

[33] I. López-Yáñez, "Teoría y aplicación del clasificador asociativo Gamma," Mexico, D.F., 2011.

[34] I. R. Miller, J. E. Freund and R. Johnson, Probabilidad y Estadística para Ingenieros, La Habana: Félix Varela, 2005.

[35] G. T. Toussaint, "Proximity Graphs for Nearest Neighbor Decision Rules: Recent Progress," de *34 Symposium on Computing and Statistics INTERFACE-2002*, Montreal, Canada, 2002.

[36] J. Demsar, "Statistical comparison of classifiers over multiple datasets," *The Journal of Machine Learning Research,* vol. 7, pp. 1-30, 2006.

[37] Z. Pawlak, "Rough Sets," *International Journal of Information & Computer Sciences,* vol. 11, pp. 341-356, 1982.

[38] Y. Caballero, R. Bello, Y. Salgado and M. M. García, "A method to edit training set based on rough sets," *International Journal of Computational Intelligence Research,* vol. 3, pp. 219-229, 2007.

[39] J. L. Díaz and C. Yáñez-Márquez, "Memorias Asociativas basadas en relaciones de orden y operaciones binarias".

[40] M. García-Borroto, Y. Villuendas-Rey, J. A. Carrasco-Ochoa and J. F. Martinez-Trinidad, "Using Maximum Similarity Graphs to edit nearest neighbor classifiers," *Lecture Notes on Computer Science,* vol. 5856, pp. 489-496, 2009.

[41] M. A. Medina-Pérez, M. García-Borroto, Y. Villuendas-Rey and J. Ruiz-Shulcloper, "Selecting Objects for ALVOT," *Progress in Pattern Recognition, Image Analysis and Applications,* pp. 606-613, 2006.

[42] A. Newman and D. Asuncion, "UCI machine learning repository," 2007.

ISSN 2395-8618

Understanding Human Preferences for Summary Designs in Online Debates Domain

Nattapong Sanchan, Kalina Bontcheva, and Ahmet Aker

***Abstract*—Research on automatic text summarization has primarily focused on summarizing news, web pages, scientific papers, etc. While in some of these text genres, it is intuitively clear what constitutes a good summary, the issue is much less clear cut in social media scenarios like online debates, product reviews, etc., where summaries can be presented in many ways. As yet, there is no analysis about which summary representation is favored by readers. In this work, we empirically analyze this question and elicit readers' preferences for the different designs of summaries for online debates. Seven possible summary designs in total were presented to 60 participants via online channels. Participants were asked to read and assign preference scores to each summary design. The results indicated that the combination of Chart Summary and Side-By-Side Summary is the most preferred summary design. This finding is important for future work in automatic text summarization of online debates.**

***Index Terms*—Summary design, automatic summarization, summary representation, text mining, information extraction.**

I. Introduction

DUE to the availability of social media sites and the exponential growth of Internet use, online users communicate and share their opinions in textual form in online media. Debate web sites are one example of the media in which users express their opinions about their favorite debates. As more and more content is published it becomes increasingly difficult for readers and potential debate participants to easily or quickly digest and understand the overall details in controversial discussions. Automatic text summarization can be used to overcome this problem by helping users digest the information on web forums.

Related work has investigated different summarization approaches such as aspect-based [1]–[3], meeting [4], [5], contrastive, [6]–[9] and comparative summarization [10]–[13]. The summary either contains statistics about negative and positive opinions provided for each aspect [14], lists most frequent positive and negative opinionated sentences [1] or contains positive and negative sentences side-by-side so that they are contrastive to each other [7]. Some studies claim that one of these outputs is preferred to the another (e.g. [8]). However, there is no empirical evidence establishing which summary output is favored by human readers. This lack of evidence requires an empirical study in order to acquire appropriate information about user preferences and summary outputs for specific purpose.

In this paper, we present an empirical study that investigates different types of summary outputs, called summary designs, for debate discussion. We aim to answer the research question: "Which summary design is the most preferred for presenting the abridged version of debate content?". To answer this question, we collected opinionated comments about climate change from the Debate discussion forum[1] and manually constructed the following summary designs: a Chart Summary, a Table Summary, a Side-By-Side Summary and a Conceptual Map. The first three designs were informed by prior research (i.e. [1], [7], [14]) and the latter was proposed in this study. In addition, we also manually constructed the combined versions of those summary designs. In total, there are 7 summary designs used in this study. Next, 60 participants were recruited to an online study. The study asked the participants to give preference scores to each summary design. We found that, the combination of the Chart Summary and the Side-By-Side Summary is the most preferred summary design. To the best of our knowledge, this is the first empirical study conducted to understand which type of summarization outputs is favored by humans, and we think that our results are a valuable contribution for future studies that aim to summarize online debates.

The rest of this paper is organized as follows: first, we briefly describe the climate change data and our approach to select salient sentences from it to construct our summaries in Section II. Section III introduces 7 different summary designs and the methodology we used to manually construct them. We discuss about the empirical study in Section IV and analyze the results in Section V. Section VI is the conclusion.

II. Data and Salient Sentence Selection

A. Data

Previous research has focused on summarizing documents in news articles, product reviews, movie reviews, medical data, and other related domains. Our aim is to investigate how to

Manuscript received on February 5, 2016, accepted for publication on June 16, 2016, published on October 30, 2016.

Nattapong Sanchan is a PhD student at the Natural Language Processing group, Department of Computer Science, University of Sheffield, United Kingdom (email: nsanchan1@sheffield.ac.uk).

Kalina Bontcheva and Ahmet Aker are with the Natural Language Processing group, Department of Computer Science, University of Sheffield, United Kingdom (email: {k.bontcheva, ahmet.aker}@sheffield.ac.uk).

The last two authors contributed equally to this work.

[1] http://www.debate.org

ISSN 2395-8618

summarize debates on the highly discussed topic of global warming or climate change[2].

Within the Debate discussion forum, people position themselves differently in the debate on the existence of global warming. This leads to debates, in which proponents and opponents of the global warming phenomenon controversially express their sentiments and opinions on diverse global warming topics. Contradictory opinions are voiced on many topics of global warming such as its characteristics, causes, consequences, and its existence. Due to a high volume of contributions, reading and digesting all these discussions are not possible for readers. A summary covering the different topics as well as the different opinions in each topic would help the reader digest the overall discussion. However, it is not clear at present what such a summary should look like. Therefore, we empirically investigate how to best present such a summary to the readers.

The data that we used to construct the summary designs were collected from the Debate discussion forum. Overall, 259 debates with total 1600 comments were collected. Examples of the debates are "Is global warming a myth?", "Is global warming fictitious?", "Is global warming true?", etc. The comment's length varies between 16 and 385 words, averaging at 91 words. Figure 1 shows an extract from the debate "Is global climate change man-made?". From the figure we see that the debate contains two opposing sides, *Agree* and *Disagree*, which are originally divided by the forum. As shown in the figure, one side argues that climate change is man-made and the other side thinks that is not the case. Both opposing sides also provide evidences for their propositions about the existence of global warming. We stored the data for each opposing side separately.

TABLE I
THE DISTRIBUTION OF SALIENT SENTENCES IN EACH FREQUENT TOPIC

Frequent Topics	Agree Side	Disagree Side	Total
gas	5	3	8
plant	15	6	21
carbon dioxide	38	14	52
climate change	17	7	24
global warming	6	6	12
government	10	5	15
science	13	6	19
Total	104	47	151

B. Salient Sentence Selection

We started exploring the debate "Is global climate change man made?"[3] since it is one of the longest debates and covers diverse topics compared to the other debates in our data. The debate contains two opposing sides of opinions: Agree and Disagree. One side argues that climate change is man-made and the other side thinks that it is not the case. We explored the data and manually extracted the top 7 frequent topics, which are mentioned in opinions expressed by global warming proponents. Those topics include *gas*, *plant*, *carbon dioxide*, *climate change*, *global warming*, *government*, and *science*. For each of these topics, we manually selected salient sentences. Our selection process was guided by the following aspects:

1) **Topic Filter.** For each opposing side, the sentences should contain or mention one of the frequent topics. Otherwise they were ignored.
2) **One Topic Assumption.** In the salient sentence selection process, sentences are considered based on the assumption that one sentence refers to only one primary topic.

This process leads in total to 151 salient sentences. Table I demonstrates the distribution of these sentences across the 7 frequent topics. The stance of the sentences is derived from the stance of the original comments, from which these sentences were extracted. After the selection process we manually presented them in the summary designs described in the next section.

III. SUMMARY DESIGNS

From the data described in the previous section we manually extracted salient sentences by using the frequent topics as the keywords. Once the sentences from each opposing side were selected they were mapped to the different summary designs. We constructed four summary designs: a Chart Summary, a Table Summary, a Side-By-Side Summary and a Conceptual Map. We also constructed the combined versions of those summary designs. In total, there are 7 summary designs used in this study.

A. Chart Summary

The Chart Summary is shown in Figure 2. It was first reported by [14]. From the figure we can see that it shows the frequent topics that are discussed in debate data, in high level. The numbers indicate the frequency of the salient sentences that agrees or disagrees with particular frequent topics (see Section II-B). The labels on the bars in the chart are the names of groups of salient sentences which indicate the central meaning of the groups.

B. Table Summary

The second summary design was proposed by [1] for the summary of product reviews. In our work, we adopt it to represent summaries for climate change debates and call it a *Table Summary*. A Table Summary mentions only one primary topic. The rows in the table are the salient sentences expressing different opinions about a frequent topic from both opposing

[2]In this paper, we use the term "global warming" and "climate change" interchangeably. In scientific context, climate change has broader meaning: the changes in climate characteristics. The earth's average temperature change, the flow of ocean current that causes the decrease and increase of temperate in some areas, rainfall, and snow falling are examples of climate change. Global warming has more specific meaning in which the temperature increases over the time [15], [16].

[3]http://www.debate.org/opinions/is-global-climate-change-man-made

ISSN 2395-8618

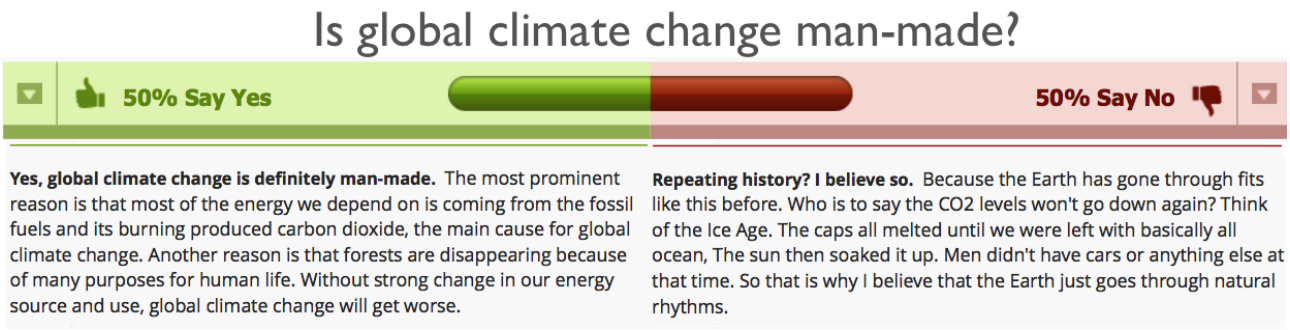

Fig. 1. An example of comments in a climate change debate

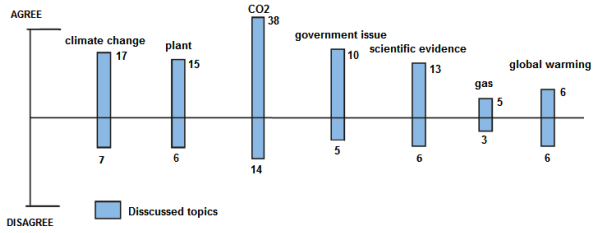

Fig. 2. Chart Summary

sides, Agree and Disagree. As shown in Figure 3, the table shows an example of a *Carbon Dioxide* topic. The numbers indicate the frequency of the salient sentences that supports the topic in each opposing side.

C. Side-By-Side Summary

Another summary design is a Side-By-Side Summary. It is adopted from [7]. Similar to the Table Summary, the Side-By-Side Summary only shows one topic at a time. As shown in Figure 4, the Side-By-Side Summary contains pairs of Agree and Disagree sentences in which each pair mentions the same topic (i.e. Carbon Dioxide) – one sentence is from the Agree side and the other is from the Disagree side. A pair is called *rebuttal*. The figures in the brackets show the frequency of the salient sentences that have been mentioned in each opposing side. The content shown in the table is only a list of rebuttals.

To construct a rebuttal, we manually matched two salient sentences from each opposing side which have the closest meaning, but opposite direction of the opinions. For instance, in the Side-By-Side Summary shown in Figure 4, one sentence mentions that carbon dioxide is the main problem that causes global warming, but the other sentence argues that it is because of the sun.

D. Conceptual Map

A Conceptual Map is a graphical representation of ideas, usually enclosed in circles or boxes. A connection of circles or boxes is drawn by a line or an arrow, which presents the relationship between ideas [17]. We applied this concept and redesigned a Conceptual Map to represent a summary for the existence of global warming issue. Similar to the Table Summary and the Side-By-Side Summary, the Conceptual Map only presents one topic at a time.

As shown in Figure 5, the opinions of public responses, regarding a *Carbon Dioxide* topic causing the global warming, are separated into two opposing sides, Agree and Disagree. On both opposing sides, people mention arguments to support their opinions about carbon dioxide. Each branch of the side shows the main category of a topic. The sub branches contain additional arguments to support the main category.

A Conceptual Map was manually constructed by determining salient sentences in each opposing side. The number of salient sentences in each opposing side that relate to a frequent topic, as the Carbon Dioxide in this example, was counted. From Figure 5, the objective of constructing sub-branches is to give additional details about Carbon Dioxide topic. When additional detail of Carbon Dioxide is found, a sub-branch is created (i.e. the sub-branch "the consumption of products leading to the emission of Carbon Dioxide"). Deeper sub-branches which elaborate the previous sub-branch are constructed until no elaboration is found.

E. Combination of Summary Designs

The Chart Summary as shown in Figure 2 is an abstract representation of topics. It does not provide full details of opinions expressed on topics whereas the other three summary designs provide evidential sentences about different opinions. Therefore, one possible way to present summaries is to combine the abstract chart with a more detailed summary. For instance, a combination of a Chart Summary and another detailed summary design would benefit readers to have a high-level summary and a detailed summary. If a reader is interested in further details, he can click on one of the chart bars (topics) to obtain more details. The detailed summary can be displayed as one of the other three summary designs. Figure 6 illustrates a combination of summary designs, namely the Chart Summary combined with the Side-By-Side Summary. In the figure, the topic CO_2 is highlighted (simulating the case where a user has clicked that topic). This activates the Side-By-Side Summary

ISSN 2395-8618

Carbon Dioxide
Agree (38)
Global warming is caused by human since the carbon content of the atmosphere is the highest over the past 650,000 years.
Carbon emissions should be lessened due to the harm they cause to the environment and to people, and a tax on them is a great way to encourage their reduction.
Transportation is one of the primary causes of the release of carbon dioxide into the air, so tackling that problem would be a good step forward in solving global warming.
I mean carbon emission is one of greatest reason for global warming and it should not be taken slightly.
Carbon dioxide, Methane and other greenhouse gasses are having unpredictable effects on the climate.
The carbon dioxide levels gradually rise as the climate cycle continues!
Carbon dioxide is one of the main problems that causes the greenhouse effect, as it traps heat on the earth's surface.
I think anything that we can do to reduce carbon emissions and thus reduce global warming is positive.
In particular, the emission of carbon dioxide and other greenhouse gasses have caused global warming to rise.
Not only automobiles are a cause, even deforestation causes a tremendous decrease in green cover resulting in decease of oxygen level which also means the increase in the level of carbon dioxide.
Disagree (14)
What people don't understand is that the greenhouse effect is only one small factor in Earth's global temperature, and carbon dioxide is only one small factor in the greenhouse effect.
But, I believe this is due to factors with the sun, not with carbon emission.
Also, as a greenhouse gas, carbon dioxide makes up only .03% of the overall gases that contribute to global warming.
Global warming is caused by the ocean temperature.
The idea that the natural process of climate change is caused by man and carbon dioxide levels is silly, as silly as Al Gore's book "An Inconvenient Truth: The Planetary Emergency of Global Warming and What We Can do about it."

Fig. 3. Table Summary

Carbon Dioxide

AGREE (38)	DISAGREE (14)
Carbon dioxide is one of the main problems that causes the greenhouse effect, as it traps heat on the earth's surface.	But, I believe this is due to factors with the sun, not with **carbon emissions**.
Not only automobiles are a cause, even deforestation causes a tremendous decrease in green cover resulting in decease of oxygen level which also means the increase in the level of **carbon dioxide**.	One of the major sources of **carbon emissions** is that there are a lot of vehicles producing greenhouse gases.
Global warming is caused by human since the **carbon content** of the atmosphere is the highest over the past 650,000 years.	All of which were no caused by **CO2 emissions**, therefore I have reason to believe that it isn't just man made but also the natural way things go on the earth.
We can reverse the global warming trends by drastically reducing **carbon dioxide emissions** into the atmosphere.	Even if we completely halt **carbon dioxide** emissions right now, abundant amounts of **CO2** and methane will be released from the now receding arctic permafrost and oceans.
By standardizing fuel economy standards, we would be reducing our **carbon footprint**, thus reducing **carbon** and aiding in fighting global warming.	Also, as a greenhouse gas, **carbon dioxide** makes up only .03% of the overall gases that contribute to global warming.
Industrialization in first world countries has led to the production and release of **carbon emissions**, masses of air, water and land pollution as well as the release of other environmentally damaging greenhouse gases into our atmosphere.	**Carbon emissions** occur naturally.
Neither party is wrong: The **carbon dioxide** levels gradually rise as the climate cycle continues!	The weather on the Earth has natural cycles and a significant amount of **carbon** is natural, emitted on Earth and naturally taken back.

Fig. 4. Side-By-Side Summary

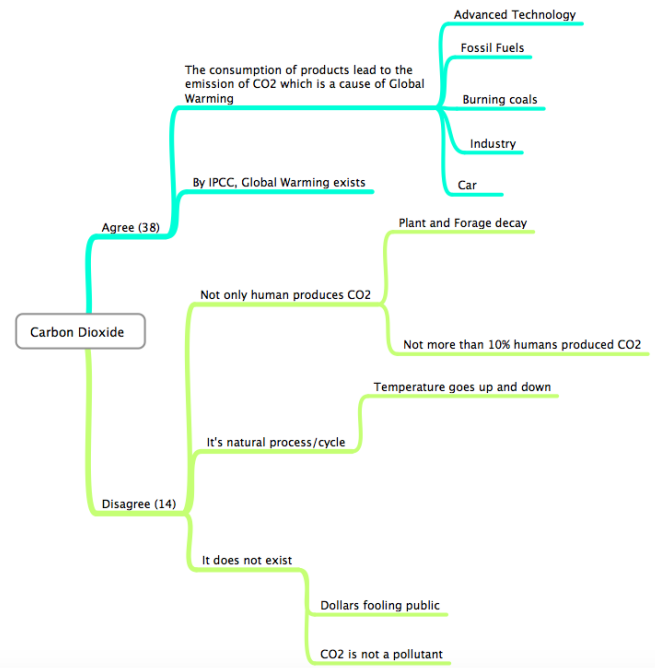

Fig. 5. Conceptual Map

and shows rebuttals for the activated topic. The idea of the combination is also applied to the Table Summary and the Conceptual Map. The combination of the Chart Summary and the Table Summary, the Chart Summary and the Side-By-Side Summary, and the Chart Summary and the Conceptual Map are called Combination 1, Combination 2 and Combination 3 respectively.

IV. The Empirical Study

To collect user preferences for the seven different summary designs we recruited 60 participants to an online questionnaire advertised via Facebook, Twitter, and the Pantip discussion forum[4].

The participants were asked to read a portion of a debate article similar to Figure 1, which contains two sets of comments with opposing opinions on the existence of global warming.

Next, the seven different summary designs and their descriptions were shown to the participants. The participants were asked to read and understand each summary design. Then, each summary design along with a list of questions was shown. They were asked to give opinions, answer questions and specify preference scores to rate each summary design.

[4]http://www.pantip.com/

ISSN 2395-8618

TABLE II
DESCRIPTIVE STATISTICS OF THE QUESTIONS TOWARD EACH SUMMARY DESIGN

Questions	Mean	Median	Mode	SD.	Min	Max
By reading the summary in the summary design, is it easy to follow ideas in debate article?						
Chart Summary	3.72	4.00	4	1.075	1	5
Conceptual Map	3.92	4.00	4	.926	1	5
Table Summary	3.23	3.00	3	1.015	1	5
Side-By-Side Summary	3.95	4.00	4	.811	2	5
Combination 1	3.70	4.00	4	.850	2	5
Combination 2	**4.22**	4.00	4[a]	.825	1	5
Combination 3	3.93	4.00	4	.989	1	5
How much the summary design is suitable for debate data?						
Chart fSummary	3.32	3.00	3	1.033	1	5
Conceptual Map	3.73	4.00	4	.841	2	5
Table Summary	3.30	3.00	4	1.124	1	5
Side-By-Side Summary	3.88	4.00	4	.922	1	5
Combination 1	3.65	4.00	4	.840	2	5
Combination 2	**4.20**	4.00	4	.755	2	5
Combination 3	3.73	4.00	4	.880	1	5
Overall, please specify your preference on the summary design.						
Chart Summary	3.58	4.00	4	1.013	1	5
Conceptual Map	3.68	4.00	4	.911	1	5
Table Summary	3.20	3.00	3	.971	1	5
Side-By-Side Summary	3.92	4.00	4	.979	1	5
Combination 1	3.57	4.00	4	.871	2	5
Combination 2	**4.17**	4.00	4[a]	.827	2	5
Combination 3	3.73	4.00	4	.954	1	5

a. Multiple modes exist. The smallest value is shown

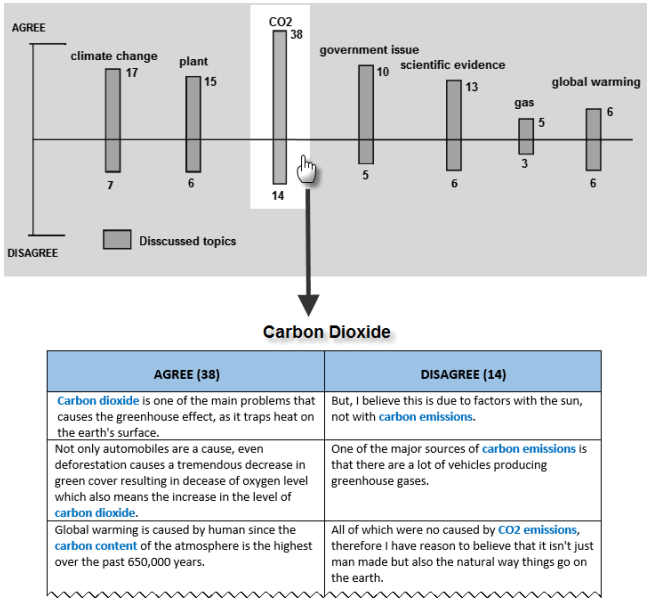

Fig. 6. The combination of a Chart Summary and a Side-By-Side Summary

Five-point Likert scales were used: excellence (5), good (4), fair (3), poor (2) and very poor (1). The questions below illustrate example questions used in the study. The first three questions are Likert-Scale questions and the last two questions are the open ended questions.

1) By reading the summary in the XXX[5], is it easy to follow ideas in debate article?
2) How much the XXX is suitable for debate data?
3) Overall, please specify your preference on the XXX.
4) What do you think is the best part of the XXX?
5) What do you think is the worst part of the XXX?

V. RESULTS AND ANALYSIS

A. *Quantitative Results*

The descriptive statistics of the empirical study shown in Table II justifies the conclusion that, the Combination 2, the combination of the Chart Summary and the Side-By-Side Summary, is the best one in representing the idea in the debate article, the most suitable one for representing debate content, and the most preferred summary design. For instance, the statistical information for the third question shows that the Combination 2 is the most preferred summary design. It has the highest means score of 4.22. This is further supported by the standard deviation. It has lower value than of the other summary designs (0.825) showing that individual responses are closer to the mean. This also applies to other questions.

Moreover, we also conducted statistical tests using the Kruskal-Wallis tests to determine if there is any statistical difference between the Combination 2 and the other summary designs. We conducted the tests for the first three questions. In the first question, the Kruskal-Wallis test indicates that there

[5]XXX refers to the name of summary design.

ISSN 2395-8618

is a statistical difference between the Combination 2 and the other summary designs, χ^2 (6, n = 60) = 51.453, $p < .001$. Also in the second question, χ^2 (6, n = 60) = 41.094, $p < .001$, reveals a statistical difference. Similarly, in the last question, χ^2 (6, n = 60) = 37.039, $p < .001$ indicates there is a statistical difference as well. For these reasons, there is a statistical difference between the Combination 2 and the other designs.

According to the descriptive statistics evidence and the results of the statistical test, we therefore conclude that the Combination 2, the combination of the Chart Summary and the Side-By-Side Summary is the most preferred output for representing the abridged version of debate content.

B. Qualitative Results

The qualitative comments that participants were asked to provide along with the Likert scores reflect the quantitative results. Participants were asked to give the most advantages and the most disadvantages for each summary design.

Positive feedback for the Chart Summary primarily focused on the concise information that the chart provides. Participants can see a clear summary at the first glance. Some points of views from our participants were "The chart can represent the overall picture of the debate topic very well.", "Picture: easy to understand and eliminate a lot of texts", and "It is an option to see the content of an article at a glance". However, we found that due to its conciseness the Chart Summary cannot provide enough information. It is unable to identify subordinated topics mentioned in debates. Readers may instantly jump to the conclusion without reading the content behind. Some participants mentioned in the study that "The chart does not provide any detail why they agree or disagree.", "Lack of details. The presenter cannot identify the sub-debated topics under each issue.", and "Opinions and argumentation are not shown".

Participants praised the Table Summary as giving detailed summary of the debate and showing clear division between Agree and Disagree information. "Full of details from each side." and "The augmentations are spitted up in two categories, it's very clear and easy to use." were the opinions from our participants. Conversely, the Table Summary is too deep in details which takes time for readers to make comparisons for each arguments. Some examples of the opinions are that "Too much data. It couldn't count as summary. It is an essay.". Another viewpoint is "It's a bit slow to read and hard to make comparison on each. It's too much wording and difficult to follow.".

In general, the advantages of the Conceptual Map focused on its readability. Participants viewed that "Key points of the topic are shown in a very easy to read and tidy way.", "Readers might want to know details briefly but not too big paragraph". In contrast, the disadvantages are "It is not so clear to a quick look. If I did not know what was this article about, I would need more time to get the correct picture.", "Might be hard to read when there are more branches in the map.", and "It's not so immediate for the comparison between each argumentation.".

The positive feedback on the Side-By-Side Summary focused on the comparison between issues and readability. The example standpoints of participants are "Easy comparison, quite concise, points laid out in a logical order" and "Compare to previous summary. It is easy to follow agree/disagree opinion as I can see it side by side. This is the most useful summary for me. and this is well-arranged.". Participants rarely provided negative feedback for this summary. Few comments mentioned that the Side-By-Side Summary contains a long list of rebuttals which takes time to read.

Participants argued that the Combination 1 (the combination of the Chart Summary and the Table summary) is better than just the Chart itself. For example, one feedback mentioned that "It is good to have details to the chart.". Still, the deep details and long representation of the Table Summary are the drawbacks of this combination. A participant said that "Still too long to be called a summary".

The positive feedback on the Combination 3 (the combination of the Chart Summary and the Conceptual Map) was similar to the feedback on the Chart Summary only. The participants commented that it is simple and concise to read. However, it is less informative compared to other summary designs. The participants indicated that the Conceptual Map is limited in providing details and thus combining it with the abstract Chart Summary does not make the Combination 3 detailed enough. For instance, participants commented that "Sometime the conceptual map is complex, especially, when the sub-issues are varied. Lacking in details compared to previous combinations.", "Less informative than previous ones overall.", and "Not easy to read and understand".

In general, participants agreed that Combination 2 (the combination of the Chart with the Side-by-Side Summary) provides a good insight into topics and is a helpful alternative to follow the discussion of debates line by line. This side-by-side visualization helps readers compare the logic and fact in each debate. Another qualitative feedback is that, Combination 2 also provides high level summary and detailed summary for each debate which provides readers clear discussion and simplicity to follow the discussion. For example, participants mentioned that "It is better arranged than combination 1, but still requires more action to see details (need to click to see the detailed summary). However, it is good option to have a chart and details as well", "Contains high level summary and details highlighted by keywords.", and "Easy to follow, logical order of points.". Negative feedback on the Side-By-Side Summary was rarely found. Only a few comments mentioned that a long list of rebuttals takes long time to read.

VI. Conclusions

Currently, there is no analysis about which summary representation for debate summaries is preferred by human

ISSN 2395-8618

readers. In this paper we have empirically investigated which summary designs humans prefer, an important question for automatically generated summaries of debates in online forums. To answer our research question, "Which summary design is the most preferred for presenting the abridged version of debate content?", we conducted an empirical study by recruiting 60 participants to give preference scores for each summary design. Our results indicated that the Chart Summary combined with the Side-By-Side Summary is the most preferred summary design for presenting the summary of debate content. Our hypothesis test indicated that there is a statistical difference in the user preferences among the summary designs. Moreover, in this study, we proposed a novel summary representation that represents summary of debate contents in a Conceptual Map. Even though it is not the most favored one, it has received some positive feedback by the participants.

These findings are important for future work in automatic text summarization of online debates. The usability of summarization systems crucially depends on their acceptability by the users, so it is necessary to address users' requirements in creating such systems. In addition, it is likely that understanding of the topic and perhaps the opinion itself may depend on the way the users access information. To determine whether this is the case, it will be one of our future work directions along with the actual automatic summarization of online debates on climate change data. Furthermore, as Combination 2 was designed for summarizing arguments that are mentioned in both opposing sides, there might be an occasion where readers want to read arguments that are mentioned by only one opposing side. We will also explore this issue in our future work.

REFERENCES

[1] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD 2004. New York, NY, USA: ACM, 2004, pp. 168–177. [Online]. Available: http://doi.acm.org/10.1145/1014052.1014073

[2] L. Zhuang, F. Jing, X. Zhu, and L. Zhang, "Movie review mining and summarization," in *Proceedings of the ACM SIGIR Conference on Information and Knowledge Management (CIKM)*, 2006.

[3] Y. Lu, C. Zhai, and N. Sundaresan, "Rated aspect summarization of short comments," in *Proceedings of the 18th international conference on World wide web*. ACM, 2009, pp. 131–140.

[4] S. Banerjee, P. Mitra, and K. Sugiyama, "Abstractive meeting summarization using dependency graph fusion," in *Proceedings of the 24th International Conference on World Wide Web*, ser. WWW 2015 Companion. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2015, pp. 5–6. [Online]. Available: http://dx.doi.org/10.1145/2740908.2742751

[5] L. Wang and C. Cardie, "Focused meeting summarization via unsupervised relation extraction," in *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, ser. SIGDIAL 2012. Stroudsburg, PA, USA: Association for Computational Linguistics, 2012, pp. 304–313. [Online]. Available: http://dl.acm.org/citation.cfm?id=2392800.2392853

[6] K. Lerman and R. McDonald, "Contrastive summarization: An experiment with consumer reviews," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, ser. NAACL-Short 2009. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009, pp. 113–116. [Online]. Available: http://dl.acm.org/citation.cfm?id=1620853.1620886

[7] M. J. Paul, C. Zhai, and R. Girju, "Summarizing contrastive viewpoints in opinionated text," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2010, pp. 66–76.

[8] H. D. Kim and C. Zhai, "Generating comparative summaries of contradictory opinions in text," in *CIKM*, D. W.-L. Cheung, I.-Y. Song, W. W. Chu, X. Hu, and J. J. Lin, Eds. ACM, 2009, pp. 385–394. [Online]. Available: http://dblp.uni-trier.de/db/conf/cikm/cikm2009.html#KimZ09

[9] M. Campr and K. Ježek, "Comparative summarization via latent semantic analysis," in *Lastest Trends in Information Technology;Proceedings of the 1st WSEAS International Conference on Information Technology and Computer Networks (ITCN 2012), Proceedings of the 1st WSEAS International Conference on Cloud Computing (CLC 2012), Proceedings of the 1st WSEAS International Conference on Programming Languages and Compilers (PRLC 2012)*, ser. Recent Advances in Computer Engineering Series 7. Stroudsburg, PA, USA: WSEAS Press, 2012, pp. 279–284. [Online]. Available: http://textmining.zcu.cz/publications/wseas-mcampr.pdf

[10] C. Zhai, A. Velivelli, and B. Yu, "A cross-collection mixture model for comparative text mining," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD 2004. New York, NY, USA: ACM, 2004, pp. 743–748. [Online]. Available: http://doi.acm.org/10.1145/1014052.1014150

[11] X. Huang, X. Wan, and J. Xiao, "Comparative news summarization using linear programming," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, ser. HLT 2011. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 648–653. [Online]. Available: http://dl.acm.org/citation.cfm?id=2002736.2002862

[12] D. Wang, S. Zhu, T. Li, and Y. Gong, "Comparative document summarization via discriminative sentence selection," *ACM Trans. Knowl. Discov. Data*, vol. 6, no. 3, pp. 12:1–12:18, Oct. 2012. [Online]. Available: http://doi.acm.org/10.1145/2362383.2362386

[13] R. Witte and S. Bergler, "Next-generation summarization: Contrastive, focused, and update summaries," in *International Conference on Recent Advances in Natural Language Processing (RANLP 2007)*, Borovets, Bulgaria, September 27–29 2007. [Online]. Available: http://rene-witte.net/next-generation-summarization

[14] B. Liu, M. Hu, and J. Cheng, "Opinion observer: analyzing and comparing opinions on the web," in *Proceedings of the 14th international conference on World Wide Web*. ACM, 2005, pp. 342–351.

[15] M. T. Boykoff and J. M. Boykoff, "Climate change and journalistic norms: A case-study of US mass-media coverage," *Geoforum*, vol. 38, no. 6, pp. 1190–1204, 2007, theme Issue: Geographies of Generosity. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0016718507000188

[16] B. Markner-Jäger, *Technical English for Geosciences: A Text / Work Book*. Springer, 2008. [Online]. Available: http://books.google.co.uk/books?id=dprCbs10Mn8C

[17] J. D. Novak and A. J. Cañas, "The theory underlying concept maps and how to construct them," Technical Report IHMC CmapTools 2006-01, Tech. Rep., 2006. [Online]. Available: http://cmap.ihmc.us/Publications/ResearchPapers/TheoryCmaps/TheoryUnderlyingConceptMaps.htm

ISSN 2395-8618

Journal Information and Instructions for Authors

I. Journal Information

Polibits is a half-yearly open-access research journal published since 1989 by the *Centro de Innovación y Desarrollo Tecnológico en Cómputo* (CIDETEC: Center of Innovation and Technological Development in Computing) of the *Instituto Politécnico Nacional* (IPN: National Polytechnic Institute), Mexico City, Mexico.

The journal has double-blind review procedure. It publishes papers in English and Spanish (with abstract in English). Publication has no cost for the authors.

A. Main Topics of Interest

The journal publishes research papers in all areas of computer science and computer engineering, with emphasis on applied research. The main topics of interest include, but are not limited to, the following:

- Artificial Intelligence
- Natural Language Processing
- Fuzzy Logic
- Computer Vision
- Multiagent Systems
- Bioinformatics
- Neural Networks
- Evolutionary Algorithms
- Knowledge Representation
- Expert Systems
- Intelligent Interfaces
- Multimedia and Virtual Reality
- Machine Learning
- Pattern Recognition
- Intelligent Tutoring Systems
- Semantic Web
- Robotics
- Geo-processing
- Database Systems
- Data Mining
- Software Engineering
- Web Design
- Compilers
- Formal Languages
- Operating Systems
- Distributed Systems
- Parallelism
- Real Time Systems
- Algorithm Theory
- Scientific Computing
- High-Performance Computing
- Networks and Connectivity
- Cryptography
- Informatics Security
- Digital Systems Design
- Digital Signal Processing
- Control Systems
- Virtual Instrumentation
- Computer Architectures

B. Indexing

The journal is listed in the list of excellence of the CONACYT (Mexican Ministry of Science) and indexed in the following international indices: Web of Science (via SciELO citation index), LatIndex, SciELO, Redalyc, Periódica, e-revistas, and Cabell's Directories.

There are currently only two Mexican computer science journals recognized by the CONACYT in its list of excellence, *Polibits* being one of them.

II. Instructions for Authors

A. Submission

Papers ready for peer review are received through the Web submission system on www.easychair.org/conferences/?conf=polibits1; see also updated information on the web page of the journal, www.cidetec.ipn.mx/polibits.

The papers can be written in English or Spanish. In case of Spanish, author names, abstract, and keywords must be provided in both Spanish and English; in recent issues of the journal you can find examples of how they are formatted.

The papers should be structures in a way traditional for scientific paper. Only full papers are reviewed; abstracts are not considered as submissions. The review procedure is double-blind. Therefore, papers should be submitted without names and affiliations of the authors and without any other data that reveal the authors' identity.

For review, a PDF file is to be submitted. In case of acceptance, the authors will need to upload the source code of the paper, either Microsoft Word or LaTeX with all supplementary files necessary for compilation. Upon acceptance notification, the authors receive further instructions on uploading the camera-ready source files.

Papers can be submitted at any moment; if accepted, the paper will be scheduled for inclusion in one of forthcoming issues, according to availability and the size of backlog.

See more detailed information at the website of the journal.

B. Format

The papers should be submitted in the format of the IEEE Transactions 8x11 2-column format, see http://www.ieee.org/publications_standards/publications/authors/author_templates.html. (while the journal uses this format for submissions, it is in no way affiliated with, or endorsed by, IEEE). The actual publication format differs from the one mentioned above; the papers will be adjusted by the editorial team.

There is no specific page limit: we welcome both short and long papers, provided that the quality and novelty of the paper adequately justifies its length. Usually the papers are between 10 and 20 pages; much shorter papers often do not offer sufficient detail to justify publication.

The editors keep the right to copyedit or modify the format and style of the final version of the paper if necessary.

See more detailed information at the website of the journal.