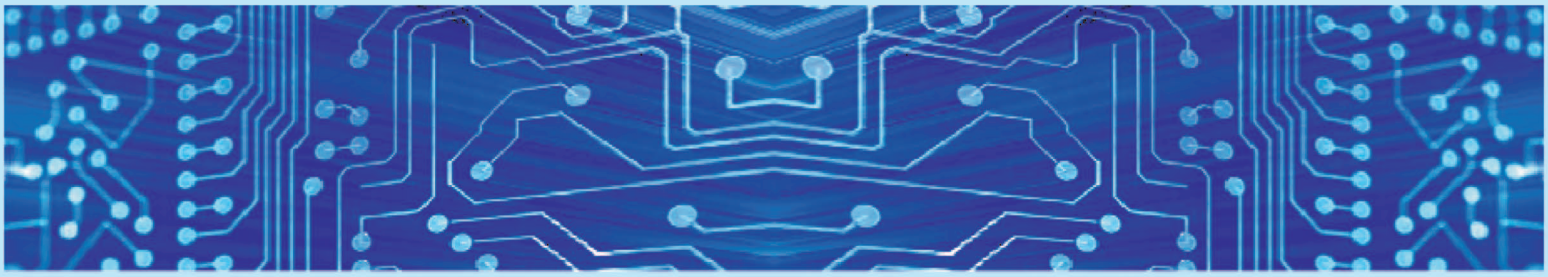
Princess Sumaya University for Technology
Since 1991 تأسست
PSUT
جامعة الأميرة سمية للتكنولوجيا

صندوق دعم البحث العلمي
Scientific Research Support Fund

Jordanian Journal of Computers and Information Technology

June 2020 VOLUME 06 NUMBER 02
ISSN 2415 - 1076 (Online)
ISSN 2413 - 9351 (Print)

www.jjcit.org jjcit@psut.edu.jo



An International Peer-Reviewed Scientific Journal
Financed by the Scientific Research Support Fund

Jordanian Journal of Computers and Information Technology (JJCIT)

The Jordanian Journal of Computers and Information Technology (JJCIT) is an international journal that publishes original, high-quality and cutting edge research papers on all aspects and technologies in ICT fields.

JJCIT is hosted by Princess Sumaya University for Technology (PSUT) and supported by the Scientific Research Support Fund in Jordan. Researchers have the right to read, print, distribute, search, download, copy or link to the full text of articles. JJCIT permits reproduction as long as the source is acknowledged.

Aims and Scope

The JJCIT aims to publish the most current developments in the form of original articles as well as review articles in all areas of Telecommunications, Computer Engineering and Information Technology and make them available to researchers worldwide. The JJCIT focuses on topics including, but not limited to: Computer Engineering & Communication Networks, Computer Science & Information Systems and Information Technology and Applications.

Indexing

JJCIT is indexed in:

ELSEVIER Scopus
DOAJ DIRECTORY OF OPEN ACCESS JOURNALS
crossref Text and Data Mining
OCLC WorldCat
BASE
ACADEMIA
Google scholar

Editorial Board Support Team

JJCIT Address

Website: www.jjcit.org
Email: jjcit@psut.edu.jo
Address: Princess Sumaya University for Technology, Khalil Saket Street, Al-Jubaiha
B.O. Box: 1438 Amman 11941 Jordan
Telephone: +962-6-5359949
Fax: +962-6-7295534

Editorial Board

International Advisory Board

“Opinions or views expressed in papers published in this journal are those of the author(s) and do not necessarily reflect those of the Editorial Board, the host university or the policy of the Scientific Research Support Fund”.

"ما ورد في هذه المجلة يعبر عن آراء الباحثين ولا يعكس بالضرورة آراء هيئة التحرير أو الجامعة أو سياسة صندوق دعم البحث العلمي".

ACCURATE AND FAST RECURRENT NEURAL NETWORK SOLUTION FOR THE AUTOMATIC DIACRITIZATION OF ARABIC TEXT

Gheith Abandah[1] and Asma Abdel-Karim[2]

(Received: 2-Sep.-2019, Revised: 27-Oct.-2019 and 21-Nov.-2019, Accepted: 16-Dec.-2019)

ABSTRACT

Arabic is mostly written now without its diacritics (short vowels). Adding these diacritics decreases reading ambiguity among other benefits. This work aims to develop a fast and accurate machine learning solution to diacritize Arabic text automatically. This paper uses long short-term memory (LSTM) recurrent neural networks to diacritize Arabic text. Intensive experiments are performed to evaluate proposed alternative design and data encoding options towards a fast and accurate solution. Our experiments involve investigating and handling problems in sequence lengths, proposing and evaluating alternative encodings of the diacritized output sequences and tuning and evaluating neural network options including architecture, network size and hyper-parameters. This paper recommends a solution that can be fast trained on a large dataset and uses four bidirectional LSTM layers to predict the diacritics of the input sequence of Arabic letters. This solution achieves a diacritization error rate of 2.46% on the LDC ATB3 dataset benchmark and 1.97% on the larger new Tashkeela dataset. This latter rate is 47% improvement over the best-published previous result.

KEYWORDS

Automatic diacritization, Arabic natural language processing, Sequence transcription, Arabic text, Recurrent neural networks, Long short-term memory, Bidirectional neural network.

1. INTRODUCTION

Automatic diacritization of Arabic text is one of the challenging and important tasks in Arabic Natural Language Processing (NLP). Arabic scripts consist of sequences of words written from right to left using two types of symbols: letters and diacritics. Letters should always be written, whereas diacritics can be omitted, resulting in partially diacritized or undiacritized texts [1]. Except for educated native speakers, lack of diacritization often causes incorrect pronunciation and consequently ambiguity in understanding the text. This is especially true for children and non-native speakers who lack sufficient mastery of the language grammar and lexicon.

Arabic text has two categories: *Classical Arabic* (CA) and *Modern Standard Arabic* (MSA) [1]. CA is the language of the Qur'an and old books and poems. MSA is the primary language used today in the media, education, news and formal speeches in Arabic-speaking countries. MSA is the modern form of CA and is based on it syntactically, morphologically and phonologically. As opposed to CA, most MSA texts are written with partial or no diacritization. In addition to these two categories, there are many informal spoken Arabic dialects. These dialects vary significantly geographically and socially and are neither standardized nor taught in schools. However, they are becoming more popular in writing Arabic texts on smart phones and over the internet [2]-[3].

The Arabic language has 28 letters and eight basic diacritics. Table 1 shows parts of the Unicode Block 0600-06FF that includes the Arabic letters and diacritics [2]. There are 36 variants of the 28 Arabic letters, which have Unicode hexadecimal codes 0621–063A and 0641–064A. These variants come from adding the six *Hamza* letters (ء، آ، أ، ؤ، إ، ئ), the *Teh Marbuta* (ة) and the *Alef Maksura* (ى) to the basic 28 letters. Arabic diacritics have Unicode codes 064B–0652. There are three types of Arabic diacritics: Vowel diacritics, Nunation diacritics and *Shadda*. Vowel diacritics include short vowels (*Fatha* ◌َ, *Damma* ◌ُ, *Kasra* ◌ِ) and the absence of vowel (*Sukun* ◌ْ). Nunation diacritics look like double versions of their corresponding short vowels (*Fathatan* ◌ً, *Dammatan* ◌ٌ, *Kasratan* ◌ٍ). The *Shadda* diacritic (◌ّ)

G. Abandah and A. Abdel-Karim are with Computer Engineering Department, The University of Jordan, Amman, Jordan. Emails: [1]abandah@ju.edu.jo and [2]a.abdelkarim@ju.edu.jo

implies doubling the letter it appears on [1].

Table 1. Unicode Arabic code block showing 36 letter variants and the basic eight diacritics.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
U+062x		ء	آ	أ	ؤ	إ	ئ	ا	ب	ة	ت	ث	ج	ح	خ	د
U+063x	ذ	ر	ز	س	ش	ص	ض	ط	ظ	ع	غ					
U+064x		ف	ق	ك	ل	م	ن	ه	و	ى	ي	◌ً	◌ٌ	◌ٍ	◌َ	◌ُ
U+065x	◌ِ	◌ّ	◌ْ													

In Arabic, words that have the same letters but different diacritics have different pronunciations and meanings. For example, the undiacritized word كتب has several meanings based on the way it is diacritized. If it is diacritized as كَتَبَ, it is pronounced "kataba" and means "wrote". However, if it is diacritized as كُتُب, then it is pronounced "kutub" and means "books". It can also be diacritized as كُتِبَ, pronounced "kutiba" and means "was written" indicating the past passive voice. A native reader can infer which diacritization form to use for a word based on the context. For example, for the statement كتب الطالب رسالة, the reader can infer that this is a verb-subject-object sentence "The student wrote a message" and hence the correct diacritization of the word كتب is كَتَبَ "kataba" [4].

The diacritization problem is even more complex when considering the *inflectional diacritics*. This type of diacritics is based on rules that inflect the word according to the context. Arabic words are inflected according to the word's tense, person, voice, gender, number, case and definiteness. The inflectional diacritics mostly occur on the last letter. For instance, the past verb كَتَبَ "kataba", which was not inflected in the last example, is inflected for first person as كَتَبْتُ "katabtu" and for feminine second person as كَتَبْتِ "katabti". In some cases, the inflectional diacritics do not appear on the last letter. For example, the composite word of noun and pronoun كتابه "his book" is diacritized كِتَابُهُ "kitabuhu" when it is a subject and كِتَابَهُ "kitabahu" when it is an object [5].

In order to correctly diacritize a sentence, the entire sentence context should be analyzed. Table 2 shows an example sentence where the diacritization of the first four words depends on the fifth word. In (a), the fifth word "كَثِيرَةٌ" "numerous" is adjective and implies that كتب is a noun and should be diacritized كُتُبُ "kutubu". In (b), the fifth word "الدَّرْسَ" "lesson" is noun and implies that كتب is a verb and should be diacritized كَتَبَ "kataba". This example also shows how the last letter diacritic of each word in the sentence differs based on the last word of the sentence. In fact, obtaining the correct diacritics of words' last letters, also known as *end cases*, is considered the most challenging part of automatic text diacritization. Therefore, automatic text diacritization requires an approach that takes into account both past and future contexts. Moreover, long-term context should be checked, since diacritics may depend on three or more distant words [4].

Table 2. Example of two different diacritizations of four words based on the fifth word.

Sentence	***Possible Diacritizations***	***Meaning in English***
كتب أحمد وعلي وعمر ______	كُتُبُ أَحْمَدَ وَعَلِيٍّ وَعُمَرَ كَثِيرَةٌ	(a) Books of Ahmed, Ali and Omar are numerous.
	كَتَبَ أَحْمَدُ وَعَلِيٌّ وَعُمَرُ الدَّرْسَ	(b) Ahmed, Ali and Omar wrote the lesson.

The objective of this work is to develop a fast and accurate model that uses *recurrent neural networks* (RNNs) to transcribe raw undiacritized sequences into fully diacritized sequences. We concentrate on networks that exploit long-term past and future contexts to make diacritics predictions and that can be trained using datasets in reasonable times. We train and test RNN models using two datasets: Linguistic Data Consortium's Arabic Treebank part 3 (LDC-ATB3) [6], which serves as an example of MSA and the cleaned subset of *Tashkeela* [7], which serves as an example of CA. As mentioned earlier, automatic diacritization of Arabic text provides help to children and non-native speakers in learning the language. In addition, it is an important step in text-to-speech (TTS) software and automatic speech recognition (ASR) engines.

Throughout our experiments, we explore and analyze the effect of tuning several network parameters, such as the number of network layers and using dropout, on the accuracy and execution time of the tested models. We experiment alternative approaches to handle problems in sequence lengths and propose wrapping of sequences to solve the problem. We also use multiple encoding methods for the diacritized output sequences and propose two new encoding methods. In addition, we experiment with three network architectures: unidirectional *long short-term memory* (LSTM), *bidirectional* LSTM and *encoder/decoder* LSTM networks.

The rest of this paper is organized as follows. In the next section, we provide a review of automatic Arabic diacritization systems proposed in the literature. Section 3 provides background information of the RNN models we use in this work. Section 4 describes our experimental setup. Section 5 presents and discusses the results of our experiments. Section 6 compares our best results with the results of previous best-performing models and analyzes the errors. Finally, Section 7 gives the conclusions.

2. Literature Review

Systems developed for automatic diacritization of Arabic text can be classified into three categories: *rule-based* systems, *statistical* systems and *hybrid* systems.

2.1 Rule-based Systems

Rule-based approaches require defining a set of well-formed rules that exploit human knowledge in the form of morphological analyzers, dictionaries and grammar modules. Although rule-based approaches solve the problem with acceptable results, they rely on linguistic knowledge or parsing tools and require rules to be continuously maintained and updated [8]-[9].

2.2 Statistical Systems

Statistical approaches, on the other hand, predict the probable diacritics for a sequence of characters without the need for language-specific knowledge or parsing tools. Instead, they require a large corpus of diacritized text. Machine learning statistical methods that have been applied to Arabic text diacritization include hidden Markov models (HMMs), n-grams, finite state transducers (FSTs) and more recently RNNs [8].

Gal [10] used an HMM to restore Arabic diacritics with the Holy Quran as a corpus. His system restores only short vowels and correctly diacritizes 89% of the words in the test set. Elshafei et al. [11] proposed a similar approach that uses an HMM for modeling and Viterbi search algorithm to find the most optimal diacritics of a sentence. Their training data was taken from multiple knowledge domains and the tests used randomly picked verses from the Quran. They achieved a 4.1% diacritization error rate. Refer to Subsection 4.5 for the definition of diacritization error rates.

Hifny [12] proposed an automatic diacritization system that combines dynamic programming (DP) with n-gram language model and smoothing. He used n-gram language modelling to assign scores to possible diacritized word sequences. Dynamic programming is then used to search for the most likely sequence. Different smoothing algorithms are tested to solve the problem of unseen n-grams. The author used the Tashkeela dataset [13] for training and testing his model with a corpus of 5.25 million words for training and a testing set of 1.9 million words. His approach achieved a word error rate of 3.4% when end cases' are excluded and 8.9% when these cases are included. This is due to the difficulty of retrieving end cases diacritics as outlined in the previous section. Definition of word error rate is also provided in subsection 4.5.

Azim et al. [14] proposed a statistical approach that uses weighted combination of two diacritizers: one is text-based and the other is speech-based. The system uses a correctly vocalized speech of the text to complement and correct errors generated by the text-based model. The text-based diacritizer is modelled by *conditional random fields* (CRFs) and the speech-based diacritizer is modelled by HMM. Using LDC ATB3, their approach achieves very accurate diacritization and word error rates of 1.5% and 4.9%, respectively. However, their system requires the availability of an acoustic signal that corresponds to the raw text data.

2.3 Hybrid Systems

Most current systems use hybrid approaches that make use of language-specific rules to guide statistical techniques. Vergyri and Kirchhof [15] explored multiple combinations of acoustic information with morphological and contextual sources. In their experiments, they used two corpora: the Foreign Broadcast Information Service (FBIS) corpus of MSA speech and the LDC Call Home Egyptian Colloquial Arabic (ECA) corpus. Without modelling the *Shadda* diacritic, they achieved diacritization and word error rates of 11.5% and 27.3%, respectively. Nelken et al. [16] proposed a hybrid model that uses a cascade of finite state transducers with integrated word-based model, letter-based model and morphological model. The diacritization and word error rates of their model using LDC Arabic Treebank of diacritized news stories (Part 2) are 12.8% and 23.6%, respectively.

Zitouni et al. [17] proposed an approach based on maximum entropy framework that learns the correlation between several input features and the output diacritics. These features include lexical features, segment-based features and part-of-speech (POS) features. They trained and tested their model using LDC ATB3. They provided an in-detail description of their usage of the LDC ATB3 and produced a clearly defined split of the dataset into training and testing subsets. This split established LDC ATB3 as a benchmark in this area and allowed for reproduction of results and accurate comparison with subsequent techniques. Their approach achieves diacritization and word error rates of 5.5% and 18.0%, respectively.

In [5], Habash and Rambow extended their morphological analysis and disambiguation of Arabic system (MADA) such that it consults the Buckwalter Arabic Morphological Analyzer (BAMA) to get a list of all potential analysis of a word. Fourteen Support Vector Machine (SVM) predictors are then used to narrow this list to a smaller one. Finally, n-gram language models are used to select one solution from the narrowed list. They trained and tested their approach using LDC ATB3 as proposed by Zitouni et al. [17]. Their approach achieves diacritization and word error rates of 4.8% and 14.9%, respectively.

Rashwan et al. [18] introduced a system that uses two stochastic layers in order to perform automatic diacritization. The first layer is an un-factorized layer that diacritizes letters by searching a dictionary that was built offline. It retrieves all diacritized forms of the word if it is found. The most likely sequence is then selected using the n-gram probability estimation and A* lattice search. The second layer factorizes words that were not diacritized in the first layer into their morphological components (prefix, root, pattern and suffix). N-gram probability estimation and A* lattice search are also used in this layer to select the most likely diacritization from the generated factorizations. The reported diacritization and word error rates of their approach using LDC ATB3 are 3.8% and 12.5%, respectively.

The hybrid system developed by Said et al. [19] includes an automatic corrector, rule-based and statistical morphological analyzers, a POS tagger and an out-of-vocabulary diacritizer. Their rule-based analyzer was formed based on comprehensive lexicon and handcrafted rules. The statistical analyzer was trained using LDC ATB3. Given an input word, these analyzers produce a lattice of diacritized forms. The POS tagger disambiguates this lattice and selects the most likely diacritized form for the word using HMM and Viterbi algorithm. Their approach achieved diacritization and word error rates of 3.6% and 11.4%, respectively.

Our previous work in [4] was the first to use RNN to solve the diacritization problem as a sequence transcription problem. More specifically, we proposed, trained and tested a bidirectional LSTM network that takes as an input raw undiacritized sequences and transcribes them into diacritized sequences. Our approach did not apply lexical, morphological or syntactical analysis prior to or in line with the data training. We used error-correction techniques to post-process the output of the network. We used LDC ATB3, the simple version of the holy Quran and ten books drawn from the Tashkeela dataset [13]. We achieved state-of-the-art performance with diacritization and word error rates of 2.09% and 5.82%, respectively, for Tashkeela and 2.72% and 9.07%, respectively, for ATB3. A follow up work shows that the RNN accuracy can even be slightly improved when a morphological and syntactical analyzer preprocesses the RNN input [20].

Rashwan et al. [21] proposed a system that consists of two frameworks: a deep learning framework that uses the confused sub-set resolution (CSR) method to improve the classification accuracy and an Arabic part-of-speech (PoS) tagging using deep neural networks. Their system achieves an accuracy of 97% using LDC ATB3. In [7], Fadel et al. compared the performance of some publicly available rule-based systems with the neural-based approach Shakkala [22] using their cleaned subset of Tashkeela. Shakkala outperformed the best performing rule-based approaches, mainly Mishkal [23] and Harakat, with diacritization and word error rates of 3.73% and 11.19%, respectively.

More recently, Mubarak et al. [24] implemented a sequence-to-sequence model using an encoder-decoder LSTM RNN with attention mechanism. They used sliding window to divide sequences into fixed lengths. The most likely diacritic form of a word is selected using n-gram probability estimation. They trained their model using 4.5 million tokens and tested it using the freely available WikiNews corpus of 18,300 words [25]. They do not identify their training data or refer to its source. Their best reported results are 1.21% and 4.49% diacritization and word error rates, respectively. Although these are the best results reported so far, we do not include them in our comparison, because they are not generated using the same datasets commonly used in this domain.

In this paper, we build on our previous work in [4] by implementing our model using state-of-the-art tools. Moreover, we perform intensive experiments that explore alternative implementation options (e.g., network architecture, optimization techniques and number of layers) and data preparation options (e.g., encoding methods and handling sequence lengths) towards a faster and more accurate model.

3. Sequence Transcription

Sequence Transcription is the process of translating an input sequence into the corresponding output sequence of a different type. This includes language translation, voice recognition and diacritizing Arabic texts. Recurrent neural networks have proved to perform best on sequence transcription [26]. This is due to their ability to preserve correlations between data points in the sequence, as their hidden states are functions of all previous states with respect to time [27].

3.1 Recurrent Neural Networks

Given a sequence of inputs $(x_1, x_2, \dots, x_T)$, a standard RNN computes a sequence of outputs $(y_1, y_2, \dots, y_T)$ based on the computation of a sequence of hidden vectors by iterating the following equations from $t = 1$ to T:

$$h_t = \sigma(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \tag{1}$$

$$y_t = W_{hy}h_t + b_y \tag{2}$$

where W terms denote weight matrices and b terms denote bias matrices. For computing each hidden state, there are two sets of weights: one for the inputs x_t and one for the previous hidden state h_{t-1} [28]. A sigmoid function σ is normally used as the activation function for basic RNNs [26]. RNNs can be used to solve four types of sequence transcription problems according to the lengths of the input and the output [28]. These types are shown in Figure 1.

The first RNN type, shown in Figure 2a, takes an input sequence and produces an output sequence of the same length. These networks are referred to as *one-to-one* networks. The second type is the *sequence-to-vector* network, where input sequences are transcribed into one final output by ignoring all previous outputs. The third type is the *vector-to-sequence* network, where one input vector is used to produce an output sequence. The fourth type is the general *sequence-to-sequence* network, where the output sequence is generally not of the same length as the input sequence. This type is often implemented using the *encoder-decoder* architecture.

Given that automatic diacritization of Arabic text is a sequence-to-sequence problem, both the one-to-one and the encoder-decoder networks can be used to solve the problem. In this work, we implement both types using multiple encoding methods for the output sequences. In this problem, the encoder-decoder approach is implemented with output sequences (that include letters and diacritics) that are longer than the input undiacritized sequences and hence is considered a *one-to-many* sequence-to-sequence transcription.

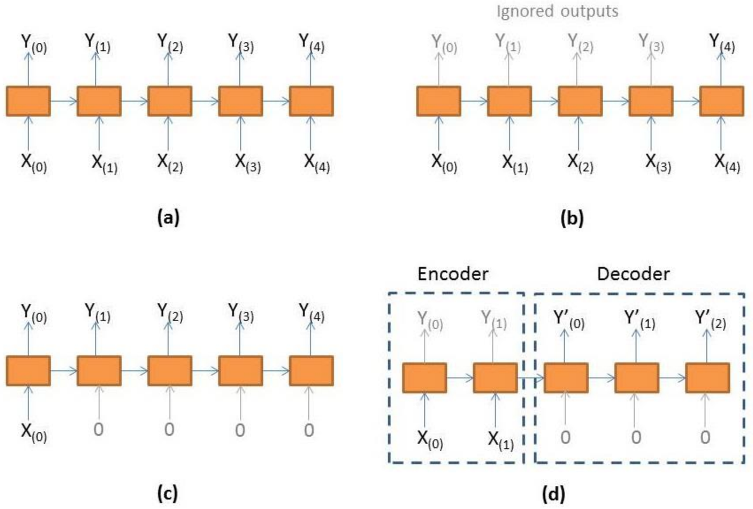


Figure 1. RNN types based on lengths of the input and the output: (a) one-to-one sequence-to-sequence, (b) sequence-to-vector, (c) vector-to-sequence and (d) general sequence-to-sequence.

3.2 Long Short-Term Memory Cells

The basic recurrent networks described above consist of memory cells that can store representation of recent inputs only. Hence, they have a short-term memory that results in slowly changing weights [29]. LSTM networks, on the other hand, which use purpose-built memory cells, are capable of learning from long-term contexts [27]. Each memory cell has an *input* gate, a *forget* gate, an *output* gate and a *cell activation* unit. These units are represented by the vectors i, f, o *and* c, respectively, which are of the same size as the hidden vector h_t. The following equations show how the hidden vector's activation function for LSTM is a composite function that results from computing the aforementioned vectors.

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (3)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (4)$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (5)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (6)$$

$$h_t = o_t \tanh(c_t) \quad (7)$$

Notice that these vectors depend on the layer input x_t and the previous states: short-term state h_{t-1} and long-term state c_{t-1}.

3.3 Encoder/Decoder Networks

Encoder-decoder RNNs are designed such that the network consists of two RNNs. The first RNN represents the encoder which maps an input sequence into a fixed-length vector acting as a sequence-to-vector network. More specifically, this vector is a summary of the input sequence. The second RNN is the decoder which maps the encoder vector into an output sequence forming a vector-to-sequence network. The two networks are jointly trained to maximize the probability of an output sequence given an input sequence. As introduced above, this architecture allows mapping input sequences to output sequences of different lengths [30].

3.4 Bidirectional RNNs

Conventional unidirectional RNNs can make use only of previous context. However, many sequence transcription problems, including diacritization, require exploiting future context as well. Bidirectional RNN layers achieve this by comprising two adjacent unidirectional networks in each layer. One network is trained by presenting the sequence in the forward direction and the other is trained by presenting it in the backward direction. The output is a function of both networks and, consequently, exploits past and future contexts. Specifically, the forward hidden vector is computed by iterating in the positive time direction (i.e., from $t = 1$ to T), while the backward hidden vector is computed by iterating in the negative time direction (i.e., from $t = T$ to 1) [31]. Both vectors are used to update the output vector y_t, as specified in the following equations:

$$\vec{h}_t = \sigma\left(W_{x\vec{h}} x_t + W_{\vec{h}\vec{h}} \vec{h}_{t-1} + b_{\vec{h}}\right) \tag{8}$$

$$\overleftarrow{h}_t = \sigma\left(W_{x\overleftarrow{h}} x_t + W_{\overleftarrow{h}\overleftarrow{h}} \overleftarrow{h}_{t-1} + b_{\overleftarrow{h}}\right) \tag{9}$$

$$y_t = W_{\vec{h}y} \vec{h}_t + W_{\overleftarrow{h}y} \overleftarrow{h}_t + b_o \tag{10}$$

3.5 Deep RNNs

RNNs are made even more powerful by stacking multiple layers on top of each other, forming a deep RNN. Deep networks are necessary to solve complex transcription functions. In such architectures, the output sequence of one layer acts as the input sequence for the next layer. Assuming that the same hidden function σ is used for all N layers in the stack, the hidden vectors h^n are computed by iterating from $n = 1$ to N and from $t = 1$ to T, as shown in Equation 11, where $h^0 = x$. The network final output y_t is computed according to Equation 12.

$$h_t^n = \sigma(W_{h^{n-1}h^n} h_t^{n-1} + W_{h^n h^n} h_{t-1}^n + b_h^n) \tag{11}$$

$$y_t = W_{h^N y} h_t^N + b_y \tag{12}$$

4. Experimental Setup

We use an experimental setup similar to that used in our previous work in [4]. During the training process, both input undiacritized sequences and target diacritized sequences are presented to the model after encoding. The model is tested by applying undiacritized sequences to its input and comparing the transcribed output sequences with the correctly diacritized sequences. Our final reported results are obtained after post-processing is performed on the predicted output sequences to correct some transcription errors. Post-processing techniques include *Sukun* correction, *Fatha* correction and dictionary correction, which were proposed and discussed in our previous work in [4]. The processing and memory specifications of the platform on which our experiments were performed are shown in Table 3. The following subsections describe other aspects of our experimental setup.

Table 3. Processing and memory specifications of the experimental platform.

CPU	Intel Core i7-6700 @ 3.4 GHz, 4 cores (8 threads), 8 MB cache
GPU	Nvidia GeForce RTX 2080 @ 2.1 GHz, 2944 CUDA cores, 8 GB memory
Memory	32 GB DDR4-SDRAM @ 1066MHz

4.1 Data

Our experimental data consists mainly of text from the Linguistic Data Consortium’s (LDC) Arabic Treebank (LDC2010T08) [6] and the cleaned subset of Tashkeela corpus extracted in [7]. More specifically, we use the LDC’s Arabic Treebank Part 3 (ATB3) v3.2, which consists of 599 distinct newswire stories from the Lebanese publication An Nahar. Text in this dataset is an example of the modern standard Arabic (MSA). We split this dataset, as proposed by Zitouni et al. [17], such that the first 509 newswire stories, in chronological order, are used for training the model and the last 90 stories are used for validation and testing. This accounts for 22,170 sequences for training and 3,857 sequences for validation.

The used Tashkeela dataset includes 55K lines randomly chosen by Fadel et al. [7] from the classical Arabic (CA) and Holy Quran datasets. The provided dataset is a processed subset of the original datasets with some file formatting errors removed and many diacritization issues fixed. The dataset is split into 50K lines for training, 2,500 lines for validation and 2,500 lines for testing. Table 4 shows size statistics of these two datasets: word count, sequence count, average letters per word and average words per sequence.

Table 4. Datasets' size statistics.

Dataset	*Word Count*	*Sequence Count*	*Letters per Word*	*Words per Sequence*
LDC ATB3	305 K	26,027	4.6	11.3
Tashkeela	2,312 K	55 K	4.0	42.1

Table 5 provides statistics of diacritics usage in these datasets in terms of the percentage of letters without diacritics, with one diacritic and with two diacritics.

Table 5. Datasets' diacritics usage statistics.

Dataset	*No Diacritics*	*One Diacritic*	*Two Diacritics*
LDC ATB3	39.8%	54.8%	5.4%
Tashkeela	17.8%	77.2%	5.0%

Tashkeela is larger than ATB3 in number of sequences and sequence lengths. However, both datasets have close average number of letters per word, which is a property of the Arabic language [4]. Tashkeela has smaller percentage of characters with no diacritics compared to ATB3. This is due to the extraction process conducted in [7] of the used Tashkeela subset which ensured diacritics to characters rate greater than 80%.

One of the aspects that we address in this study is the effect of variation in sequence lengths and having very long sequences on both the execution time and the accuracy. The maximum sequence length for ATB3 is 695 letters, whereas Tashkeela has a maximum sequence length of 7,542 letters. Nevertheless, both datasets have small percentages of very long sequences. Figure 2 shows the cumulative distribution function (CDF) of the sequence length for ATB3 and Tashkeela datasets. Only 1% of ATB3's sequences are longer than 233 characters, whereas only 1% of Tashkeela's sequences are longer than 1,194 characters.

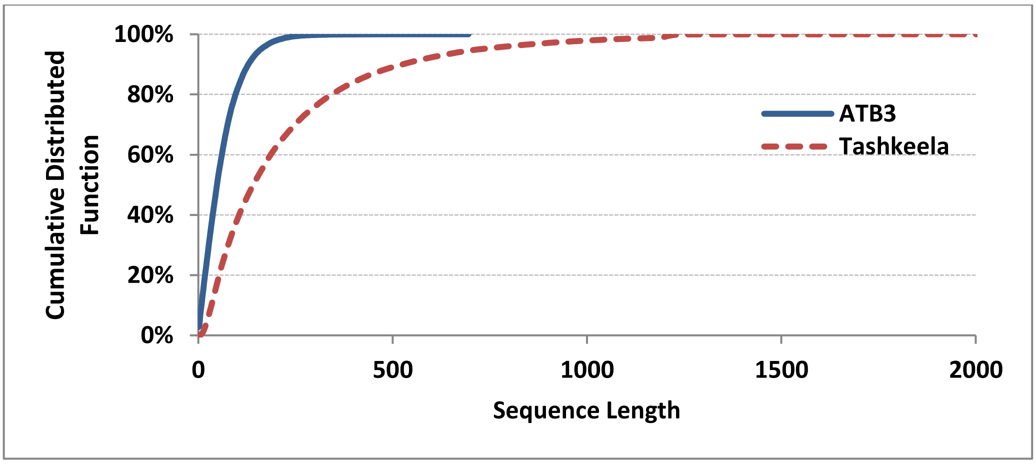


Figure 2. Cumulative distribution function (CDF) of sequence length of ATB3 and Tashkeela datasets.

4.2 Data Preparation

The Tashkeela dataset was cleaned by Fadel et al. [7]. Their cleaning process included solving some

diacritization issues, such as fixing misplaced diacritics and removing the first diacritic in cases of letters with multiple diacritics. They also prepared the dataset by removing English letters, separating numbers from words by adding whitespaces and removing multiple whitespaces. In addition, they performed some file formatting, such as removing tags from HTML files and removing URLs.

In order to perform machine learning using datasets larger than the computer memory, we converted both datasets into TensorFlow records (TFRecords) [32]. The TFRecord format is a format for storing data on the disk and allows reading huge data efficiently during training and testing. Our TFRecords consist of sequences, each sequence consists of tokens and each token is one-hot encoded in a vector. Storing these datasets, in their original dense format, results in consuming very large disc spaces. For

ATB3, the training TFRecord file is 3.5 GB and is 102 GB for Tashkeela. To reduce this space, we considered skipping sequences longer than 300 characters. This reduces ATB3 training TFrecord file to 1.5 GB and to 4 GB for Tashkeela. However, skipping long sequences reduces the number of sequences used in training.

Therefore, we experimented with using a sparse format that exploits the fact that the one-hot encoding gives vectors that mostly consist of 0’s. The size when using the sparse format depends on the number of tokens, but unlike the dense format, does not depend on the maximum sequence length. Moreover, for the dense case when sequences longer than 300 characters are not skipped, we wrap long sequences to a maximum of 400 characters (see Section 5.1 for more detail).

Figure 3 shows ATB3 and Tashkeela TFrecord training file sizes, in logarithmic scale, for the dense and sparse formats both with and without skipping of sequences longer than 300 characters. Sparse format is much smaller than the dense format for both datasets even when sequences longer than 300 characters are not skipped. Hence, we use sparse files to store the entire datasets while maintaining reasonable usage of the disc space. Moreover, the execution time when using the sparse format is comparable to that of the dense format. The overhead of converting sparse format back into dense matrices is hidden by the time saved when avoiding the long disc access time of the large dense files.

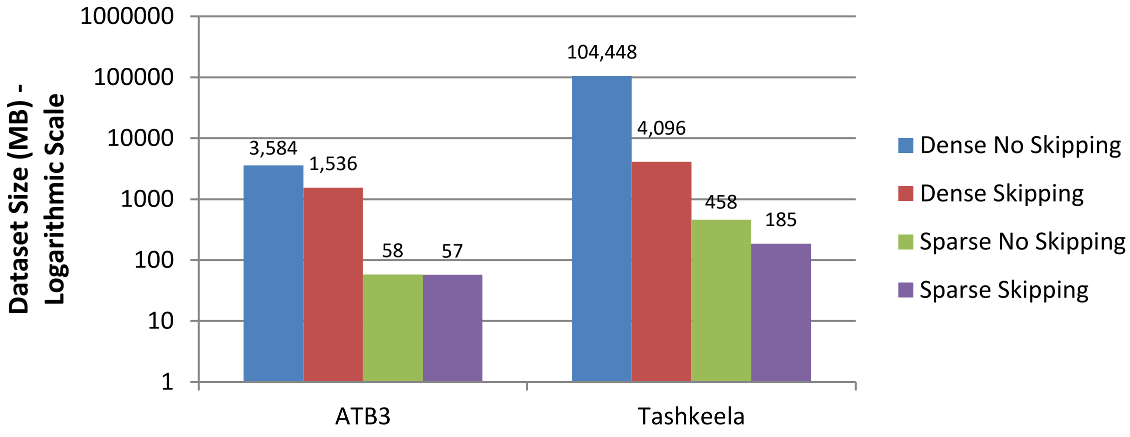


Figure 3. Training dataset sizes when using dense and sparse TFRecords with and without skipping of sequences longer than 300 characters.

4.3 Data Encoding

This subsection describes how input and output sequences are encoded. Input undiacritized sequences are obtained by removing diacritics from target diacritized sequences. Since they consist of letters only, input sequences are encoded using the Unicode representations of their letters. For example, the undiacritized word “ثم” is encoded as “062B 0645”: the Unicodes of the letters ث and م, respectively.

For diacritized target sequences, we experiment with four encoding methods as described below. Table 6 shows the eight main Arabic diacritics and their shapes, sounds, hexadecimal Unicode numbers and the binary bit codes used to encode them in this work. Each letter in Arabic may have no diacritics, one diacritic or two diacritics. When a letter has two diacritics, one of these letters must be *Shadda*. The diacritized sequence encoding methods used in this work are a one-to-many encoding method and three one-to-one encoding methods.

1) One-to-many encoding

In one-to-many encoding, we use separate symbols for the letter and its diacritics as in Unicode. The Unicode representations of the letter’s diacritic(s) follow the letter Unicode representation. For example, the diacritized word “ثُمَّ” is encoded as “062B 064F 0645 0651 064E”. Therefore, diacritized target sequences are usually longer than undiacritized input sequences. In this work, we use one-to-many encoding with the encoder-decoder network.

Table 6. The main eight Arabic diacritics with their shapes, sounds, hexadecimal Unicode numbers and used binary bit codes.

Name	*Shape*	*Sound*	*Unicode*	*Bit code*
Fathatan	◌ً	/an/	064B	0001
Dammatan	◌ٌ	/un/	064C	0010
Kasratan	◌ٍ	/in/	064D	0011
Fatha	◌َ	/a/	064E	0100
Damma	◌ُ	/u/	064F	0101
Kasra	◌ِ	/i/	0650	0110
Sukun	◌ْ	None	0652	0111
Shadda	◌ّ	Doubling	0651	1000

2) Many classes, one-to-one encoding

In one-to-one encoding, one symbol is used to encode each letter with its diacritic and hence input and target sequences have the same length. The first method is the encoding method used in our previous work [4]. Using this method, each diacritized letter is encoded in a symbol that results from combining the letter code with its diacritic(s) bit code(s), as shown in Equation 13. Each letter is encoded into a unique code L that is formed by clearing the most significant byte of the letter's Unicode number l, which is 0x06 for all Arabic letters. Then, the masked code is shifted four-bit positions to the left and ORed with the bit code of the letter diacritic d_1 if it has one diacritic or the bit codes of its two diacritics d_1 and d_2 if it has two diacritics. Notice that this encoding method gives many output classes in the order of the number of letters times the number of diacritics.

$$L = \begin{cases} (l \wedge \text{0x00ff} \ll 4), & \text{no diacritic} \\ (l \wedge \text{0x00ff} \ll 4) \vee d_1, & \text{one diacritic} \\ (l \wedge \text{0x00ff} \ll 4) \vee d_1 \vee d_2, & \text{two diacritics} \end{cases} \tag{13}$$

For example, in order to encode the letter مَّ of the word ثُمَّ, the Unicode of the letter م which is 0645 is masked into 0045. Then, the code is shifted 4 bits into 0450 and the combined bit code of *Fatha* and *Shadda* (0100 ∨ 1000 = 1100 = C) is inserted in the least significant four bits of the letter code (0450) to form the code 045C.

3) Diacritics only, one-to-one encoding

In this work, we propose and test two other one-to-one encoding methods. In the first method, each diacritized letter is encoded using its diacritics only. This encoding scheme relies on the fact that letters in the undiacritized input sequence do not change in the target diacritized sequence except for adding diacritics to them. This has the advantage of limiting the number of possible output classes to the number of possible diacritics and hence simplifies the output layer. Equation 14 shows the way in which a unique code *L* is formed using this encoding scheme using the diacritics bit code(s) without involving the letter unicode representation. For example, the letter مَّ of the word ثُمَّ is encoded using its diacritics bit codes only which is C (0100 ∨ 1000 = 1100 = C).

$$L = \begin{cases} 0, & \text{no diacritic} \\ d_1, & \text{one diacritic} \\ d_1 \vee d_2, & \text{two diacritics} \end{cases} \tag{14}$$

4) Multiple label, one-to-one encoding

The second one-to-one encoding method that we propose in this work assumes that each bit in the code represents a label that contributes to the diacritization of the letter. Table 7 illustrates the labels assigned to each bit position in this encoding method. A value of 1 in a bit position indicates that the corresponding diacritic is present. For example, the letter مَّ of the word ثُمَّ is encoded using this method by placing 1s in the *Shadda* label (bit position 5) and the *Fatha* label (bit position 1) to form the binary code 100010 (hexadecimal 22).

Table 7. Labels assigned to bits in multiple label encoding.

Bit Position	***5***	***4***	***3***	***2***	***1***	***0***
Label	*Shadda*	*Nunation*	*Kasra*	*Damma*	*Fatha*	*Sukun*

To summarize, Table 8 shows the example word ثُمَّ encoded using these four encoding methods.

Table 8. Encoding the diacritized word ثُمَّ using one-to-many, many classes, diacritics only and multiple label encoding.

Encoding Method	***Encoding of letters***		***Word Encoding***
	ثُ	مَّ	
One-to-many	062B 064F	0645 0651 064E	062B 064F 0645 0651 064E
Many Classes	02B5	045C	02B5 045C
Diacritics Only	5	C	5 C
Multiple Labels	04	22	04 22

4.4 Base Model

We use Keras (Python deep learning library) with TensorFlow at the backend to develop our machine learning models [32]. This combination implements the state-of-the-art algorithms in deep machine learning. Our baseline model is an LSTM RNN with two bidirectional layers and 256 cells per layer preceded by a masking layer and followed by a fully-connected output layer. This model uses *softmax* as the activation of the output layer, the *RMSprop* optimizer in training, categorical cross entropy as the loss function and a batch size of 64 sequences [28]. Figure 4 shows the core code of this model.

```
model = Sequential()
model.add(Masking(mask_value= 0, input_shape=(seq_len, num_inp_tokens)))
model.add(Bidirectional(LSTM(256, return_sequences=True), merge_mode='concat'))
model.add(Bidirectional(LSTM(256, return_sequences=True), merge_mode='concat'))
model.add(TimeDistributed(Dense(num_tar_tokens, activation='softmax')))
model.compile(loss='categorical_crossentropy', optimizer='rmsprop', metrics=['acc'])
```

Figure 4. Python Keras core code of the base model.

4.5 Evaluation Metrics

We evaluate models in terms of execution time required to train the model and the accuracy of the model in diacritizing the input sequences. Throughout our experiments, we evaluate the accuracy of multiple designs and data handling/encoding options using the *diacritization error rate*. DER is the percentage of characters with incorrectly predicted diacritics. For all experiments, we use the original DER definition where punctuation marks and numbers are counted [17]. We also report the *word error rate* of our best performing model. WER is the percentage of incorrectly diacritized words. A word is considered incorrectly diacritized if it has at least one incorrectly diacritized letter.

For the experiment that evaluates alternative network architectures, we use the accuracy as the evaluation metric, because DER and WER cannot be obtained for the encoder/decoder network, as explained in Section 5.3.

5. Experiments and Results

The following subsections present the experiments and their results.

5.1 Handling Sequence Lengths

As discussed earlier, both datasets have high variation in their sequence lengths. We conducted three experiments to find the best approach to handle this variation. Figure 5 shows the training time required and the diacritization error rate obtained for the three experiments for both datasets. We first included all sequences for the ATB3 dataset (i.e., maximum sequence length is 695) and sequences not exceeding 1,260 characters for Tashkeela (i.e., this accounts for 99.94% of Tashkeela sequences). Then, we included only sequences with a maximum of 300 characters for both datasets. Finally, we experimented including all sequences, but wrapping long sequences to have a maximum of 400 characters per input. The network input is arranged in tensors (dense matrices) with a width that equals the longest sequence, e.g., 695, 300 and 400, respectively for the three ATB3 experiments.

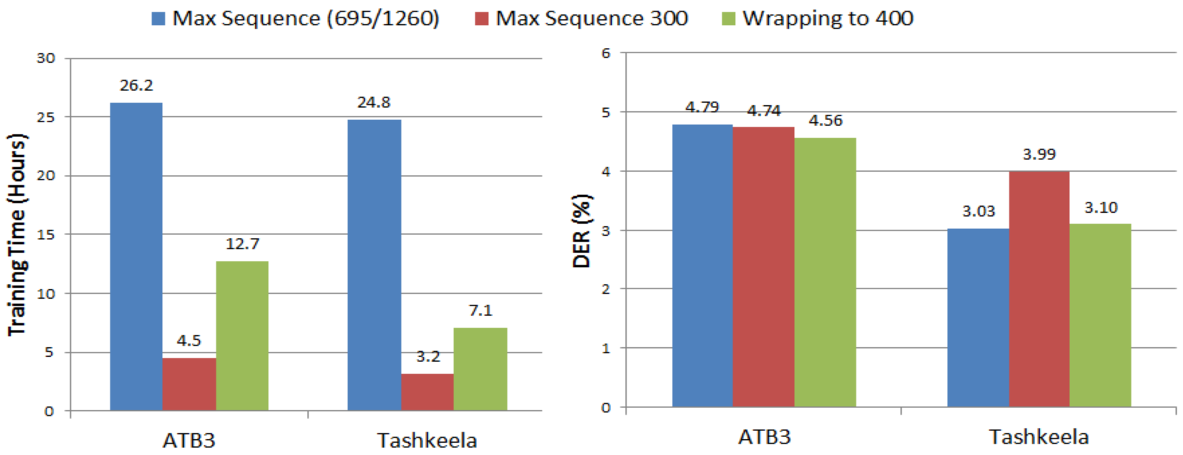


Figure 5. Training time and DER for ATB3 and Tashkeela datasets when including: (a) Maximum sequence length (695 for ATB3 and 1260 for Tashkeela), (b) Sequences not longer than 300 and (c) All sequences and wrapping long sequences to a maximum of 400 characters.

As expected, the execution time of the long sequences experiment is the highest: 26.2 hours for ATB3 and 24.8 hours for Tashkeela. However, including only sequences shorter that 300 adversely affects Tashkeela accuracy (a DER of 3.99%). In fact, the best DER for Tashkeela is obtained when all sequences are included (3.03% DER). Wrapping sequences to 400 characters is a good compromise; it gives reasonable training times of 12.7 hours for ATB3 and 7.1 hours for Tashkeela. At the same time, obtained DER values using wrapping is very close to the best DER. These are 4.56% for ATB3 and 3.10% for Tashkeela. We use this third method in the following experiments. Notice that Tashkeela achieves lower DER than ATB3. Tashkeela always achieves better accuracy than ATB3, because its

training set is larger and its diacritized letters ratio is higher.

5.2 Data Encoding Methods

Figure 6 shows the results of experiments using the three one-to-one target sequences encoding methods: many classes encoding, diacritics only encoding and multiple label encoding. For both datasets, encoding using diacritics only achieves the best results with DER of 4.83% for ATB3 and 3.10% for Tashkeela. The new diacritics only encoding is consistently better than the many classes encoding. It simplifies the network function from predicting letters and diacritics to predicting the diacritics only. We first had high expectations for multiple label encoding, because it exposes the contextual significance of the diacritics. However, this manual split of diacritics to multiple labels turned out to be less efficient than machine learning with one-hot encoding.

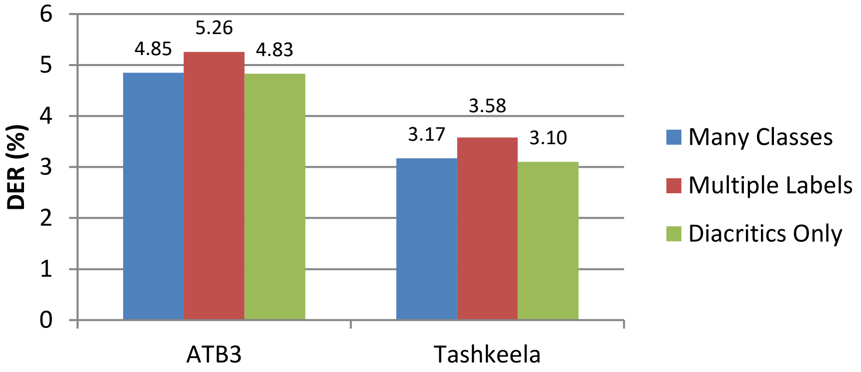


Figure 6. DER for ATB3 and Tashkeela using one-to-one encoding methods: many classes, multiple labels and diacritics only.

5.3 Network Architecture

This subsection presents the results of experimenting with three network architectures. The tested architectures are the encoder/decoder model, unidirectional LSTM and bidirectional LSTM. The tested encoder/decoder network consists of two encoder layers and two decoder layers. We limit the sequences lengths to 100 characters for this model, because longer sequences do not converge to useful output. Notice that for this set of experiments, we use the validation accuracy to evaluate the three network architectures, because the DER cannot be calculated for the encoder/decoder network. This network often outputs sequences that are not only wrong in diacritics, but also are wrong in the letter sequence output, making the DER calculation impossible. Figure 7 shows the validation accuracy of the three tested architectures. The encoder/decoder architecture has the worst validation accuracy among the three architectures. Moreover, as expected, unidirectional LSTM has inferior performance compared to bidirectional LSTM, since diacritization of a word depends on future context as well as on past context. Bidirectional LSTM is better than unidirectional LSTM by at least 11%.

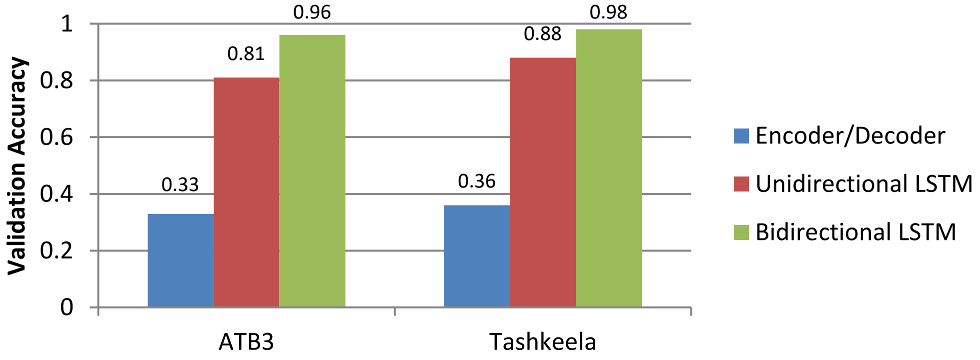


Figure 7. Accuracy of the validation set for ATB3 and Tashkeela using three network architectures.

Although there is a recent work that used encoder/decoder architecture to add the diacritics [24], we do not recommend it. Bidirectional LSTM gives better results without the trouble of employing sophisticated techniques, such as sliding windows, voting and attention.

5.4 Network Size

Figure 8 shows the results of changing the network size by varying the number of bidirectional LSTM layers. As expected, increasing the number of layers increases the time required to train the network. However, a deeper network with four layers achieves better accuracy than fewer layers; a DER of 4.19% for ATB3 and 2.83% for Tashkeela.

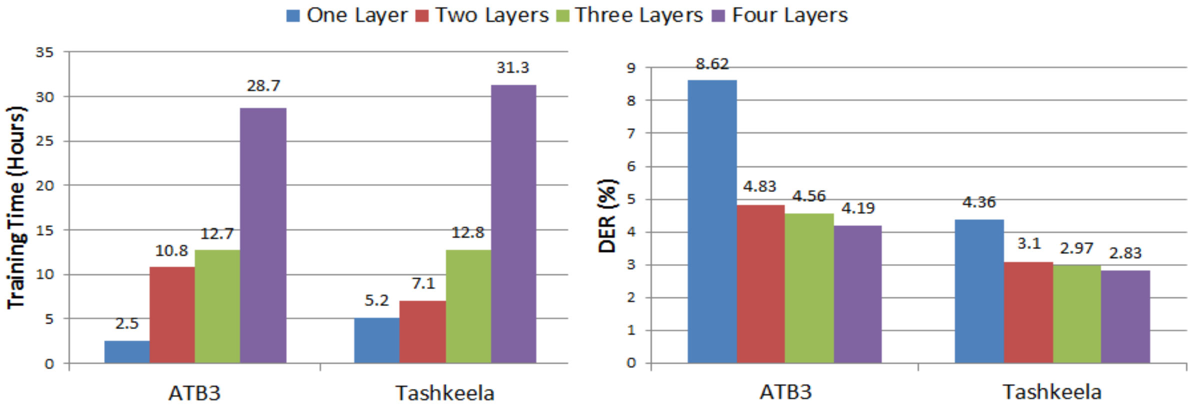


Figure 8. Training time and DER when varying network size from one layer to four layers.

5.5 Influence of Dropout

We used dropout regularization to overcome training overfitting [28]. We used grid search to find the best dropout option. Best results are obtained when a dropout rate of 0.1 is used for the layer input and a dropout rate of 0.3 is used for the recurrent state. Figure 9 shows the result of applying this dropout to the network with varying number of layers. The results show that applying dropout improves the accuracy, achieving a DER of 3.19% for ATB3 and 2.03% for Tashkeela with a network of four layers. We experimented going for a deeper network of five layers when dropout is used. Results did not show improvement in the obtained DER for both ATB3 and Tashkeela, as shown in Figure 9.

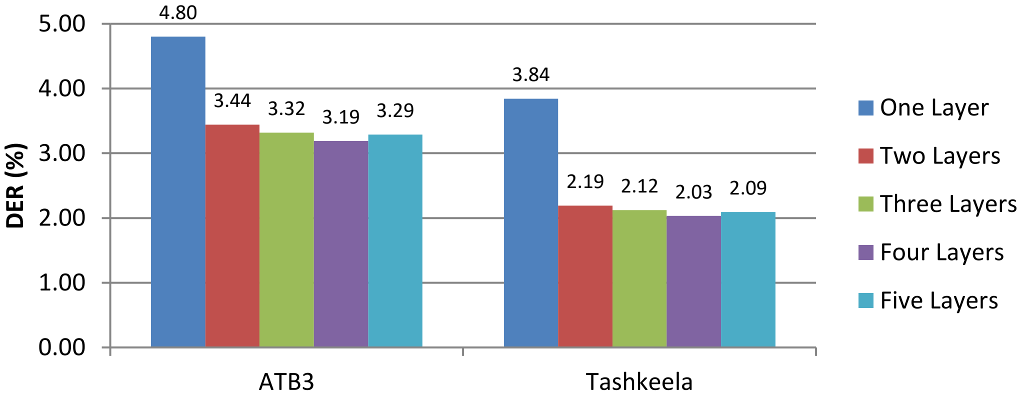


Figure 9. DER of five network sizes with dropout.

6. Discussion

Our best results are reported here using four bidirectional LSTM layers with dropout, diacritics only encoding of the target sequences and wrapping long input sequences to 400 characters. The following three subsections compare the results of this work with previous work, analyze the output diacritization errors and summarize other studied model hyper-parameters in this work.

6.1 Comparison with Existing Systems

Table 9 summarizes the comparison of this work and previous work. With the post processing techniques proposed in our previous work [4], the best DER and WER are 2.46% and 8.12%, respectively, for ATB3. This improves over our previous work that previously reported a DER of 2.7% for ATB3. For Tashkeela, the best DER and WER are 1.97% and 5.13%, respectively. This provides 47% DER and 54% WER improvement over the best-reported DER of the Shakkala framework tested by Fadel et al. [7]. In addition, Table 9 shows DER and WER when errors in diacritizing the last letter of each word are ignored. Especially for ATB3, error rates significantly improve when these errors are ignored. Last letter diacritization depends on the context and hence is considered more difficult than diacritizing other letters. The last column in Table 9 shows DER resulting from last-letter diacritization errors only. For both ATB3 and Tashkeela, our model provides better last-letter diacritization error rates compared to other systems.

Table 9. Comparison with previous work on LDC ATB3 and Tashkeela (TKL) datasets

System	***All Diacritics***				***Ignore Last***				***DER Last***	
	ATB3		***TKL***		***ATB3***		***TKL***		***ATB3***	***TKL***
	DER	***WER***	***DER***	***WER***	***DER***	***WER***	***DER***	***WER***	***DER***	***DER***
Zitouni et al. (2006) [17]	5.5	18	-	-	2.5	7.9	-	-	3.0	-
Habash & Rambow (2007) [5]	4.8	14.9	-	-	2.2	5.5	-	-	2.6	-
Rashwan et al. (2011) [18]	3.8	12.5	-	-	**1.2**	**3.1**	-	-	2.6	-
Said et al. (2013) [19]	3.6	11.4	-	-	1.6	4.4	-	-	2.0	-
Abandah et al. (2015) [4]	2.7	9.1	-	-	1.38	4.34	-	-	1.34	-
Fadel et al. (2019) [7]	-	-	3.73	11.19	-	-	2.88	6.53	-	0.85
This work	**2.46**	**8.12**	**1.97**	**5.13**	1.24	3.81	**1.22**	**3.13**	**1.22**	**0.75**

6.2 Diacritization Error Analysis

We analyze errors of our system by tallying the errors according to the number of errors per word and the presence of last-letter diacritization error. The results of this analysis for ATB3 and Tashkeela are shown in Table 10. The table shows that most of the miss-diacritized words have one diacritic error at 79.4% and 74.3% for ATB3 and Tashkeela, respectively. Words with three or more diacritic errors are not frequent at 4.8% and 4.4% for ATB3 and Tashkeela, respectively. The table also shows that in ATB3, 62.8% of word errors include an error in last letter diacritic. Tashkeela has a smaller ratio of these errors (49.6%), because Tashkeela is a larger dataset and has a lower ratio of missing diacritics (see Table 5). Also, notice that when there is an error in the last letter diacritics, the error distribution tail is longer reaching, e.g., 0.5% for four errors or more *versus* 0.2% or 0.1% when the last letter is OK.

Table 10. Distribution of word errors in percent (%).

Dataset	***Errors per word***	***One***	***Two***	***Three***	***Four +***	***Total***
ATB3	**Last letter OK**	26.3	9.1	1.6	0.2	37.2
	Error in last letter	53.1	6.6	2.5	0.5	62.8
	Total	79.4	15.7	4.1	0.7	100.0
Tashkeela	**Last letter OK**	35.2	13.5	1.6	0.1	50.4
	Error in last letter	39.1	7.8	2.2	0.5	49.6
	Total	74.3	21.3	3.8	0.6	100.0

We have inspected 200 diacritization error samples of the ATB3 test set. For these samples, we analyzed the sources of the errors in the last-letter and other letters (internal) diacritics. Additionally, we report ratios of some specific error types, such as errors in words that have *Shadda*, errors that are not harmful and errors in composite words. Table 11 shows the results of this analysis.

Almost three quarters of the errors in end word diacritics are due to incorrect prediction by the proposed model. For the example output sentence shown in the table, there is no reason for the underlined miss-diacritized word خُطُورَةٍ to have *Kasratan* ◌ٍ instead of Fathatan ◌ً. Another source of errors is having undiacritized letters in the training and testing sequences. Having undiacritized letters in the training set leads to a model that does not diacritize some letters (13% of the selected samples). Target sequences with undiacritized letters are responsible of 8% of the sample errors. The underlined word المجد of the third example target sentence is not diacritized, whereas the output word المجْدِ is correctly diacritized. The rest end-word diacritic errors (6%) are due to not having enough context for the model to predict the last-letter diacritics. For example, our model fails to diacritize the last letter of the word السبت, since it comes alone, not included in a proper sentence.

Table 11. Analysis of sample errors.

Criteria	***Ratio***	***Examples***	
		Target	***Output***
End word diacritics			
Incorrect prediction	73%	إِنَّنا نُواجِهُ هَجْمَةً أَكْثَرَ خُطُورَةً	إِنَّنا نُواجِهُ هَجْمَةً أَكْثَرَ خُطُورَةٍ
Not diacritized	13%	الَّذِي رَأَى أَوَّلَ مِن أَمْسِ	الَّذِي رَأَى أَوَّلَ مِن أَمْس
Target error	8%	بَيْنَما سَجَّلَ إِصابَةَ المجد	بَيْنَما سَجَّلَ إِصابَةَ المَجْدِ
Not enough context	6%	السَّبْتُ	السَّبْت
Internal diacritics			
Incorrect prediction	35%	إِلَى عَشْرِ كُراتٍ مُرْتَدَّةٍ	إِلَى عَشْرِ كُرّاتٍ مُرْتَدَّةٍ
Valid word	31%	وَلَوْ كُنّا نُسَلِّمُ بِهذا الأَمْرِ	وَلَوْ كُنّا نُسَلَّمَ بِهذا الأَمْرِ
Name word	18%	وَانْتَقَلَ بَعْدَ ذلِكَ مِن بينارول إِلَى ميلان	وَانْتَقَلَ بَعْدَ ذلِكَ مِن بينارُول إِلَى ميلانٍ
Possible alternative	7%	مَعْلُوماتٍ تُدَعِّمُ حُجَجَ بُوش	مَعْلُوماتٍ تَدْعَمُ حُجَجَ بُوش
Target error	6%	عادَ مُجَدِّداً إِلَى تكتل لِيكُود	عادَ مُجَدِّداً إِلَى تَكَتُّلِ لِيكُود
Not diacritized	3%	وَمِن نَفْسِ مادَّةِ الدِّينامِيت	وَمِن نَفْسِ مادَّةِ الدِّيناميت
Error types			
Has *Shadda*	23%	يُجَنِّبُهُ ضَرْبَةً عَسْكَرِيَّةً	يَجْنّبُهُ ضَرْبَةً عَسْكَرِيَّةً
Not harmful	21%	أَوْضَحَ بورعد	أَوْضَحَ بُورعد
In composite word	12%	حَصَرَ مَجْلِسُ الوُزَراءِ مُناقَشاتِهِ	حَصَرَ مَجْلِسُ الوُزَراءِ مُناقَشاتُهُ

For the diacritization errors in the internal letters, incorrect prediction that gives invalid words is at 35% of the samples. In 31% of the cases, the proposed model produces a diacritization output that gives a valid Arabic word, but the output word is not suitable for the context. For example, the word نسلم was diacritized as نُسَلَّم (passive voice meaning "we will be given") instead of نُسَلِّمُ (active voice meaning "we accept"). In 7% of the cases, the model produces a diacritization output that is a correct diacritization alternative, but is different from the target diacritics. For example, the model diacritizes the word تدعم as تَدْعَمُ instead of تُدَعِّمُ and both words provide the same meaning of "it supports". Other internal letter errors are in words that represent names (18%). Diacritizing foreign names such as بينارول (Club Atlético Peñarol) is hard, because they are often out-of-the-vocabulary and not diacritized in the dataset. The rest internal diacritics errors are due to lack of diacritization in the target or output sequences at 6% and 3%, respectively.

Of the selected samples, 23% of the errors are in words that have *Shadda*. Restoring diacritics of these words is more difficult, as the word diacritics tend to be more complex when *Shadda* is present. We observed also that 21% of the sample errors are not harmful, such that the miss-diacritized words can still be correctly read and understood. For example, the model diacritizes the name word بورعد as بُورعد. Adding the *Damma* in this case does not affect the word pronunciation. Finally, errors in composite words contribute 12% of the error samples. Predicting the diacritics of composite words that have prefixes and/or suffixes is harder than that of simple words. For example, in a composite word with a suffix, the inflection diacritic of the stem word is on the letter before the suffix, not on the last letter. In the last example of the table above, the model fails to retrieve the correct diacritic for the pronoun suffix ـه in the word مناقشاته by diacritizing it as مُناقَشاتُهُ instead of مُناقَشاتِهِ.

6.3 Other Experiments

In addition to the experiments reported in the previous sections, we performed other experiments for which results are not reported here, because they do not improve the accuracy of our model. These experiments included testing *Adam* optimizer instead of the RMSprop optimizer [28]. For all experiments, RMSprop performed better than Adam optimizer did. Moreover, we experimented with adding L1 and L2 regularization in lieu of dropout to overcome overfitting instead of dropout [28]. Results show that regularization does not improve training and even produces worse accuracy in some cases.

7. Conclusions

We performed intensive experiments to find a fast and accurate solution. Our experiments used the LDC ATB3 dataset as an example of MSA and a clean subset of Tashkeela dataset as an example of CA. Our experiments included studying the variation in sequence lengths of the used datasets. We experimented with handling this variation using different approaches and tested both the accuracy and training time. We recommend wrapping very long sequences to segments not longer than 400 letters. We also proposed two new encoding methods for target diacritized sequences. Our experiments show that the proposed encoding using diacritics only improves the accuracy, since it simplified the network output layer.

We tested different network architectures and the results show the superiority of the bidirectional LSTM network over the encoder/decoder network and the unidirectional LSTM. We also tuned our model by going for a deeper network and applying dropout. Our best results are reported for a bidirectional RNN LSTM with four layers that uses dropout. Best achieved DER is 2.46% and 1.97% for ATB3 and Tashkeela, respectively. Our best DER for Tashkeela provides an improvement of 47% over the best-published result.

The results of this work open doors for future work. The proposed dataset file sparse encoding, wrapping long sequences, efficient bidirectional deep LSTM and tuned hyper-parameters allow efficient training using large datasets. We intend to improve the accuracy of the proposed model based on the insights gained from the diacritization error analysis above. The diacritization accuracy should improve when we use larger MSA dataset. Note that Tashkeela has better accuracy and is larger than ATB3. Additionally, we need to solve the problem of having missing diacritics in some of training sequences. Such sequences confuse the network and result in some undiacritized output. Finally, as some diacritization differences between the output and the target sequences are not more harmful than other differences, we need to develop a better loss function that considers this issue when training the network.

References

[1] N. Y. Habash, Introduction to Arabic Natural Language Processing, Synthesis Lectures on Human Language Technologies, Morgan and Claypool Publishers, 2010.

[2] G. Abandah, M. Khedher, W. Anati, A. Zghoul, S. Ababneh and M. Hattab, "The Arabic Language Status in the Jordanian Social Networking and Mobile Phone Communications," Proc. of the 7th Int'l Conference on Information Technology (ICIT 2015), pp. 449-456, 2015.

[3] G. A. Abandah and F. Khundakjie, "Issues Concerning Code System for Arabic Letters," Dirasat-Eng. Sci. J., vol. 31, no. 1, pp. 165-177, 2004.

[4] G. A. Abandah, A. Graves, B. Al-Shagoor, A. Arabiyat, F. Jamour and M. Al-Taee, "Automatic Diacritization of Arabic Text Using Recurrent Neural Networks," International Journal on Document Analysis and Recognition (IJDAR), vol. 18, no. 2, pp. 183-197, 2015.

[5] N. Habash and O. Rambow, "Arabic Diacritization through Full Morphological Tagging," Proc. of Conference on North American Chapter of the Association for Computational Linguistics, pp. 53-56, 2007.

[6] M. Maamouri, A. Bies, T. Buckwalter and W. Mekki, "The Penn Arabic Treebank: Building a Large-scale Annotated Arabic Corpus," Proc. of Conference on Arabic Language Resources and Tools (NEMLAR), pp. 102-109, 2004.

[7] A. Fadel, I. Tuffaha, B. Al-Jawarneh and M. Al-Ayyoub, "Arabic Text Diacritization Using Deep Neural Networks," arXiv: 1905.01965v1, 2019.

[8] A. M. Azmi and R. S. Almajed, "A Survey of Automatic Arabic Diacritization Techniques," Natural Language Engineering, vol. 21, pp. 477-495, 2013.

[9] O. Hamed and T. Zesch, "A Survey and Comparative Study of Arabic Diacritization Tools," JLCL: Special Issue-NLP for Perso-Arabic Alphabets, vol. 32, no. 1, pp. 27-47, 2017.

[10] Y. Gal, "An HMM Approach to Vowel Restoration in Arabic and Hebrew," Proceedings of the ACL-02 Workshop on Computational Approach to Semitic Languages (SEMITIC ‘02), pp. 27-33, 2002.

[11] E. Elshafei, H. Al-Muhtaseb and M. Alghamdi, "Statistical Methods for Automatic Diacritization of Arabic Text," Proceedings of Saudi 18th National Computer Conference (NCC18), pp. 301-306, 2006.

[12] Y. Hifny, "Smoothing Techniques for Arabic Diacritics Restoration," Proceedings of the 12th Conference on Language Engineering (ESOLEC ‘012), pp. 6-12, 2012.

[13] T. Zerrouki, "Arabic Corpora Resources, Tashkila Collection from the Arabic Al-Shamela Library, [Online], Available: "http://aracorpus.e3rab.com/, [Accessed Aug. 27, 2019].

[14] A. S. Azim, X. Wang and K. C. Sim, "A Weighted Combination of Speech with Text-based Models for Arabic Diacritization," Proceedings of the 13th Annual Conference of International Speech Communication Association, pp. 2334-2337, 2012.

[15] D. Vergyri and K. Kirchhoff, "Automatic Diacritization of Arabic for Acoustic Modelling in Speech Recognition," Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages, pp. 66-73, 2004.

[16] R. Nelken and S. M. Shieber, "Arabic Diacritization Using Weighted Finite-state Transducers," Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages, pp. 79-85, 2005.

[17] I. Zitouni, J. S. Sorensen and R. Sarikaya, "Maximum Entropy-based Restoration of Arabic Diacritics," Proceedings of the 21st International Conference on Computational Linguistics, pp. 577-584, 2006.

[18] M. Rashwan, M. Al-Badrashiny, M. Attia, S. Abdou and A. Rafea, "A Stochastic Arabic Diacritizer Based on a Hybrid of Factorized and Unfactorized Textual Features IEEE/ACM Transactions on Audio, Speech and Language Processing, vol. 19, no. 1, pp. 166-175, 2011.

[19] A. Said, M. El-Sharqwi, A. Chalabi and E. Kamal, "A Hybrid Approach for Arabic Diacritization," In: E. Mtai, F. Mezaine, M. Saraee, V. Sugumaran and S. Vadera (Eds.), "Natural Language Processing and Information Systems," Lecture Notes in Computer Science, vol. 7934, pp. 53-64, Springer, 2013.

[20] S. Alquda, G. Abandah and A. Arabiyat, "Investigating Hybrid Approaches for Arabic Text Diacritization with Recurrent Neural Networks," Proceedings of the 2017 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), pp. 1-6, 2017.

[21] M. Rashwan, A. Sallab, H. Raafat and A. Rafea, "Deep Learning Framework with Confused Sub-set Resolution Architecture for Automatic Arabic Diacritization," Proceedings of the IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP), vol. 23, no. 3, pp. 505-516, 2015.

[22] A. Barqawi and T. Zerrouki, "Shakkala, Arabic Text Vocalization," [Online], Available: https://github.com/Barqawiz/Shakkala, 2017.

[23] Tahadz, "Mishkal," [Online], Available: https://tahadz.com/mishkal, [Accessed on October 16, 2019].

[24] H. Mubarak, A. Abdelali, H. Sajjad, Y. Samih and K. Darwish, "Highly Effective Arabic Diacritization Using Sequence-to-Sequence Modeling," Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 2390-2395, 2019.

[25] K. Darwish, H. Mubarak and A. Abdelali, "Arabic Diacritization: Stats, Rules and Hacks," Proceedings of the 3rd Arabic Natural Language Processing Workshop, pp. 9-17, 2017.

[26] I. Sutskever, O. Vinyals and Q. V. Le, "Sequence-to-Sequence Learning with Neural Networks," Advances in Neural Information Processing Systems (NIPS), arXiv: 1409.3215v3, 2014.

[27] A. Graves, A. R. Mohamed and G. Hinton, "Speech Recognition with Deep Recurrent Neural Networks," Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 6645-6649, 2013.

[28] A. Geron, Hands-on Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools and Techniques to Build Intelligent Systems, USA: O’Reilly, 2017.

[29] S. Hochreiter and J. Schmidhuber, "Long Short-term Memory," Neural Computation, vol. 9, no. 8, pp. 1735-1780, 1997.

[30] K. Cho, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio, "Learning Phase Representations Using RNN Encoder-Decoder for Statistical Machine Translation," arXiv: 1406.1078v3, 2014.

[31] M. Schuster and K. K. Paliwal, "Bidirectional Recurrent Neural Networks," IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673-2681, 1997.

[32] Google, "TensorFlow," [Online], Available: https://www.tensorflow.org/, [Accessed on Aug. 27, 2019].

ملخص البحث:

تُكتب النصوص بالعربية اليوم دون وضع الحركات على الأحرف، علماً بأنّ وضع هذه الحركات من شأنه أن يقلل من الغموض في قراءة النص، الى جانب فوائد أخرى. يهدف هذا العمل البحثي الى تطوير حلّ دقيق وسريع باستخدام تعلُّم الآلة من أجل وضع الحركات على أحرف النّصّ أوتوماتيكياً. يستخدم هذا البحث الشّبكات العصبيّة المتكرّرة اعتماداً على الذاكرة الطويلة قصيرة الأمد لوضع الحركات على أحرف النّصوص المكتوبة بالعربية.

تم إجراء تجارب مكثّفة لتقييم التّصميم المقترح وخيارات ترميز البيانات للحصول على حلّ دقيق وسريع. وتتضمن تلك التجارب البحث في المشكلات المتعلقة بطول التّتابُع، واقتراح وتقييم خيارات التّرميز للمُخرجات التي تم وضع الحركات على أحرف النّصّ فيها، وضبط الخيارات المتعلقة بالشبكات العصبية وتقييمها من حيث البِنية وحجم الشبكة والمتغيرات العُليا.

وتوصي هذه الدراسة بحل يمكن تدريبُهُ بسرعةٍ على مجموعة بياناتٍ ضخمة باستخدام طبقات ذاكرة طويلة قصيرة الأمد ثنائية الاتّجاه عددها (4) طبقات؛ من أجل توقُّع الحركات على أحرف تتابع المدخل لنصوص اللُّغة العربية. وتميَّز الحلّ المقترح بنسبة خطأ في وضع الحركات على الأحرف مقدارها 2.46% و 1.97% عند تطبيقه على مجمةعة البيانات (LDC ATB3) ومجموعة البيانات الجديدة الأضخم المعروفة باسم (تشكيلة)، على الترتيب. وهذه النسبة الأخيرة تمثّل تحسيناً بنسبة 47% مقارنة بأفضل النتائج المنشورة سابقاً.

AGE ESTIMATION USING SPECIFIC DOMAIN TRANSFER LEARNING

Arwa Al-Shannaq[1] and Lamiaa Elrefaei [2]

(Received: 19-Oct.-2019, Revised: 11-Dec.-2019, Accepted: 29-Dec.-2019)

ABSTRACT

Nowadays, the engagement of deep neural networks in computer vision increases the ability to achieve higher accuracy in many learning tasks, such as face recognition and detection. However, the automatic estimation of human age is still considered as the most challenging facial task that demands extra efforts to obtain an accepted accuracy for real application. In this paper, we attempt to obtain a satisfied model that overcomes the overfitting problem, by fine-tuning CNN model which was pre-trained on face recognition task to estimate the real age. To make the model more robust, we evaluated the model for real age estimation on two types of datasets: on the constrained FG_NET dataset, we achieved 3.446 of MAE, while on the unconstrained UTKFace dataset, we achieved 4.867 of MAE. The experimental results of our approach outperform other state-of-the-art age estimation models on the benchmark datasets. We also fine-tuned the model for age group classification task on Adience dataset and our model achieved an accuracy of 61.4%.

KEYWORDS

Age estimation, Transfer learning, Classification, Regression, VGGFace, Convolutional neural network.

1. INTRODUCTION

To characterize the human identity, different attributes can be derived from the facial image. Age is a crucial trait that can support other significant properties, such as fingerprint and iris, to get a more realistic system for the task of identification and verification of human identity [1]. Age estimation is related to the automatic process of predicting the real age as an exact age of a person or classifying the image into age groups represented by age range [2]. Therefore, estimating the real age of a person is much harder than just classifying a person to which age's category he belongs. Age estimation problem is considered as the most challenging facial task that is affected by various internal factors, such as gender, race and external factors, such as environment conditions and facial expressions. Recently, a growing interest is witnessed to automate age estimation systems and great efforts are made to enhance this challenging task.

Deep learning is one of the new technologies having been increasingly used in the field of computer vision. There is no doubt how deep learning technology outperforms the traditional algorithms of machine learning. Nonlinear features can be automatically extracted using Convolutional Neural Networks (CNNs) [3]. The power of CNNs is related to their capability of hierarchical learning for concepts across several layers. Despite the success of deep learning in many tasks, the accuracy of age estimation systems is still practically unacceptable.

Transfer learning [4] offers many benefits for deep learning-based models. It increases the training speed on new data. Furthermore, it requires less amount of data to train the model compared with training from scratch. Also, it improves the performance of the network. Generally, there are two types of transfer learning: general-domain pre-trained models where knowledge is transferred from a general task to a target unrelated task, whereas models learned on a specific domain are pre-trained on a related or similar task to the target task.

Face recognition and age estimation are different tasks. However, we can argue that they are correlated. While the recognition learning process was made for the VGGFace of the facial features and landmarks across a large number of images related to the same person in different conditions, this process can be

1. A. Al-Shannaq is with the Computer Science Department, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia. E-mail: aalshannaq@stu.kau.edu.sa

2. L. Elrefaei is with the Computer Science Department, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia and with The Electrical Engineering Department, Faculty of Engineering at Shoubra, Benha University, Cairo, Egypt. E-mail: laelrefaei@kau.edu.sa, lamia.alrefaai@feng.bu.edu.eg

considered as an initial step for an age estimation system rather than using a random initialization of weights for the network. To gain the advantage of reusing specific domain pre-trained models, our model is considering the VGGFace [5] as the CNNs architecture. The aim of choosing this network is related to many reasons: Firstly, the state-of-art results that have been achieved by this model in face recognition task. Secondly, VGGFace has been pre-trained on a large database that contains 2.6 M images of 2.6 K people and this can overcome the overfitting problem when training the same model on small datasets on related task of face recognition. Thirdly, using a pre-trained model makes the training process faster while increasing the overall performance. Finally, age can be considered as a facial trait, thus the learning process will be easier than using general domain transfer learning.

In this paper, we prove that selecting a good online data augmentation helps improve the performance. Using a pre-trained model on a large dataset for face recognition task has a good impact on extracting related features to age and can prevent the overfitting problem. Moreover, combining the Global Average Pooling (GAP) with Fully Connected (FC) layers to generate the classifier increases its ability to perform the age estimation task. Finally, treating the age as a multi-classification problem is good when using a balanced distribution over the dataset's classes.

The remainder of this paper is divided into five sections as follows: Section 2 gives a brief overview of the related work on age estimation. The methodology with the baseline network is outlined in the third section. Section 4 displays the different experiments that have been conducted to achieve the best performance with related results. A discussion of the results is addressed in the fifth section. Our conclusions are drawn in the final section.

2. Related Work

To estimate real age, two main steps should be followed. Firstly, the image representation should be extracted, then the feature vector is formalized. Secondly, an encoding algorithm is used to estimate the age [6]. Traditional methods carry out the process of age estimation by extracting the related features to age then formalizing the feature vector. Before representing the different facial details, the faces should be detected and aligned. Then, for the stage of extraction of the local and global features, a descriptor such as Active Appearance Model (AAM) [7], Biologically Inspired Feature (BIF) [8] or Local Binary Patterns (LBPs) [9] should be carefully chosen. The dimension of the resulted feature vector could be very high. Therefore, a reduction algorithm such as Principle Components Analysis (PCA) is used to reduce the dimension. The traditional techniques are classified into six groups according to [10]: anthropometric-based [11][12][13] that are based on the face geometry. To measure the geometric ratios from images correctly, faces should be in frontal view, since the computations of ratios from 2D images are sensitive. The second method is texture-based model [8]-[9]. Texture information can be directly calculated from images using pixel intensities. To extract these features, many effective descriptors such as BIF have been utilized in several works [14][15][16]. AAM [7] is another effective algorithm that combines shape and texture. The models are learned through the training process using a number of images. After that a parametric model for faces is generated by PCA. Another model is aging pattern subspace [17]-[18] which defines the aging pattern as a sequence of personal facial images belonging to the same person and finally the aging manifold [19]-[20], where the aging pattern can be commonly learned as a trend for several subjects at different ages. The features from different descriptors can be also fused to obtain a more robust system [21]-[22].

Since the development of deep learning technology, many researchers turned into replacing the traditional techniques with this technology. CNNs [3] have been widely used in computer vision field due to their effective learning ability for nonlinear features. Recent works on age estimation using deep learning [23][24][25] have only focused on developing their approaches using models that were pre-trained on ImageNet [26] dataset for object classification task. Other approaches [27][28][29] found that using a model pre-trained for a specific-domain related to age prediction task such as face recognition can achieve a better performance on age estimation. Yang et al. [30] trained their CNN model from scratch to derive different information from human faces, such as age, gender and race. Their results were lowest compared to previous works on age estimation which were based on traditional techniques for feature extraction. Instead of extracting the features from the top layer, Wang et al. [18] obtained the features from different layers. They enhanced the system by adopting the manifold algorithm with the basic model. Liu et al. [28] built a multi-path model and combined a VGGFace network with two shallow VGG16 [31] networks. The feature vector of the three networks was normalized and fed to the age estimator. Hu et al. [32] learned the age from two pairs of images which belong to the same person. They found the age difference by using the Kullback-Leibler divergence as loss function. Rodríguez et al. [33] tried to place more attention on VGGFace [5] that is pre-trained on face recognition task. The model consists of two CNNs; the patch network which receives the low-resolution image and the attention CNN which is fed with the high-resolution image to place more attention on the important regions on the face.

Estimating real age can be implemented as a multi-classification task [34][35]. In this case, ages are represented as separated labels. Therefore, the classifier will predict the class of the age after the learning process using one of the available benchmarks annotated with age labels. On the other hand, we can use the regression algorithm [36]-[37] because of the nature of the human ages, as they are continuous values. A new algorithm which shows good results in age estimation is called ranking algorithm [38] [39] [40]. Instead of considering the age as a multi-class task, ranking algorithm convers the problem into different binary classification tasks and the resulted age ranks are the aggregation of the outputs from different classifiers. Rothe et al. [27] pre-trained the VGG16 on a large unconstrained dataset: IMDB_WIKI [27] for real age estimation task. They considered the age as a multi-class problem with linear regression activation.

Antipov et al. [29] used VGG16 and pre-trained it on face recognition task. They used a soft classification to encode age. They found that pretraining on face recognition task is more suitable for age and gender classification than general task pre-training, while the strategy of multi-task pretraining is useful in case of training the model from scratch. To overcome the problem of sample imbalance, Li et al. [25] used AlexNet for feature extraction with a cumulative hidden layer. The main advantage of using a cumulative hidden layer is to learn the ages from faces with neighboring ages. Shang and Ai [41] separated the related features of aging into different groups by using clustering algorithm k-means++. They trained the network again for each group to estimate the final age for each subject. Later, Zhang et al. [42] used the DEX method [27] to estimate the real age. They improved the age estimation system by extracting the fine-grained features using the attention mechanism. A new loss function was proposed by [43] based on finding the age distribution based on the mean and variance of the ground-truth age.

Deep network can be utilized as a feature extractor. Duan et al. [15] extracted the features using CNNs. They combined classification and regression by firstly classifying the images according to age groups using Extreme Machine Learning (ELM) and then regressing the final value of the age using ELM regressor. Their model was a combination between classification and regression. Chang and Chen [38] used the scattering transform to extract the Gabor coefficients. They treated the age labels as a ranking algorithm, where the aggregation of the results from a series of binary classifiers performs the age ranks. Chen et al. [39] trained a set of CNNs on ordinal age labels. Different outputs were obtained and aggregated to predict the final age. Recently, Li et al. [44] extracted the features using CNNs and fed the BridgeNet which consists of local regressors and gated networks. The aim of using the gated networks was to weigh the regression results. Thus, the final age was calculated by taking the summation of all weights resulted from the local regressors.

The literature shows that using a pre-trained model on general task, such as on ImageNet, needs more efforts and deep networks to achieve reasonable results for age estimation task. A more powerful model is to pre-train the network on a specific task, such as face recognition or gender classification. Inspired from this consequence, we use a pre-trained model on face recognition that can increase the system ability to extract the related features of age.

3. METHODOLOGY

In deep learning, transfer learning technique can be defined as the process of reusing a model that has already been trained for a specific task to perform a similar or related task [4]. The aim of using a pre-trained model is to take the advantage from the features that have been extracted in the front layers instead of developing the model from scratch. Moreover, the computation time for training can be reduced while using a pre-trained model. Different policies can be followed while reusing pre-trained models:

1. Some late layers can be set as a trainable layer, so that their weights will be fine-tuned for the new task. This policy can be used when a dataset is available with plenty labels.
2. Freezing the convolutional base and adding classification layers in case that a small dataset is available which is similar to the source dataset that has been used in the pre-training stage.
3. Training the entire model or training from scratch which needs extra computational time and power with a very large dataset.

To build the classification layers, we can use one of the following approaches:

1. Adding an FC layer or a set of stacked FC layers followed by a Softmax activated layer for classification task or a linear activated layer for a regression task.
2. Adding GAP as proposed by Lin et al. [45] and connecting this layer directly to the output layer. The main concept of GAP is that it reduces the dimension of each tensor by taking the average of each feature map. For example, if we have a tensor with dimension (*h*x*w*x*d)*, GAP reduces the dimension from (*h*x*w*x*d)* to (*1*x*1*x*d*) by taking the average of each feature map (*h*x*w)*. Moreover, this layer has a similar affect as the FC layer except that it can avoid overfitting, since there are no parameters to optimize.

In this study, different approaches were investigated to reuse the base model to build the classifier by adapting deeper and wider schemes. The model was also examined in terms of the power of concatenating the GAP with FC layers to leverage performance. Classification and regression were both implemented to estimate real age, while for the age group task, classification was implemented.

3.1 Network Architecture

VGGFace-based VGG16 consists of eight convolutional layers and the classifier layers. On the classification block, there are two FC layers, where each layer has a 4096-dimensional output. An activation layer that has the rectification operator, such as Rectified Linear Activation Unit (ReLU), is added between these FC layers. After each block, a max pooling layer is added to down sample the feature map. The last layer represents the output layer with 2622 classes reflecting the number of subjects in the database. The activation in the output layer is chosen to be Softmax function for multi-class classification problem. The base model with the top connected layers pre-trained on face recognition task is shown in Table *1*.

3.2 Adding Batch Normalization between FC Layers

A good technique that approved its efficiency in avoiding overfitting problem is using regularization. Inserting Batch Normalization (BN) [46] between the convolution layers will regularize and make the model more stable. BN layer takes the output of the preceding activation layer and normalizes it by subtracting the mini-batch mean and dividing by the mini-batch standard deviation. In other words, the normalizing transform aims to repair the means and the variances of layer inputs by adding two trainable parameters at each layer. Moreover, it reduces the network dependency on the initialization of each layer, which allows to use a higher learning rate.

3.3 Improving the Model with Online Data Augmentation

One of the most important techniques to enhance the performance and robustness of deep learning models is to train the neural network with a large amount of data. Unfortunately, most of the image-based applications have limited datasets or the conditions do not reflect the real-world scenarios under which images have been taken. Age estimation task is considered as a challenging computer vision task that is affected by many internal and external factors [28], [47]. There are no general patterns of aging for all humans. A more realistic age estimation system should be able to learn more irrelevant patterns of ages in different conditions.

Data augmentation [48] is a regularization technique used to feed the neural network with more synthetic images to reflect more realistic conditions and perspectives. By applying data augmentation, the network can avoid the overfitting problem that is caused by using small datasets. Different conditions can be applied to make additional modified images, such as translation, rotation, scaling, brightness, …etc. In this work, we select online augmentation to be applied to the images during the training process as the batches are passed. Thus, the training process will be quicker and there is no need to load the original data with the augmented data to the memory as in offline augmentation [48]. We diagnosed different transformation functions of online data augmentation to select the most appropriate one that enhances the performance of the model.

Table 1. Main architecture of VGGFace network.

	Layer Name	# of Filters	Feature Map	Stride
	input_1 (InputLayer)		(224, 224, 3)	
Block 1	conv1_1 (Conv2D)	64	(224, 224, 64)	3x3
	conv1_2 (Conv2D)	64	(224, 224, 64)	3x3
	pool1 (MaxPooling2D)		(112, 112, 64)	2x2
Block 2	conv2_1 (Conv2D)	128	(112, 112, 128)	3x3
	conv2_2 (Conv2D)	128	(112, 112, 128)	3x3
	pool2 (MaxPooling2D)		(56, 56, 128)	2x2
Block 3	conv3_1 (Conv2D)	256	(56, 56, 256)	3x3
	conv3_2 (Conv2D)	256	(56, 56, 256)	3x3
	conv3_3 (Conv2D)	256	(56, 56, 256)	3x3
	pool3 (MaxPooling2D)		(28, 28, 256)	2x2
Block 4	conv4_1 (Conv2D)	512	(28, 28, 512)	3x3
	conv4_2 (Conv2D)	512	(28, 28, 512)	3x3
	conv4_3 (Conv2D)	512	(28, 28, 512)	3x3
	pool4 (MaxPooling2D)		(14, 14, 512)	2x2
Block 5	conv5_1 (Conv2D)	512	(14, 14, 512)	3x3
	conv5_2 (Conv2D)	512	(14, 14, 512)	3x3
	conv5_3 (Conv2D)	512	(14, 14, 512)	3x3
	pool5 (MaxPooling2D)		(7, 7, 512)	2x2
Classification block	Flatten layer			
	Fully Connected (FC6)	Input size= 4096		
	RELU Activation			
	Fully Connected (FC7)	Input size= 4096		
	RELU Activation			
	Fully Connected (FC8)	Input size= 2622		
	Softmax Activation layer			

3.4 Fine-tuning VGGFace Model for Age Estimation

A single network of VGGFace is used and fine-tuned for real-age estimation. We propose two approaches to reuse the basic model:

Approach-1: In this approach, we keep the base convolutional layers with the top classification layers and remove the last Softmax activation layer. Then, the resulted feature map from FC8 layer is connected to extra FC layers. We freeze all the layers except the new additional one. The model is examined when adding different numbers of FC layers that have different numbers of neurons. The last layer is the output layer which is connected to the model as a dense layer. This approach is shown in Figure 1.

Approach-2: In this approach, we want to combine the GAP layer and the FC layers. We keep the base convolution layers and remove the top classification layers, then connect the last max pooling layer with GAP layer. The output of GAP is then fed to the FC layers. The model is examined when adding different numbers of FC layers that have different numbers of neurons. The last layer is the output layer which is connected to the model as a dense layer. Figure 2 shows the proposed structure of approach-2 to fine-tune VGGFace network on age estimation.

3.5 Classification *vs.* Regression

Age can be estimated using one of the age encoding algorithms. If the age is considered as a multi-classification problem, then each age will represent a single class. However, we can find a correlation

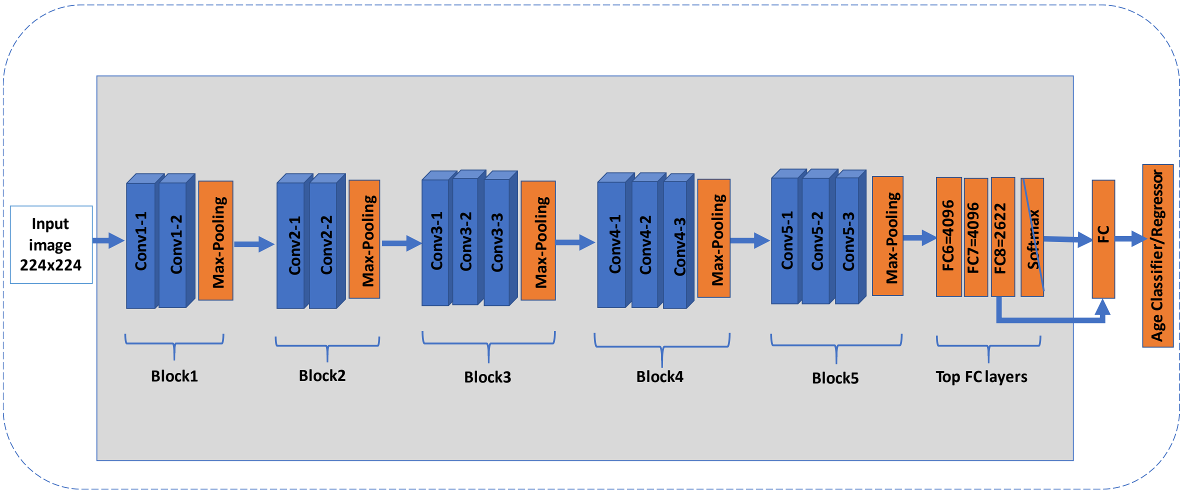


Figure 1. Approach-1 for fine-tuning VGGFace for age estimation.

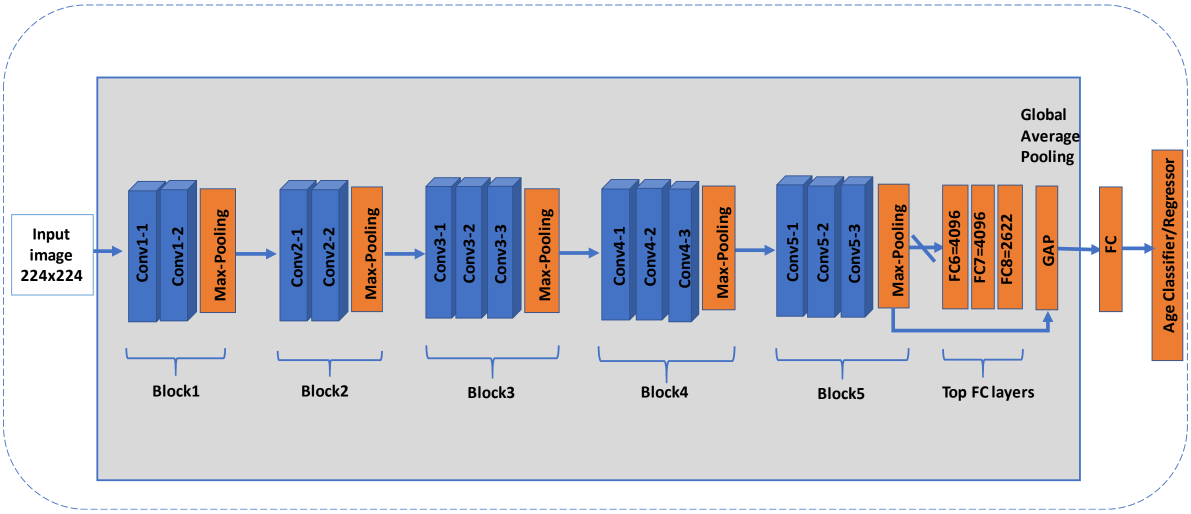


Figure 2. Approach-2 for fine-tuning VGGFace for age estimation.

and continuous behavior across different ages. In this case, ages can be considered as a regression problem. In this work, we investigated the classification and regression algorithms to encode age.By considering age as a multi-class problem, we added an age classifier on the top of the network as an output. We used the one-hot encoding method to represent the age labels. Thus, each sample will have a probability value of 1.0 for the correct class and 0.0 for other class values. To predict this probability, Softmax was applied as the activation function. Softmax function [49] can be defined as in Equation 1:

$$\mathrm{P}(\, y_i \mid x_i, W\,) = \frac{e^{f_{y_i}}}{\sum_j e^{f_j}} \tag{1}$$

where P is the probability value that is assigned to the correct label y_i given the image x_i and parameterized by W. Furthermore, to monitor the training process, categorial-cross entropy was selected as the loss function. The number of outputs in this case is equal to the number of classes on the dataset. Regression method treats the age as continuous values. In our proposed model, we added an age regressor with one output and selected the Mean Absolute Error (MAE) as a loss function to calculate the cost of training process. To estimate ages, we selected the linear activation function. A linear regression line can be represented using Equation 2:

$$Y = \; a + bX \tag{2}$$

where X is the explanatory variable and Y is the dependent variable. The slope of the line is b and a is the intercept (the value of y when x = 0).

4. Experiments and Results

We practically examined the efficiency of our proposed approaches to fine-tune a pre-trained network of VGGFace for real-age estimation. We conducted many experiments to achieve the minimum value of error. The process of learning was further enhanced using online data augmentation, such as rotation, shearing and flipping. After trying many functions for image transformation, we selected the proper augmentation functions and considered them for all experiments. Thus, each image on the training set will be flipped horizontally, sheared randomly to 0.5 and rotated 45 degrees.

4.1 Evaluation Metrics

The most common metric for real-age estimation is the MAE. This metric can be more representative than classification accuracy to evaluate the age estimation system, since it shows the difference between the estimated age and the real age. This metric can be mathematically defined as in Equation 3:

$$MAE = \frac{1}{N}\sum_{i=1}^{n}|\widehat{x_i} - x_i| \qquad (3)$$

The estimated age is represented by $\widehat{x_i}$, where x_i represents the real age and N is the number of testing samples. The evaluation between the different models is based on obtaining a minimum value of *MAE,* which implies a good performance [2].

For the task of age group classification, accuracy is used as the evaluation metric. We calculated accuracy by using Equation 4:

$$CS(x) = \frac{n_x}{N_x} * 100\% \qquad (4)$$

where n_x is the number of correctly classified images to a specific group *x* and N_x represents the number of testing samples of group *x* [17].

4.2 Benchmark Datasets

FG_NET Dataset [50] has 1002 images which are belonging to 82 subjects. There is a grayscale resolution for all images in addition to the colored version. The dataset covers a range of ages from 0 to 69 years, but most of the available images are in the range less than 40 years. On average, there are 12 images per person. The human gender and race annotations are also provided in this dataset. FG_NET is considered as a constrained dataset, where all the images have frontal head poses captured on restricted conditions. Moreover, for the task of face modelling, the database provides 68 landmark points. Because of the problem of highly biased classes in this dataset, we considered a different protocol. Different recent age estimation models used the Leave One Person Out (LOPO), which depends on testing the model using the different images that belong to one person each time and taking the average obtained from all 82 subjects. Thus, to make the testing process more realistic, we considered another splitting protocol. The dataset is divided into 80 % as training set, 10% as validation set and 10% for testing. Figure 3 shows some samples of the dataset.

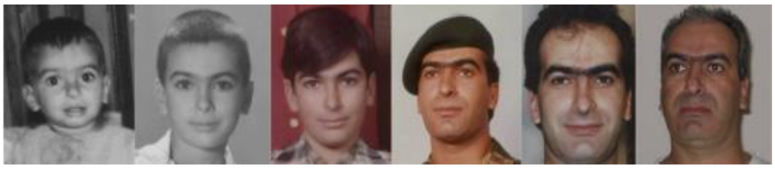


Figure 3. Samples of FG-Net dataset [50].

UTKFace Dataset [51] has over 20,000 face images labelled with age, gender and ethnicity. The images cover a large span of ages from 0 to 116 years. Huge variations in head poses, illumination and occlusion are contained in the images. It has images captured in unconstrained conditions with correct real-age annotations.

Figure 4 shows some images from the dataset. The dataset has been used to evaluate the model on an unconstrained dataset for real-age estimation using validation splitting of 80% for training, 10% for validation and 10% for testing.

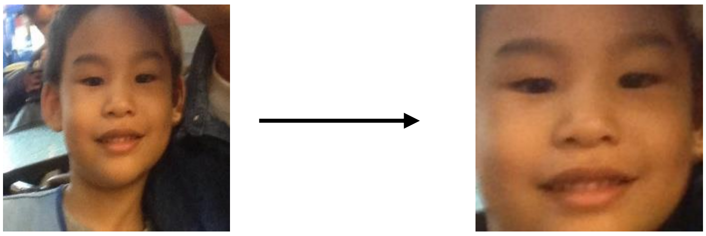
Figure 4. Samples from UTKFace dataset [51].

Adience Dataset [22][52]. It was collected from Fliker.com albums. It is used to make age group and gender classification. Some images from the dataset are shown in Figure 5. Adience dataset consists of 26,580 images that belong to 2,284 subjects. The set contains 8 age groups from 0 to 60 years and older. Big challenges are embodied in this database, while the images have low resolution with extreme blurring, occlusion, different head poses and expressions. Five models are trained for each fold. As the Adience dataset is already divided into five folders, four folders have been used for training and validation while folder0 has been used as a hold-out testing set. Then, accuracy is tested by the average or mean of all accuracies obtained from the five folds. The faces have been detected and the images were cropped to remove the annoying background.

Figure 5. Samples from Adience dataset [22], [54].

4.3 Data Pre-processing

For the Adience dataset, we detected the faces from the images and cropped them to remove the noisy background, as shown in Figure 6.



Original image After face detection and cropping

Figure 6**.** Pre-processing step for images in Adience dataset.

For the UTKFace dataset, we used the aligned and cropped version of the dataset offered from the dataset's website. For the FG_NET dataset, we used the images without making any processing. For all datasets, we rescaled the images to 224x224 resolution to be compatible with VGGFace input size.

4.4 Hardware and Software Tools

The system and its stages were executed using a High-Performance Computer (HPC) with NVidia Tesla GPU. Our model is implemented in Python using Keras with TensorFlow backend.

4.5 Experimental Results

This subsection summarizes the results that were obtained from the experiments for both encoding algorithms: classification and regression using the earlier proposed approaches. **Section 4.5.1** shows the results on the constrained FG_NET dataset. In **section 4.5.2,** we show the results on the unconstrained UTKFace dataset using the proposed approaches. One of the main concerns that has a large effect on performance while reusing the pre-trained model is to optimize the appropriate topology of the classifier with strong generalization of the task. In our experiments, we studied different schemes to connect the pre-trained model with the output layers with changeable number of hidden layers (depth) and number of neurons (width) to construct the most proper structure that can design the age estimation task. We analyzed the proposed approaches using the following different schemes:

- **Scheme-1**: We fine-tuned the base model without adding any FC layers. For Approach-1, we just connected FC8 to the output layer. For Approach-2, we just connected the GAP to the output layer.
- **Scheme-2,3,4**: Varying width, single layer with varying feature size. We started to expand the model by adding one layer with different neuron sizes, such as 1024, 5000 and 6000.
- **Scheme-5,6,7,8**: Varying depth, instead of using one layer, adding a set of two layers. Within this scheme, we examined many combinations. We combined BN with FC layers. After each BN layer, we added ReLU activation layer connected to the FC layer with the following order: FC1+BN+ ReLU +FC2+ BN+ ReLU. The reverse order was also tested: BN+ ReLU +FC1+ BN+ ReLU +FC2 to examine which order is more effective. Another scheme with FC layers and ReLU activation has also be considered, where each FC layer is followed with ReLU layer.

After conducting a number of experiments with different values of the training's parameters, we adjusted the values of these parameters for both datasets as follows: epochs=64, batch size=16, learning rate= 0.0001, split ratio= (80 training,10 validation,10 testing) with Stochastic Gradient Descent (SGD) as an optimizer. To boost performance, we used online data augmentation.

4.5.1 Results on the Constrained FG_NET Dataset

In this section, we show the results of evaluating the proposed approaches on the constrained FG_NET dataset. We used the images from the dataset without making any pre-processing. For the case of classification, the number of nodes for the output layer was set to 70, which is related to the number of classes in the dataset. For regression, the number of nodes was set to 1.

Approach-1: Fine-tuning the base convolutional layers with including top

The model was constructed by freezing the entire model and removing the Softmax activation layer. Then, we connected the feature vector from FC8 with different sizes of extra dense layers. Table 2 shows the results of classification and regression algorithms while connecting different schemes to the base model. It is clear from the results that, using Approach-1 with regression is better than using it with classification and resulted with a more robust model to age estimation. The least MAE was achieved when using the (BN+ ReLU+ Two FC (4096)) with adding BN and activation before each FC layer.

Approach-2: Fine-tuning the base convolutional layers without including top

The fine-tuning process was handled by freezing the base convolutional layers and removing the classification layers. We used GAP to connect the base convolutional layers with the new FC layers.

Table 2. Age estimation results on FG_NET using approach-1.

Scheme Name	Additional FC layer	MAE of Classification	MAE of Regression
No FC	Without FC layer	4.222	5.455
FC1(1024)	FC1(input_size= 1024)	**3.839**	3.855
FC1(5000)	FC1(input_size= 5000)	4.041	3.894
FC1(6000)	FC1(input_size= 6000)	4.260	4.086
BN+ ReLU+Two FC (4096)	BN+ ReLU + FC1(input_size= 4096) + BN+ ReLU + FC2(input_size= 4096)	4.134	**3.446**
Two FC (4096) + BN+ ReLU	FC1(input_size= 4096) +BN+ ReLU + FC2(input_size= 4096) +BN+ ReLU	4.039	4.347
Two FC (4096) + ReLU	FC1(input_size= 4096) + ReLU + FC2(input_size= 4096) + ReLU	4.267	3.855
Two FC (4096)	FC1(input_size= 4096) + FC2(input_size= 4096)	4.092	3.976

Table 3 shows the results of Approach-2 for real-age estimation task on FG_NET dataset using classification and regression to encode age. The lowest value of MAE was achieved when using (Two FC (4096) + ReLU) with regression algorithm.

Figure 7 shows the results of classification and regression encoding for different proposed schemes. We observed that, when using Approach-1, the regression algorithm on average obtained lower MAE than classification algorithm. We can see the good impact of using Approach-2 with the structure of stacked FC with activation layers (Two FC (4096) + ReLU) and without activation layer (Two FC (4096)). Reusing the model with (No FC) scheme was the worst case for both approaches. When using the scheme (BN+ ReLU+ Two FC (4096)), Approach-1 obtained the least MAE. On scheme (Two FC (4096) + ReLU), the second Approach got the least MAE. Better results of regression over classification when fine-tuning the model on FG_NET are related to the biased classes on the dataset for the younger age and there is inadequate data to represent each class. Thus, learning the age as continuous values rather than discrete age labels is more efficient.

Table 3. Age estimation results on FG_NET using approach-2.

Scheme Name	MAE of Classification	MAE of Regression
No FC	5.852	6.542
FC1(1024)	4.050	4.179
FC1(5000)	4.198	4.441
FC1(6000)	**3.807**	3.845
BN+ ReLU+ Two FC (4096)	5.300	6.184
Two FC (4096) + BN+ ReLU	5.934	5.048
Two FC (4096) + ReLU	3.837	**3.545**
Two FC (4096)	3.986	3.618

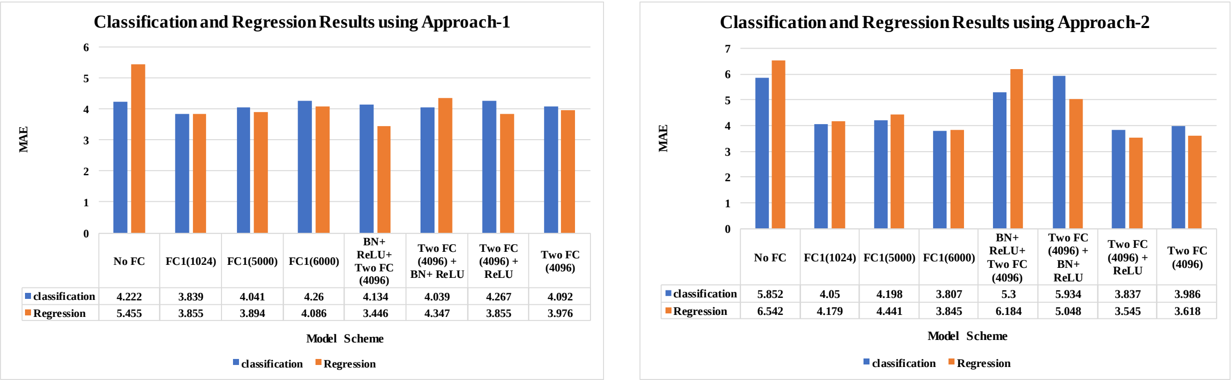


Figure 7. Age estimation results on FG_NET using the proposed approaches.

4.5.2 Results on the Unconstrained UTKFace Dataset

To make the model closer to real applications, we should evaluate it under rougher conditions. UTKFace is considered as an unconstrained dataset, where its images contain a diversity of head poses, illumination and occlusion. In this subsection, we show the results of evaluating the proposed approaches on the unconstrained UTKFace dataset. We used the aligned and cropped version of this dataset. For the case of classification, the number of nodes for the output layer was set to 101, which is related to the number of classes in the dataset. For regression, the number of nodes was set to 1.

Figure 8 shows the age estimation results when using Approach-1 and Approach-2 on UTKFace dataset. The results show the effectiveness of using classification algorithm with the proposed approaches to encode age. The lowest value of MAE was achieved by the scheme (Two FC (4096) + ReLU). It can be observed from the preceding experiments, how the performance of the model using Approach-2 outperforms that of Approach-1 for both encoding algorithms. Both approaches show an instable performance when using scheme (BN+ ReLU+ Two FC (4096)) and scheme (Two FC (4096) + BN+ ReLU) with regression algorithm, which is related to use a small batch size with large dataset, thus the

BN may have an adverse effect. The least MAE value was achieved when using the FC layers with ReLU activation for both approaches. It is noticed that approach-1 shows more stable performance than approach-2 across all schemes.

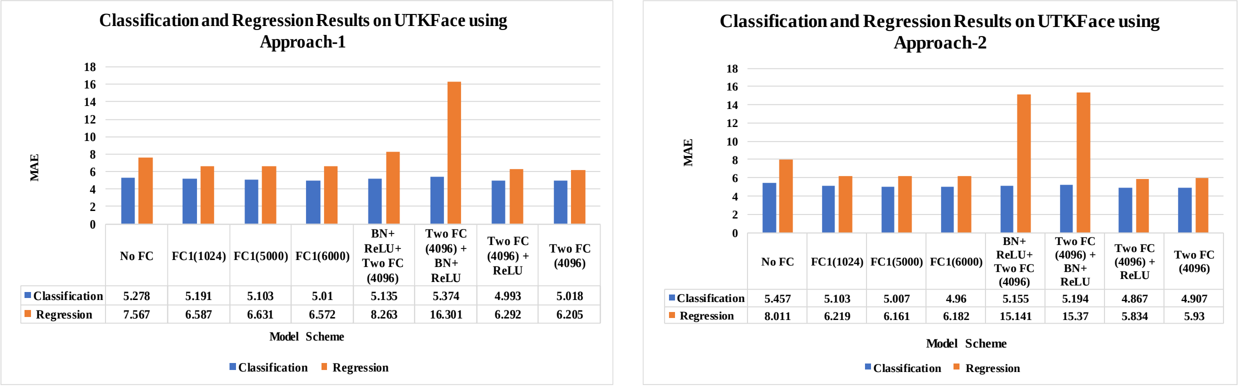


Figure 8. Age estimation results on UTKFace dataset using proposed approaches.

4.5.3 Results on the Unconstrained Adience Dataset

In this section, we show the results of reusing the base model with the second approach on Adience dataset for the age group classification task. In this experiment, the images were pre-processed by detecting and cropping the faces to remove the background. We fine-tuned the VGGFace using Approach-2 that showed a better performance than Approach-1, by connecting the GAP layer with different schemes to find the age group classification task. The output layer was set to 8 neurons related to the number of classes with Softmax as activation function. In this model, we considered accuracy as the evaluation metric. As shown in Figure 9, the classification for 8 classes of age groups needs a less complex, less deep network than the task of real-age estimation. The best accuracy was achieved by the base model without adding any extra layers, by directly connecting the GAP layer with the output layer.

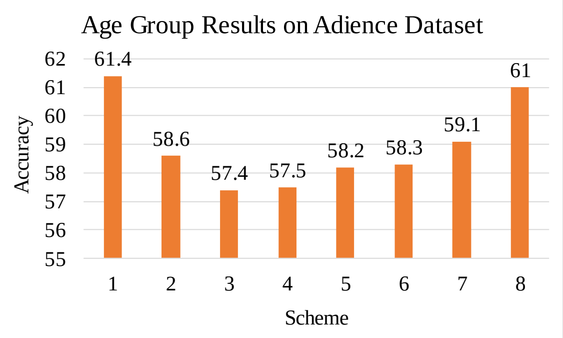


Figure 9. Age group classification results on Adience dataset using approach-2.

5. Discussion

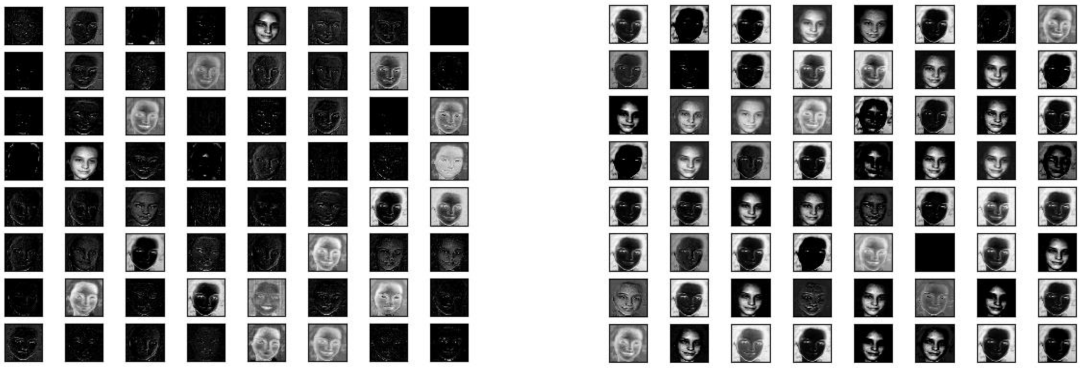
In this section, the proposed model is investigated according to some different criteria such as the effectiveness of specific domain transfer learning, the robustness and the complexity of the model. Moreover, some failure and success cases with more analysis are presented.

The effectiveness of using specific domain transfer learning for age estimation system: The typical structure of VGGFace is based on the pre-trained version of VGG16 network on VGGFace dataset for face recognition task. To show the efficiency of our proposed method, general domain VGG16 model pre-trained on ImageNet was tested. We tested VGG16 with approach-2 while using scheme (no FC). From the results shown in Table 4, we can see how approach-2 has profited from specific domain transfer learning compared with general domain for age estimation task.

We compare the pre-trained model of VGGface and VGG16 on how age estimation can benefit in extracting more features from face. The feature maps are visualized from the first block. Figure 10 shows the ability of the model VGGFace pre-trained on faces to consider and extract more features related to face. On the other side, we can see how VGG16 pre-trained on object task classification on the ImageNet dataset needs building a more complex network to be capable to extract more features related to face. From this point, we can conclude that using models that pre-trained on a task related to age such as face recognition is more effective than using models pre-trained on a general task.

Table 4. Results on FG_NET using specific and general domain pre-trained models.

Pre-training Strategy	Model	MAE
Proposed model (Specific domain)	VGGFace-Approach-2	5.852
Pre-trained model (General domain)	VGG16-Approach-2	6.479



Feature map from VGG16

Feature map from VGGFace

Figure 10. Feature's map from the second convolutional layer from VGG16 and VGGFace.

The complexity of the model: adding two FC layers means extra computational efforts and large number of parameters. Nowadays, the common direction tends to design less complex models that can be adapted to real application. For both datasets used, the system achieved reasonable results while using scheme (one FC with 6000 inputs) when encoding age using classification algorithm. In this case, the resulting models have lower sizes compared with models constructing using two FC layers, thus these models can be used in low memory devices that have a limited capacity, such as mobile applications.

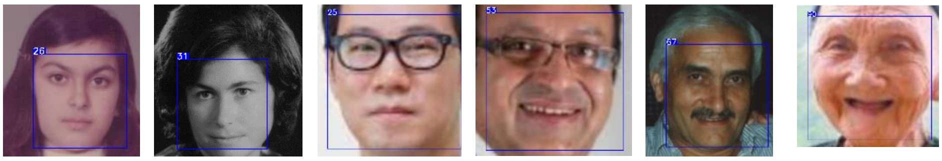
The robustness of the model: using data augmentation in deep neural networks increases the ability of the network to learn from the same images with different conditions that were created using different transformation functions. Moreover, testing the model using only constrained datasets with clear data does not add any extra benefit to the system when using it in real application. Thus, using unconstrained datasets such as UTKFace and Adience that contain a large diversity of illusion, facial expressions and occlusion improves the system robustness to real conditions. Also, age estimation system needs a large number of images that cover a large range of ages to achieve more robust system.

Success and failure cases of the model: we tested our best model on real age estimation using different images from both datasets (FG_NET, UTKFace). As shown in Figure 11, our model can successfully estimate the real age for both constrained and unconstrained images. In some cases, our system failed to find the real-age, where a person may look younger or older than his/her real age.

Successful Samples

Real age	1	3	25	50	63	73
Predicted age	1	5	26	51	66	72

Failed Samples



Real age	12	24	31	40	53	101
Predicted age	26	31	25	53	67	90

Figure 11. successful and failed samples tested by our best model.

6. Comparison with The-State-of-The-Art

Corresponding to the earlier experiments, transfer learning from a pre-trained model on a specific domain, such as face recognition, can improve age estimation performance and overcome the overfitting problem. In addition, correct selection of data augmentation can enhance performance. In this section, a comparison of the results for the proposed model with the-state-of-the-art is described. Table 5 summarizes the results of age estimation on the constrained FG_NET dataset with the-state-of-art. Some models such as [42] needed deeper network to reach reasonable results. Antipov et al. [29] benefited from pre-training on face recognition to improve the performance of real-age estimation. Chen et al. [39] pre-trained the ranking CNNs to extract the facial features on a small unconstrained dataset. Chang and Chen [38] used the hand-crafted features on their model to extract age features from facial images. A classification model for age estimation was proposed by Rothe et al. [27]. They used a VGG16 and pre-trained it on unconstrained IMDB_WIKI dataset for age estimation task. IMDB_WIKI dataset that contains about 500k images, which is a small number compared with the Face dataset that contains 2.6M images to pre-train the VGGFace.

Table 5. Results for age estimation on FG-NET dataset in various methods.

Ref.	Model	Base Model	Pre-training Strategy	MAE
Our proposed model	Regression_Approach-1 with Scheme:(BN+ ReLU + Two FC(4096)	VGGFace	Specific	**3.446**
	Regression_Approach-2 with Scheme: (Two FC(4096) +ReLU)	VGGFace	Specific	**3.545**
Zhang et al. (2018) [42]	Fine-Grained with attention mechanism	ResNets-34	Specific	2.39*
Li et al. (2019) [44]	BridgeNet (Local regressor + Gated network)	VGG-16	General	2.56*
Antipov et al. (2017) [29]	Soft classification + using LDAE	VGG-16	Specific	2.84*
Shang and Ai (2017) [41]	Feature Clustering using k-means++	GoogleNet	General	3.85*
Chen et al. (2018) [39]	Ranking-CNN	Shallow CNN	Training From scratch	4.13
Rothe et al. (2016) [27]	Classification + using DEX method	VGG16	Specific	4.63
Chang and Chen (2011) [38]	Ranking using Scattering Transform	Hand-crafted-based	-	4.48*

*These models used a different testing protocol; LeaveOnePersonOut (LOPO).

The results show the effectiveness of using specific domain pre-training when extracting features related to age. Moreover, the combination of GAP and FC layers in Approach-2 with scheme (Two FC (4096) +ReLU) has regularized the model and prevented the overfitting problem. Using the scheme (BN+ ReLU + Two FC(4096)) has also a good influence on model stability.

Table 6 summarizes the results of age estimation on the unconstrained UTKFace dataset with other models. Niu et al. [53] constructed their model of multi-output CNN by considering less number of layers than VGGFace. Cao et al. [54] used the VGG16 as the base network that was pre-trained on a general domain. It is clearly observed from the results that using VGGFace based on VGG16 for age estimation model is more efficient than using other models. For our proposed model, the least error was obtained when using classification as an age encoding algorithm while using Approach-2 for fine-tuning the base model with (Two FC (4096) +ReLU). It is noticed that our proposed model has obtained lower MAE when using classification than regression. This can be obviously related to that UTKFace is a large dataset with balanced distribution according to age. Thus, treating age as a multi- classification task will obtain good results if each class in the dataset has an adequate number of images with balanced distribution over a wide range of ages.

Table 6. Results for age estimation on UTKFace in various methods.

Ref.	Model	Base Model	Pre-training Strategy	MAE
Our proposed model	Classification_Approach-2 with Scheme: (Two FC (4096) +ReLU)	VGGFace	Specific	**4.867**
	Classification_Approach-1 with Scheme: (Two FC (4096) +ReLU)	VGGFace	Specific	**4.993**
Cao et. al. (2019) [54]	CORAL-CNN: Ordinal Regression	VGG-16	General	5.83
Niu et. al. (2016) [53]	Multiple Output CNN + Ordinal Regression	CNN	General	6.39

Table 7. Results of age group on Adience dataset in various methods.

Ref.	Model	Base Model	Pre-training Strategy	Accuracy
Our proposed model	Classification_Approach-2 with Scheme: No FC	VGGFace	Specific	**61.4**
Zhang et al. (2018) [42]	Fine-Grained + Attention	ResNets-152	Specific	67.83
Rothe et al. (2016) [27]	DEX method	VGG16	Specific	64.0
Rodríguez et al. (2017) [33]	Attention	VGGFace	Specific	61.8
Rodríguez et al. (2018) [55]	Attention	WRN	General	59.7
Chen et al. (2018) [39]	Ranking-CNN	Shallow CNN	Training From scratch	53.7
Levi and Hassner (2015) [52]	CNNs	Shallow CNN	Training From scratch	50.7

For age group classification task, as shown in Table 7, we achieved a good accuracy in spite of the simple fine-tuning scheme and the approach that were used in this experiment. Although Zhang et al. [42] achieved a higher accuracy than other methods, they used a very deep residual network with 152 layers compared with less deep network such as VGG with 16 layers which was the base model for [27] and was pre-trained on a large dataset. Rodríguez et al. *[55]* used a Wide Residual Network (WRN) pre-trained on a general domain task. For their previous work [33], despite adding the attention mechanism to improve the performance of VGGFace, they achieved a minor enhancement in accuracy compared with our model using the GAP layer that can reduce the overfitting problem. Other models just used shallow CNNs to extract age features.

7. CONCLUSIONS

This paper proposes an age estimation model based on VGGFace, which was pre-trained on a specific domain. Age is an attribute that is derived from face, thus reusing a model pre-trained on a related task to age can have the capability to extract discriminative features related to age effectively and avoid the overfitting problem when using limited data. In this work, we utilized different approaches for fine-tuning the basic VGGFace model. The first approach kept the top classification layers with the base convolutional layers and connected the model with different strategies of adding extra FC layers. The second approach joined the base convolutional layers with the GAP connected to different strategies of adding extra FC layers.

The proposed approaches were tested under different schemes, by varying feature size and model depth. We also investigated the effectiveness of two algorithms for encoding age: classification and regression. A minimum error was obtained when using a balanced distributed dataset like UTKFace, where each class is represented by enough data at a specific age. On the other hand, in case of unbalanced datasets like FG_NET, regression performance was better than classification performance. Furthermore, selecting appropriate data augmentation can improve performance, such as rotation, shearing and flipping. We evaluated our model using two kinds of datasets: constrained FG_NET and unconstrained UTKFace and Adience. Our model achieved state-of-the-art results on FG_NET when using regression to encode age, while the lowest MAE was obtained when using classification for UTKFace dataset. According to the age group classification task, the model was fine-tuned using the second approach and good results on Adience dataset were achieved despite the simple approach of fine-tuning that was implemented. The small degree of complexity to classify 8 age groups compared to large age range could explain the good performance that was obtained.

For future work, we will try to use a hybrid system that combines classification and regression in one model. The idea is to firstly classify the images into age groups, then for each group, a regression CNN will be trained to estimate age.

ACKNOWLEDGEMENTS

Computations for the work described in this paper were supported by King Abdulaziz University’s High-Performance Computing Centre (Aziz Supercomputer) (http://hpc.kau.edu.sa).

REFERENCES

[1] A. K. Jain, S. C. Dass and K. Nandakumar, "Soft Biometric Traits for Personal Recognition Systems," Proceedings of International Conference on Biometric Authentication, Berlin, Heidelberg, vol. 3072, pp. 731–738, 2004.

[2] Y. Fu, G. Guo and T. S. Huang, "Age Synthesis and Estimation via Faces: A Survey," IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 11, pp. 1955–1976, 2010.

[3] Y. LeCun, Y. Bengio and G. Hinton, "Deep Learning," Nature, vol. 521, no. 7553, pp. 436–444, May 2015.

[4] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," IEEE Trans. Knowl. Data Eng., vol. 22, no. 10, pp. 1345–1359, 2010.

[5] O. M. Parkhi, A. Vedaldi and A. Zisserman, "Deep Face Recognition," British Machine Vision Conference, 2015.

[6] T. Zheng, W. Deng and J. Hu, "Age Estimation Guided Convolutional Neural Network for Age-Invariant Face Recognition," Proc. of IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 503–511, 2017.

[7] T. F. Cootes, G. J. Edwards and C. J. Taylor, "Active Appearance Models," IEEE Trans. Pattern Anal. Mach. Intell., vol. 23, no. 6, pp. 681–685, 2001.

[8] G. Guo, T. S. Huang, Y. Fu and T. S. Huang, "Human Age Estimation Using Bio-inspired Features," Proc. of IEEE Conference on Computer Vision and Pattern Recognition, USA, pp. 112–119, 2009.

[9] T. Ahonen, A. Hadid and M. Pietikainen, "Face Description with Local Binary Patterns: Application to Face Recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 12, pp. 2037–2041, 2006.

[10] A. S. Al-Shannaq and L. A. Elrefaei, "Comprehensive Analysis of the Literature for Age Estimation From Facial Images," IEEE Access, vol. 7, pp. 93229–93249, 2019.

[11] Y. H. Kwon and N. da V. Lobo, "Age Classification from Facial Images," Comput. Vis. Image Underst., vol. 74, no. 1, pp. 1–21, Apr. 1999.

[12] A. Gunay and V. V Nabiyev, "Automatic Detection of Anthropometric Features from Facial Images," Proc. of the 15th IEEE Signal Processing and Communications Applications, pp. 1–4, 2007.

[13] M. M. Dehshibi and A. Bastanfard, "A New Algorithm for Age Rcognition from Facial Images," Signal Process., vol. 90, no. 8, pp. 2431–2444, Aug. 2010.

[14] G. Guo and G. Mu, "A Framework for Joint Estimation of Age, Gender and Ethnicity on a Large Database," Image Vis. Comput., vol. 32, no. 10, pp. 761–770, 2014.

[15] G. Guo and G. Mu, "Simultaneous Dimensionality Reduction and Human Age Estimation *via* Kernel Partial Least Squares Regression," CVPR 2011, pp. 657–664, 2011.

[16] H. Han, C. Otto and A. K. Jain, "Age Estimation from Face Images: Human *vs.* Machine Performance," Proc. of the 2013 Int. Conf. Biom. (ICB), pp. 1–8, 2013.

[17] X. Geng, Z. Zhou, S. Member, K. Smith-miles and S. Member, "Automatic Age Estimation Based on Facial Aging Patterns," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 12, pp. 2234–2240, 2007.

[18] X. Geng, Z.-H. Zhou, Y. Zhang, G. Li and H. Dai, "Learning from Facial Aging Patterns for Automatic Age Estimation," Proceedings of the 14th ACM International Conference on Multimedia, pp. 307–316, 2006.

[19] Y. Fu and T. S. Huang, "Human Age Estimation with Regression on Discriminative Aging Manifold," IEEE Trans. Multimed., vol. 10, no. 4, pp. 578–584, 2008.

[20] G. Guo, Y. Fu, C. R. Dyer and T. S. Huang, "Image-based Human Age Estimation by Manifold Learning and Locally Adjusted Robust Regression," IEEE Transactions on Image Processing, vol. 17, no. 7, pp. 1178–1188, 2008.

[21] S. E. Choi, Y. J. Lee, S. J. Lee, K. R. Park and J. Kim, "Age Estimation Using a Hierarchical Classifier Based on Global and Local Facial Features," Pattern Recognit., vol. 44, no. 6, pp. 1262–1281, 2011.

[22] E. Eidinger, R. Enbar and T. Hassner, "Age and Gender Estimation of Unfiltered Faces," IEEE Trans. Inf. FORENSICS Secur., vol. 9, no. 12, pp. 2170–2179, 2014.

[23] Z. Hu, Y. Wen, J. Wang and M. Wang, "Facial Age Estimation with Age Difference," IEEE Trans. IMAGE Process., vol. 7149, no. c, pp. 1–11, 2016.

[24] K. E. Zhang, C. E. Gao, L. Guo, M. Sun and S. Member, "Age Group and Gender Estimation in the Wild with Deep RoR Architecture," Proc. of Chinese Conference on Computer Vision (CCCV), vol. 5, no. Cccv, 2017.

[25] K. Li, J. Xing, W. Hu and S. J. Maybank, "D2C : Deep Cumulatively and Comparatively Learning for Human Age Estimation," Pattern Recognit., vol. 66, no. July 2016, pp. 95–105, 2017.

[26] O. Russakovsky et al., "ImageNet Large Scale Visual Recognition Challenge," Int. J. Comput. Vis. (IJCV), vol. 115, no. 3, pp. 211–252, 2015.

[27] R. Rothe, R. Timofte and L. Van Gool, "Deep Expectation of Real and Apparent Age from a Single Image without Facial Landmarks," Int. J. Comput. Vis., vol. 126, no. 2–4, pp. 144–157, 2016.

[28] H. Liu, J. Lu, J. Fenga and J. Zhou, "Group-aware Deep Feature Learning for Facial Age Estimation," Pattern Recognit., vol. 66, pp. 82–94, 2016.

[29] G. Antipov, M. Baccouche, S. Berrani and J. Dugelay, "Effective Training of Convolutional Neural Networks for Face-based Gender and Age Prediction," Pattern Recognit., vol. 72, pp. 15–26, 2017.

[30] M. Yang, S. Zhu, F. Lv and K. Yu, "Correspondence Driven Adaptation for Human Profile Recognition," Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 505–512, 2011.

[31] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Proc. of the International Conference on Learning Representations (ICLR), vol. abs/1409.1, 2015.

[32] Z. Hu, Y. Wen, J. Wang and M. Wang, "Facial Age Estimation with Age Difference," IEEE Trans. Image Process., vol. 7149, no. c, pp. 1–11, 2016.

[33] P. Rodríguez, G. Cucurull, J. M. Gonfaus, F. X. Roca and J. Gonzàlez, "Age and Gender Recognition in the Wild with Deep Attention," Pattern Recognit., vol. 72, pp. 563–571, Dec. 2017.

[34] A. Lanitis, C. Draganova and C. Christodoulou, "Comparing Different Classifiers for Automatic Age Estimation," IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 34, no. 1, pp. 621–628, Feb. 2004.

[35] K. Ueki, T. Hayashida and T. Kobayashi, "Subspace-based Age-group Classification Using Facial Images Under Various Lighting Conditions," Proc. of the 7th IEEE International Conference on Automatic Face and Gesture Recognition (FGR06), pp. 1-6, 2006.

[36] A. Lanitis, C. J. Taylor and T. F. Cootes, "Toward Automatic Simulation of Aging Effects on Face Images," IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 4, pp. 442–455, 2002.

[37] Y. Fu, Y. Xu and T. S. Huang, "Estimating Human Age by Manifold Analysis of Face Pictures and Regression on Aging Features," Proc. of IEEE International Conference on Multimedia and Expo, pp. 1383–1386, 2007.

[38] K. Chang and C. Chen, "A Learning Framework for Age Rank Estimation based on Face Images with Scattering Transform," IEEE Trans. Image Process., vol. 7149, no. c, pp. 1–14, 2015.

[39] S. Chen, C. Zhang and M. Dong, "Deep Age Estimation: From Classification to Ranking," IEEE Trans. Multimed., vol. 20, no. 8, 2018.

[40] K. Li, J. Xing, C. Su, W. Hu, Y. Zhang and S. Maybank, "Deep Cost-sensitive and Order-preserving Feature Learning for Cross-population Age Estimation," Proc. of IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, pp. 399–408, 2018.

[41] C. Shang and H. Ai, "Cluster Convolutional Neural Networks for Facial Age Estimation," Proceedings of International Conference on Image Processing (ICIP), pp. 1817–1821, 2017.

[42] K. Zhang, N. Liu, X. Yuan, X. Guo, C. Gao and Z. Zhao, "Fine-grained Age Estimation in the Wild with Attention LSTM Networks," CoRR, vol. abs/1805.1, pp. 1–12, 2018.

[43] H. Pan, H. Han, S. Shan and X. Chen, "Mean-variance Loss for Deep Age Estimation From a Face," Presented at the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, pp. 5285–5294, 2018.

[44] W. Li, J. Lu, J. Feng, C. Xu, J. Zhou and Q. Tian," BridgeNet: A Continuity-aware Probabilistic Network for Age Estimation," ArXiv190403358 Cs, Apr. 2019.

[45] M. Lin, Q. Chen and S. Yan, "Network In Network," ArXiv13124400 Cs, Dec. 2013.

[46] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," ArXiv150203167 Cs, Feb. 2015.

[47] M. Duan, K. Li and K. Li, "An Ensemble CNN2ELM for Age Estimation," IEEE Trans. Inf. Forensics Secur., vol. 13, no. 3, pp. 758–772, 2018.

[48] C. Shorten and T. M. Khoshgoftaar, "A Survey on Image Data Augmentation for Deep Learning," J. Big Data, vol. 6, no. 1, p. 60, Dec. 2019.

[49] C. M. Bishop, Pattern Recognition and Machine Learning, New York, Springer, 2006.

[50] A. Lanitis and T. Cootes, "Fg-net Aging Data Base," Cyprus Coll., 2002.

[51] Zhang Zhifei, Song, Yang and H. Qi, "Age Progression/Regression by Conditional Adversarial Autoencoder," Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[52] G. Levi and T. Hassner, "Age and Gender Classification using Convolutional Neural Networks," CVPR, vol. 24, no. 3, pp. 2622–2629, 2015.

[53] Z. Niu, M. Zhou, L. Wang, X. Gao and G. Hua, "Ordinal Regression with Multiple Output CNN for Age Estimation," Proc. of IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 4920–4928, 2016.

[54] W. Cao, V. Mirjalili and S. Raschka, "Rank-consistent Ordinal Regression for Neural Networks," ArXiv190107884 Cs Stat, Jan. 2019.

[55] P. Rodríguez, J. M. Gonfaus, G. Cucurull, F. X. Roca and J. Gonzàlez, "Attend and Rectify: a Gated Attention Mechanism for Fine-Grained Recovery," ArXiv180707320 Cs, Jul. 2018.

ملخص البحث:

في الوقت الحاضر، تزيد مشاركة الشبكات العصبية العميقة في رؤية الحاسوب من القدرة على تحقيق دقة أعلى في العديد من مهام التعلم مثل التعرف على الوجوه والكشف عنها. ومع ذلك، لا يزال التقدير التلقائي لعمر الإنسان مهمة الوجه الأكثر تحدياً والتي تتطلب بذل جهود إضافية للحصول على دقة مقبولة للتطبيق الحقيقي. في هذه الورقة، نحاول الحصول على نموذج مُرْضٍ يتغلب على مشكلة التهيئة الزائدة (overfitting)، من خلال ضبط نموذج شبكات عصبية التفافية (CNNs) تم اختباره مسبقًاً في مهمة التعرف على الوجوه لتقدير العمر الحقيقي. لجعل النموذج أكثر قوة، قمنا بتقييم النموذج الخاص بتقدير العمر الحقيقي على نوعين من مجموعات البيانات: علي مجموعة البيانات FG_NET المقيدة حققنا 3.446 MAE، وعلى مجموعة البيانات UTKFace غير المقيده حققنا 4.867 MAE. وتتفوق النتائج التجريبية لنهجنا في نماذج تقدير العمر الحديثة على مجموعات البيانات المرجعية. وقمنا بضبط نموذج لمهمة تصنيف مجموعات العمر على مجموعة البيانات Adience، ونموذجنا حقق نتيجة جيدة.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 02, June 2020.

AN EIGEN-APPROACH FOR JOINT ESTIMATION OF DIRECTION-OF-ARRIVAL AND FREQUENCY OF AN UNKNOWN NUMBER OF SIGNALS

Thabet Mismar

(Received: 11-Nov.-2019, Revised: 31-Dec.-2019 and 8-Jan.-2020, Accepted: 3-Feb.-2020)

ABSTRACT

An accurate method for estimating the direction-of-arrival (DOA) jointly with the frequencies of an unknown number of source signals is proposed using the Eigen-approach. Using the minimum eigenvalues of the autocorrelation matrices produces both the DOA and the corresponding frequencies.

By moving the roots produced from the eigenvector one-by-one, the angular location is first found. The frequency is then estimated using the same procedure. Finally, the frequency is used with the angular location to estimate the DOA angle.

The results show an accurate estimation of source signals' DOA and frequency in the presence of different levels of noise.

KEYWORDS

Direction-of-arrival (DOA), Antenna arrays, Eigenvalues, Eigenvectors, Frequency estimation.

1. INTRODUCTION

Generally, the antenna array factor is designed to receive the desired signal from a particular direction while suppressing the undesired signals. Therefore, the direction of arrival (DOA) of the received source signals needs to be estimated [1].

Many DOA estimation methods have been proposed. The Eigen-approach has received wide attention, since it gives high accuracy results [2]-[6]. Most of the proposed methods needed the exact number of sources to separate signals from noise. However, the exact number of sources is typically an unknown value. Additionally, the proposed methods focused on DOA without regard to frequency estimation [2]-[4], [6].

Others have proposed different techniques to jointly estimate the DOA and the related source frequency [7]-[13]. The extended Kalman filter and unscented Kalman filter were utilized in [7] to jointly estimate the DOA and frequency of source signals. Unfortunately, high computational iterations were needed to realize good results.

Some used ESPRIT, MUSIC and Maximum Likelihood Estimation (MLE) to estimate the DOA [4], [14]-[16]. The results achieved were good, but they suffer from the complexity of the problem and the need for an exact number of sources. Although others [17] proposed methods to estimate the number of signal sources, additional computations to find the number of sources make the Eigen-approach proposed here faster, where the number of sources is found while performing other computations.

A general comparison between different DOA estimation algorithms was discussed in [18]-[19]. The methods compared in [18, 19] only discussed DOA estimation with no mention of frequency estimation, since they assume a single frequency or a known set of frequencies.

Since the DOA estimation is a nonlinear optimization problem, random search algorithms were proposed to estimate the DOA. In particular, genetic algorithm (GA) was used directly or in conjunction with other techniques to estimate the DOA [20].

In this paper, the Eigen-approach is used to find the array factor with the coefficients of the eigenvector corresponding to the minimum eigenvalue of the autocorrelation matrix, which produces the minimum output power of the array. These coefficients are represented by the roots of the polynomial lying on the

T. Mismar is with Department of Networks and Communication Engineering, Al-Ain University, Abu Dhabi, UAE. E-mail: thabet.mismar@aau.ac.ae

unit circle [21]. By moving the roots one by one, the proposed method estimates the DOA and the frequency using the array output.

Several simulations were carried out to show that this proposed method estimates the DOA, frequency and number of source signals accurately.

2. PROBLEM FORMULATION

Assuming an unknown number of sources (M), with different arrival angles θ_m and different frequencies f_m, transmitted to a uniform linear antenna array of $(N+1)$ elements such that $(N \geq M)$, the received signal at each array element $x_n(k)$ consists of the combination of the narrowband source signals $\vec{\boldsymbol{s}}(k)$ with additive white Gaussian noise $\vec{\boldsymbol{n}}(k)$:

$$\vec{\boldsymbol{x}}(k) = \vec{\boldsymbol{A}}.\vec{\boldsymbol{s}}(k) + \vec{\boldsymbol{n}}(k) = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ e^{-j\varphi_1} & e^{-j\varphi_2} & \cdots & e^{-j\varphi_M} \\ \vdots & & \ddots & \vdots \\ e^{-jN\varphi_1} & e^{-jN\varphi_2} & \cdots & e^{-jN\varphi_M} \end{bmatrix} \begin{bmatrix} s_1(k) \\ s_2(k) \\ \vdots \\ s_M(k) \end{bmatrix} + \vec{\boldsymbol{n}}(k) \qquad (1)$$

where $\varphi_m = 2\pi \frac{d}{c} f_m \sin\theta_m$, d is the distance between any two consecutive antenna array elements and c is the speed of electromagnetic waves.

The signal $s_m(k)$ represents the *k*th sample of the source signal *m* and $n_n(k)$ represents the kth sample of the noise at array element *n*. The array covariance matrix can be expressed as:

$$\vec{\boldsymbol{R}}_{xx} = \vec{\boldsymbol{A}}(\varphi)\vec{\boldsymbol{R}}_{ss}\vec{\boldsymbol{A}}^H(\varphi) + \sigma^2\vec{\boldsymbol{I}}_{(N+1)} \qquad (2)$$

where $\vec{\boldsymbol{R}}_{ss} = E[\vec{\boldsymbol{s}}\vec{\boldsymbol{s}}^H]$ is the correlation matrix of the source signals, σ^2 is the power of the uncorrelated white Gaussian noise and $\vec{\boldsymbol{I}}_{(N+1)}$ is the identity matrix of size $(N+1)$x$(N+1)$. The output signal of the array is:

$$\vec{\boldsymbol{y}}(k) = \vec{\boldsymbol{w}}\vec{\boldsymbol{x}}(k) = \vec{\boldsymbol{w}}\vec{\boldsymbol{A}}\vec{\boldsymbol{s}}(k) + \vec{\boldsymbol{w}}\vec{\boldsymbol{n}}(k) \qquad (3)$$

where $\vec{\boldsymbol{w}} = [w_1 \quad w_2 \quad \cdots \quad w_m \quad \cdots \quad w_{N+1}]$ is the weight vector of the array elements. The average output power of the array is estimated as the time average correlation of *K* samples by:

$$P_y = \vec{\boldsymbol{w}}\vec{\boldsymbol{R}}_{xx}\vec{\boldsymbol{w}}^H \qquad (4)$$

and

$$\vec{\boldsymbol{R}}_{xx} = \frac{1}{K}\sum_{k=1}^{K} \vec{\boldsymbol{x}}(k)\vec{\boldsymbol{x}}(k)^H \qquad (5)$$

When the nulls of the array factor (roots of the polynomial $((\vec{w}\vec{A}))$ on the unit circle are matched to φ_m of the source signals, the output of the array will correspond to the uncorrelated noise power only.

$$P_0 = \vec{\boldsymbol{w}}_{\boldsymbol{0}}\vec{\boldsymbol{R}}_{xx}\vec{\boldsymbol{w}}_{\boldsymbol{0}}^H = \sigma^2\vec{\boldsymbol{w}}_{\boldsymbol{0}}\vec{\boldsymbol{w}}_{\boldsymbol{0}}^H \qquad (6)$$

where $\vec{\boldsymbol{w}}_{\boldsymbol{0}}$ is the weight vector which eliminates the signals at the array output.

The optimization problem is defined as follows:

$$\min_{w} \quad \vec{\boldsymbol{w}}\vec{\boldsymbol{R}}_{xx}\vec{\boldsymbol{w}}^H \qquad (7)$$

Subject to:

$$\vec{\boldsymbol{w}}\,\vec{\boldsymbol{w}}^H = 1;$$

The solution $\vec{\boldsymbol{w}}$ for the optimization problem can be found using a Lagrange multiplier; $\vec{\boldsymbol{w}}$ is a set of eigenvectors of the autocorrelation matrix R_{xx} with corresponding eigenvalues $\vec{\boldsymbol{\lambda}}$.

To minimize the objective function in Equation (7), the eigenvector $\vec{\boldsymbol{w}}_{\boldsymbol{0}}$ should be chosen such that it corresponds to the minimum eigenvalue λ_{min} . The corresponding minimum output power P_0 will be:

$$P_0 = \lambda_{min} = \vec{\boldsymbol{w}}_{\boldsymbol{0}}^H\vec{\boldsymbol{R}}_{xx}\vec{\boldsymbol{w}}_{\boldsymbol{0}} = \sigma^2 \qquad (8)$$

It was shown in [21] that the roots of the array polynomial made by $\vec{\boldsymbol{w}}_{\boldsymbol{0}}$ coincided with φ_m for minimum output power. If w_{0n} represents the *n*th element of the eigenvector $\vec{\boldsymbol{w}}_{\boldsymbol{0}}$, the array factor can then be written as:

$$F(f,\theta) = \sum_{n=0}^{N} w_{0n} e^{-jn\varphi} = \sum_{n=0}^{N} w_{0n} e^{-j2n\pi\frac{d}{c}f\sin\theta} = \prod_{n=1}^{N} w_{0N}\left(e^{-j2\pi\frac{d}{c}f\sin\theta} - e^{-j\hat{\varphi}_n}\right) \tag{9}$$

The above equation shows that $\hat{\varphi}_n$ match φ_m of the signals as the solution is achieved for the minimum output power.

Similarly, an adaptive FIR filter of order (L) and weight vector $\vec{\boldsymbol{\eta}}$ is used to determine the frequency content of the output signal of the array. The transfer function of the FIR filter is:

$$\frac{Z(f)}{Y(f)} = H(f) = \sum_{l=0}^{L} \eta_l\, e^{-jl\psi} = \prod_{l=1}^{L} \eta_L\left(e^{-j2\pi\frac{f}{f_s}} - e^{-j\hat{\psi}}\right) \tag{10}$$

where $\psi_m = 2\pi\frac{f_m}{f_s}$ and f_s is the sampling frequency.

The output of the filter is:

$$\vec{\boldsymbol{z}}(k) = \sum_{l=0}^{L} \eta_l\, \vec{\boldsymbol{y}}(k-l) = \vec{\boldsymbol{\eta}} \begin{bmatrix} y(k) \\ y(k-1) \\ \vdots \\ y(k-L) \end{bmatrix} = \vec{\boldsymbol{\eta}}\vec{\boldsymbol{y}}(l,k). \tag{11}$$

The average output power of the filter can be estimated as:

$$P_z = \vec{\boldsymbol{z}}\vec{\boldsymbol{z}}^H = \vec{\boldsymbol{\eta}}\vec{\boldsymbol{R}}_{yy}\vec{\boldsymbol{\eta}}^H \tag{12}$$

where,

$$\vec{\boldsymbol{R}}_{yy} = \frac{1}{K}\sum_{k=1}^{K} \vec{\boldsymbol{y}}(l,k)\vec{\boldsymbol{y}}(l,k)^H. \tag{13}$$

When the nulls of $H(f)$ on the unit circle are matched to ψ_m of $Y(f)$, the output of the filter will correspond to the uncorrelated noise power only.

$$P_{z_0} = \vec{\boldsymbol{\eta}}_{\boldsymbol{0}}\vec{\boldsymbol{R}}_{yy}\vec{\boldsymbol{\eta}}_{\boldsymbol{0}}^H = \sigma_1^2\vec{\boldsymbol{\eta}}_{\boldsymbol{0}}\vec{\boldsymbol{\eta}}_{\boldsymbol{0}}^H \tag{14}$$

where $\vec{\boldsymbol{\eta}}_{\boldsymbol{0}}$ is the weight vector which eliminates the signals at the filter output and σ_1^2 is the power of the noise signal at the input of the filter.

Similarly, the optimization problem is defined as:

$$\min_{\eta} \quad \vec{\boldsymbol{\eta}}\vec{\boldsymbol{R}}_{yy}\vec{\boldsymbol{\eta}}^H \tag{15}$$

subject to

$$\vec{\boldsymbol{\eta}}\vec{\boldsymbol{\eta}}^H = 1$$

The eigenvector η_0 that corresponds to the minimum eigenvalue yields the minimum output power as in Equation (14) and is related to ψ_m as in Equation (10).

3. DOA and Frequency Estimation

The process of frequency and DOA estimation can be obtained by calculating the pseudo-spectrum at the angles and frequencies corresponding to the polynomial roots as in Equations (9, 10). The pseudo-spectrum is calculated by altering each root of the array factor and the FIR filter to obtain the corresponding output power of the array and the filter. Large variation of the output power will occur if the roots coincide with an actual angle and frequency of a signal; otherwise, this root does not correspond to any of the source signals.

The method for estimating the frequency and DOA is as follows:

(1) The eigenvalues and the eigenvectors of $\vec{\boldsymbol{R}}_{xx}$ are calculated.

(2) An eigenvector ($\vec{\boldsymbol{w}}_{\boldsymbol{0}}$), corresponding to the minimum eigenvalue (λ_{min}), is used in Equation (9) to calculate $\hat{\boldsymbol{\varphi}}_{\boldsymbol{0}} = [\hat{\varphi}_1, \hat{\varphi}_2, \ldots, \hat{\varphi}_N]$ on the unit circle.

(3) The power of the uncorrelated noise is calculated as in Equation (8).

(4) For $n = 1,2,\ldots,N$:

(a) Shifting one $\hat{\varphi}_n$ on the unit circle by π; i.e., $\left(\hat{\varphi}_{n,new} = \hat{\varphi}_{n,old} + \pi\right)$ and computing the corresponding weight vector $\overrightarrow{\boldsymbol{w_n}}$ using Equation (9).

(b) The output power $P_y(n)$ is calculated as in Equation (4).

(c) If $P_y(n) \leq P_0$, go to step (4-a).

(d) The output signal is calculated as in Equation (3).

(e) The eigenvalues and the eigenvectors of $\overrightarrow{\boldsymbol{R}}_{\boldsymbol{yy}}$ (Equation (13)) are calculated.

(f) The eigenvector $(\overrightarrow{\boldsymbol{\eta_0}})$, corresponding to the minimum eigenvalue, is used in Equation (10) to calculate $\widehat{\boldsymbol{\psi}}_{\boldsymbol{0}} = [\hat{\psi}_1, \hat{\psi}_2, \ldots, \hat{\psi}_L]$ on the unit circle.

(g) For $l = 1,2,\ldots,L$:

a. Shifting one $\hat{\psi}_l$ on the unit circle by π; i.e., $\left(\hat{\psi}_{l,new} = \hat{\psi}_{l,old} + \pi\right)$ and computing the corresponding weight vector $\overline{\eta_l}$ using Equation (10).

b. The output power $P_z(l)$ is calculated as in Equation (12).

c. If $P_z(l) \leq P_{z_0}$, go to step (4-g).

d. Set values for pseudo-spectrum plot as:

$$\widehat{P_z}(l) = P_z(l) - P_{z_0}$$

$$\hat{f}(l) = \frac{\hat{\psi}_{l,old}}{2\pi/f_s}$$

$$\hat{\theta}(l) = \sin^{-1}\left[\frac{\hat{\varphi}_{n,old}}{2\pi\frac{d}{c}\hat{f}(l)}\right]$$

e. Go to step (4-g)

f. Go to step (4)

g. Plot pseudo spectrum $\widehat{P_z}(l) = F(^{\hat{f}(l)}/_{f_s}, \hat{\theta}(l))$

4. Simulation and Results

An array of 11 elements and an inter-element spacing of $d = {^c}/_{f_s}$ was chosen to simulate the proposed method. Since the number of roots is one less than the number of elements, the number of roots will be ten. This means that this antenna array can be used to estimate the location and frequency of up to ten signal sources.

Results are presented for two simulation examples by implementing the algorithm proposed in the previous section. In the first example, the noise power was -10dBm with seven narrowband source signals transmitting to the array with normalized frequencies $f = (f_m/f_s) = \{0.26946, 0.2, 0.1, 0.3, 0.35, 0.15,\ 0.4\}$ and angles $\theta = \{40, 60, 50, 50, -15, -30,\ -22\}$.

Note that:

$$\psi_1 = \psi_2$$

$$f_1 \sin\theta_1 = f_2 \sin\theta_2$$

and that,

$$\theta_3 = \theta_4$$

to show the capabilities of the proposed method to resolve signals that appear to have the same location.

The results are shown in Figure 1 and Table 1. The actual signal power can be evaluated by subtracting the estimated noise level from the level of power at each frequency in the pseudo-spectrum. Table 1 shows the power levels, the estimated DOA and the estimated frequency for the seven source signals. The results show that the proposed method was able to estimate the angle, the frequency and power level of each source signal accurately without knowing the number of source signals.

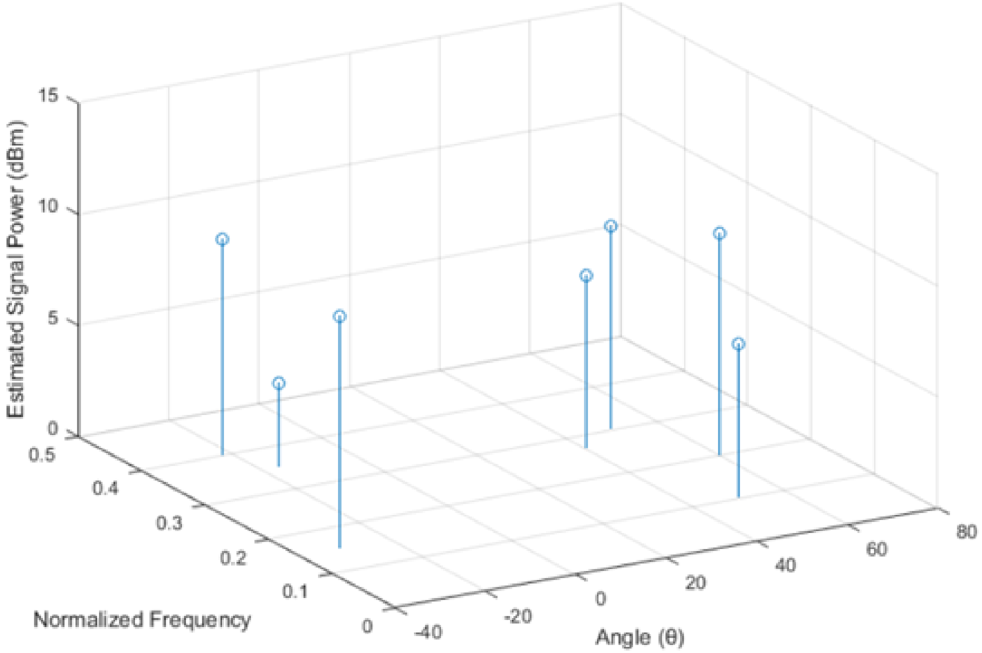


Figure 1. The estimated frequency and angle in the pseudo-spectrum for seven source signals. The source signals are transmitted with normalized frequencies of 0.26946, 0.2, 0.1, 0.3, 0.35, 0.15 and 0.4 at angles of 40°, 60°, 50°, 50°, -15°, -30° and -22°, respectively with different power levels.

Table 1. The estimated angles $\widehat{\theta_i}$, power levels $\widehat{P_i}$ and frequency $\widehat{f_i}$ using the proposed method. The number of source signals is 6, the number of array elements is 11 and the noise power is -10dBm.

θ_i	$P_i\ (dBm)$	f_i	$\widehat{\theta_i}$	$\widehat{P_i}\ (dBm)$	$\widehat{f_i}$
40	7.7815	0.26946	39.9599	7.7767	0.26946
60	10	0.2	59.9163	9.9999	0.2
50	6.9897	0.1	49.8815	6.9893	0.1
50	9.0309	0.3	49.8968	9.1995	0.3
-15	6.9897	0.35	-16.536	3.8268	0.3504
-30	9.0309	0.15	-31.318	10.4606	0.1502
-22	10	0.4	-22.203	9.7626	0.3998

Table 2. The estimated angles $\widehat{\theta_i}$, power levels $\widehat{P_i}$ and frequency $\widehat{f_i}$ using the proposed method. The number of source signals is 6, the number of array elements is 11 and the noise power is 7dBm.

θ_i	$P_i\ (dBm)$	f_i	$\widehat{\theta_i}$	$\widehat{P_i}\ (dBm)$	$\hat{f_i}$
40	7.7815	0.26946	42.44	7.293	0.2706
60	10	0.2	66.04	9.14	0.1998
50	6.9897	0.1	53.33	7.885	0.1005
50	9.0309	0.3	50.13	8.947	0.2997
-15	6.9897	0.35	-13.05	5.731	0.3507
-30	9.0309	0.15	-31.83	9.284	0.1501
-22	10	0.4	-22.26	9.884	0.4007

For the second example, the noise power was increased to 7dBm to show the effect of the noise on the proposed method. The results are shown in Table 2.

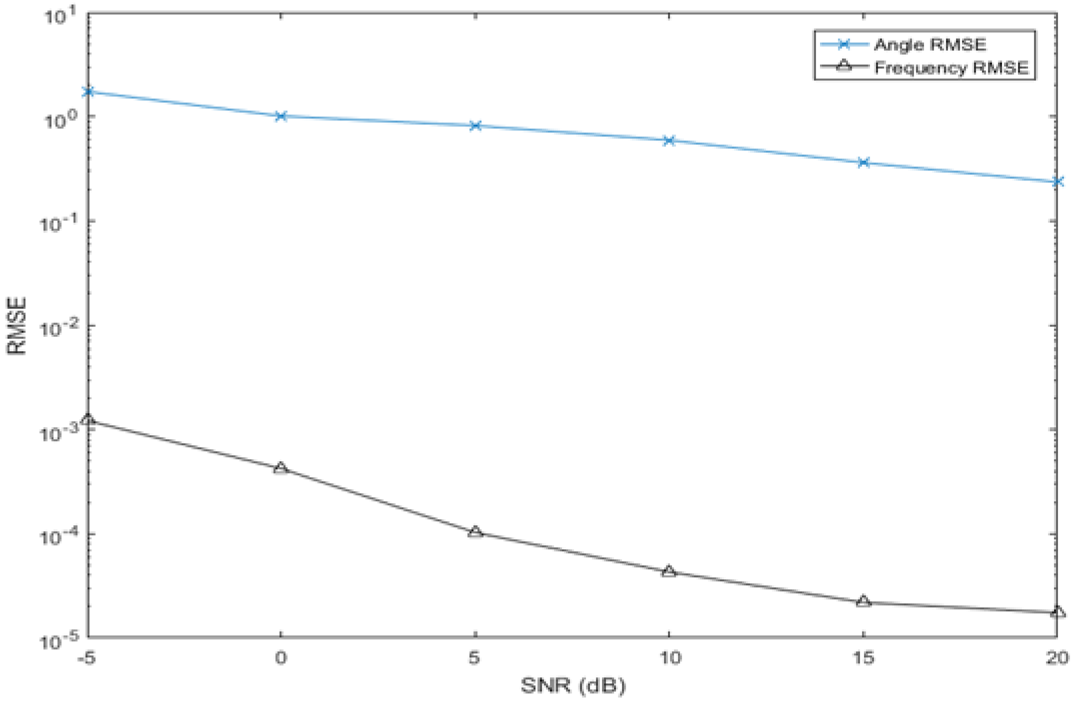


Figure 2. The root mean square error in estimating the angle $\widehat{\theta_i}$ and the frequency $\widehat{f_i}$ using the proposed method for different SNR levels.

5. Conclusion

The direction-of-arrival angles and the source signals’ frequencies were estimated using Eigen-approach with no prior knowledge of the number of signals. The proposed method first found the minimum eigenvalue of the autocorrelation matrix of the array elements input signals. The eigenvector, which corresponds to the minimum eigenvalue, represents the weights of the array factor. The output of the first stage yields the values of the angular location $(f \sin \theta)$, while the output of the second stage yields the source signal frequencies which are used to find the DOA angles using the angular locations from the first stage. The proposed method was able to handle different levels of noise to effectively find the DOA angle and source signal frequency.

References

[1] L. Godara, "Application of Antenna Arrays to Mobile Communications, Part II: Beam-Forming and Direction-of-Arrival Considerations," IEEE Proceedings of the IEEE, vol. 85 (8), pp. 1195-1245, 1997.

[2] Z. Xiaofei, L. Wen, S. Ying, Z. Ruina and X. Dazhuan, "A Novel DOA Estimation Algorithm Based on Eigen Space", Proc. of 2007 International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications, Hangzhou, China, 2007.

[3] R. Hudson and K. Yao, "A New Eigenvector-based 3D Wideband Acoustic DOA Estimator," Proc. of IEEE International Symposium on Phased Array Systems and Technology (PAST), Waltham, MA, USA, 2016.

[4] P. White, "Eigen-based DOA Estimators for Non-linear Array Configurations," Proc. of the International Conference on Acoustics, Speech and Signal Processing, Albuquerque, NM, USA, 1990.

[5] A. Liu, X. Zhang, Q. Yang and W. Deng, "Fast DOA Estimation Algorithms for Sparse Uniform Linear Array With Multiple Integer Frequencies," IEEE Access, vol. 6, pp. 29952-29965, Available: 10.1109/access.2018.2842262, 2018.

[6] D. Abu-Al-Nadi, M. Mismar and T. Ismail, "An Eigen-approach for Estimating the Direction-of Arrival (DOA) of Unknown Number of Signals," International Journal of Electrical and Computer Engineering, vol. 10, no. 9, pp. 1245-1248, Available: 10.5281/zenodo.1126702, 2016.

[7] S. Elaraby, H. Soliman, H. Abdel-Atty and M. Mohamed, "Joint 2D-DOA and Carrier Frequency Estimation Technique Using Nonlinear Kalman Filters for Cognitive Radio," IEEE Access, vol. 5, pp. 25097-25109, Available: 10.1109/access.2017.2768221, 2017.

[8] D. Ariananda and G. Leus, "Compressive Joint Angular-frequency Power Spectrum Estimation," Proc. of the 21st European Signal Processing Conference (EUSIPCO 2013), Marrakech, Morocco, 2013.

[9] D. Ariananda and G. Leus, "Compressive Joint Angular-frequency Power Spectrum Estimation for Correlated Sources," Prosiding Seminar Sistem Telekomunikasi Dan Informasi (SSTI), Jakarta, 2014.

[10] D. Ariananda, D. Romero and G. Leus, "Compressive Angular and Frequency Periodogram Reconstruction for Multiband Signals," Proc. of the 5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), St. Martin, France, 2013.

[11] A. Kumar, S. Razul and C. See, "Carrier Frequency and Direction of Arrival Estimation with Nested Sub-Nyquist Sensor Array Receiver," Proc. of the 23rd European Signal Processing Conference (EUSIPCO), Nice, France, 2015.

[12] A. Anil Kumar, S. Razul and C. See, "Spectrum Blind Reconstruction and Direction of Arrival Estimation of Multi-band Signals at Sub-Nyquist Sampling Rates," Multidimensional Systems and Signal Processing, vol. 29, no. 2, pp. 643-669, Available: 10.1007/s11045-016-0455-7, 2016.

[13] X. Yang, X. Wu, S. Li and T. Sarkar, "A Fast and Robust DOA Estimation Method Based on JSVD for Co-Prime Array," IEEE Access, vol. 6, pp. 41697-41705, Available: 10.1109/access.2018.2860680, 2018.

[14] M. Wax and A. Leshem, "Joint Estimation of Time Delays and Directions of Arrival of Multiple Reflections of a Known Signal," IEEE Trans. on Signal Processing, vol. 45, no. 10, pp. 2477-2484, Oct.1997.

[15] M. Manzanoa, J. Valenzuela-Valdesb, I. Castroc and L. Landesac, "Looking in Complex Angles for Improving the Accuracy of Antenna Array DoA Estimation," Journal of Electromagnetic Waves and Applications, vol. 27, no. 3, pp. 345-354, 2013.

[16] I. Ziskind and M. Wax, "Maximum Likelihood Localization of Multiple Sources by Alternating Projection," IEEE Trans. on Acoustics, Speech and Signal Processing, vol. 36, no. 10, pp. 1553-1560, Oct.1988.

[17] T. Salman, A. Badawy, T. M. Elfouly, A. Mohamed and T. Khattab, "Estimating the Number of Sources: An Efficient Maximization Approach," Proc. of the International Wireless Communications and Mobile Computing Conference (IWCMC), Dubrovnik, Croatia, 24-28 Aug. 2015.

[18] E. Sirignano, A. Davoli, G. M. Vitetta and F. Viappiani, "A Comparative Analysis of Deterministic Detection and Estimation Techniques for MIMO SFCW Radars," IEEE Access, vol. 7, pp. 129848-129861, 2019.

[19] J. Sanson, A. Gameiro, D. Castanheira and P. Monteiro, "Comparison of DoA Algorithms for MIMO OFDM Radar," Proceedings of the 15th European Radar Conference, Madrid, Spain, 26–28 Sept 2018.

[20] M. J. Mismar and T. H. Ismail, "DOA and Power Estimation by Controlling the Roots of the Antenna Array Polynomial," Progress in Electromagnetics Research M, vol. 46, pp. 193-201, Available: 10.2528/pierm16011604, 2016.

[21] Makhoul, "On the Eigenvectors of Symmetric Toeplitz/Matrices," IEEE Trans. on Acoustics, Speech and Signal Processing, vol. ASSP.29, no. 4, pp. 868-872, Aug.1981.

ملخص البحث:

تم اقتراح طريقة دقيقة لتقدير اتجاه الوصول جنباً الى جنب مع الترددات لعدد غير معلوم من الإشارات من مصادر مختلفة.

وتستخدم الطريقة المقترحة التي تعتمد النهج الذاتي أقّل القيم الذاتية لمصفوفات الارتباط الذاتي من أجل الحصول على كل من اتجاهات الوصول والترددات المناظرة لها.

وبتحريك الجذور المتولّدة من المتّجه الذاتي واحداً تلو الآخر، يتم أولاً إيجاد الموقع الزاوية لتقدير زاوية اتجاه الوصول. وتبين التقنية المقترحة تقديراً دقيقاً لاتجاهات وصول الإشارات وتردداتها في وجود مستويات مختلفة من الضجيج.

AR2B: FORMALIZATION OF ARABIC TEXTS WITH EVENT-B

Kheira-Zineb Bousmaha Ossoukine[1] and Lamia Belguith Hadrich[2]

(Received: 11-Nov.-2019, Revised: 31-Dec.-2019, Accepted: 3-Feb.-2020)

ABSTRACT

Transforming natural software requirements into a more formal specification is difficult and may be an excellent application for natural language processing. This problem is not recent. It aroused and still arouses great interest, because it gives rise to many challenges in various scientific fields, such as automatic language processing, requirements engineering, knowledge representation and formal verification. This paper proposes a platform and a strategy to transform software requirements specified to formal specification with event-B. The texts used are those of Arabic language, which is really a challenge. The Ar2B system is built and the experiments showed good results with an accuracy of 70%.

KEYWORDS

Arabic natural language processing, Information system, Information extraction, Formals' specification, Event-B.

1. INTRODUCTION

One of the challenges of Natural Language Processing (NLP) is the understanding of texts and their interpretation. To theorize the meaning of a text and automatically reach a level of understanding is notoriously difficult. This ambitious objective has been regularly adjourned to more local and less complex tasks. One of its axes is the formalization of the text describing the specifications of the requirements of the information system (IS).

Conceptual modeling of data is a very important phase in any development of the IS. Current design methods are based on models and data processing using various formalisms (graphs, entity / relation, UML diagrams, algorithmic notation, object representation…etc.). They are a factor in reducing costs and delays. The choice of a representation model that is sufficiently formal, precise and expressive to represent the semantics of natural language specifications allows an automated transition to formal specifications. The semi-formal and formal models often coexist in the same project, because they are complementary and each of them compensates for the disadvantages of the other, as they allow for better distribution and automation of tasks.

Modeling platforms will now be able to be used to make code generation or formal verification as well as moving back and forth between the code and the model without loss of information [1]. Automating the design and formalization has become a considerable activity which gives rise to many challenges in different scientific fields, such as requirements engineering, automatic language processing, information retrieval, representation and engineering of knowledge.

Several works have been interested in this topic which continues to attract much interest in more recent research, aiming to process more specifications in a shorter time and less subjective than an expert who relies solely on his knowledge and skills [2]. Our objective is to propose an approach to design a platform and develop a strategy to formalize the functional specification text to event-B. The originality of our research lies in the choice of the language proposed for study, the Arabic language, to which no work on this theme has been devoted. We offer assistance that can help in the processes of formalization and conceptual modeling based on reliable methods and tools. We propose a platform Ar2B (Figure 1) dedicated to Arabic Natural Language Processing (ANLP). We proceed to a linguistic treatment, conceptualization and formalization of text, oriented towards the conceptual modeling of information

[1] K. Z Bousmaha Ossoukine is with Department of Computer Science, RIIR laboratory, University Oran1 Algeria. Member of ANLP-RG (Arabic Natural Language Processing – Reseach Group), Emails: kzbousmaha@univ-oran1.dz; kzbousmaha@yahoo.fr

[2] L. Belguith Hadrich is with Departmant of Computer Science, Faculty of Economics and Management (FSEGS), University of Sfax,Tunisia, Head of ANLP-RG, Emails: l.belguith@fsegs.rnu.tn; lamia.belguith@gmail.com

systems (ISs). We provide a set of tools and techniques for the transition from the informal to the formal, knowing that no work in that direction has been ever done with Arabic text.

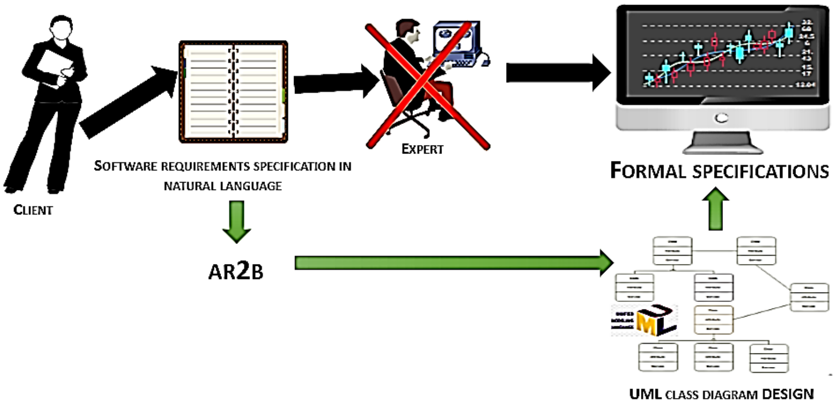


Figure 1. Automation of formalizing requirements.

An example of part of a specification text treated by Ar2B:

الدرس معرف باسم ورقم ومدة زمنية. يمكن لهذا الدرس أن يدرس خلال سنة في عدة دورات. الدورة مميزة برقم وتاريخ البداية وثمن وعدد أيام الدورة. في غالب الأحيان الدورة مؤمنة من طرف عدَة منشطين. الدورة موضوعة تحت مسؤولية المنشط.

للمنشط رقم تسلسلي فريد ولقب ومرتب وعنوان شخصي. فأما عن المنشط، فيستطيع التدخل في عدة دورات في السنة على الأكثر، فلذلك نرغب في تسجيل عدد الساعات التي يدرسها كل منشط. وأما عن الدورة، فإنها متبوعة بعدد من المشاركين الذين يتميز كل منهم باسم وعنوان ورقم هاتف، كما يمكن للمشارك أن يكون موظفاً أو شخصاً. يعتبر المنشط إما منشطاً رئيسياً أو ثانوياً. يجب أن يكون سن المنشط أكبر من أو يساوي 18 سنة.

The rest of this paper is organized as follows: Section 2 deals with related work. We present our platform approach in Section 3. We show the experiments that we conducted and the first results of the first version obtained in Section 4. Finally, a conclusion and some perspectives for this work are given in Section 5.

2. Literature Review

Several research works have been aimed at automating the development of semi-formal models based on requirements' specifications in French or English language. The proposed approaches most often use NLP techniques. We can cite the works of [3]-[6] and the list is not exhaustive.

[7] proposed NL2Alloy combining a succession of tools allowing passing from constraints written in natural language to SBVR (NL2SBVR) rules, then towards UML/OCL (SBVR2OCL). They use automatic language processing methods and semantic technologies to generate UML models from natural language requirements. Manual interactions with the designer are then inevitable leading to semi-automatic approaches. On the other hand, several works have focused on the transition from UML to formal languages, such as B [8]; the Z language [9], which uses conceptual graphs as a pivotal model; or the language Maude [10], [1].

[2] used an ontology as a pivotal model; the formal language VDM and VDM ++ [11], …etc. The application of Artificial Intelligence techniques to requirements engineering [12] suggests software to be developed faster and better [13].

For all these research works, the results are satisfactory and their f-measure exceeds 90%. However, few studies are related to the state-of-the-art. These studies proposed only semi-formalizing Arabic user requirements and generating UML diagrams from them. They used algorithms for generating use case [14], sequence diagrams [15] and activity diagrams [16].

[14] and [15] generated diagrams from user requirements written in Arabic language, in which a set of heuristic rules were proposed.

[16] used a semi-automated algorithm for generating activity diagrams using MADA+TOKAN NLP tool, in which the elements of the activity diagrams have been extracted.

3. THE AR2B PLATFORM

Modeling of natural language became an issue of particular importance. It encouraged researchers to develop a variety of linguistic models that could solve practical problems. Linguistic models deal with statements as they are used to express meanings. They involve a body of meanings and a vocabulary to express meanings, as well as a mechanism to construct statements that can define new meanings based on the initial ones. The conceptual model represents 'concepts' (entities) and relationships between them. It plays an important role in the overall system development life-cycle. It is clear that if the conceptual model is not fully developed, the execution of fundamental system properties may not be implemented properly, giving way to future problems or system shortfalls, such as lack of user input, incomplete or unclear requirements and changing requirements. The concepts of the conceptual model can be mapped into physical design or implementation constructs using either manual or automated code generation approaches. To remove ambiguity and improve precision, to verify that the requirements have been met, to reason about the requirements/designs, to test for consistency, to explore consequences, to check automatically the properties; …etc., we need formalization. The formal model is based on rigorous methods and formats. Moving from the conceptual model to the formal model seems interesting.

Event-B is a formal model; it’s an extension of the B-method (J-R. Abrial). It is devoted to system engineering (both hardware and software) and to specifying and reasoning about complex systems: concurrent and reactive systems. Event-B models are organized in terms of the two basic constructs: contexts and machines. Contexts specify the static part of a model, whereas machines specify the dynamic part. The role of the contexts is to isolate the parameters of a formal model and their properties, which are assumed to hold for all instances. A machine encapsulates a transition system with the state being specified by a set of variables and transitions modelled by a set of guarded events. Event-B allows models to be developed gradually *via* mechanisms, such as context extension and machine refinement. These techniques enable users to develop target systems from their abstract specifications and subsequently introduce more implementation details. More importantly, properties that are proved at the abstract level are maintained through refinement and hence are guaranteed to be satisfied also by later refinement. As a result, correctness proofs of systems are broken down and distributed amongst different levels of abstraction, which are easier to manage [17]. Event-B comes with a new modelling framework called Rodin (like Atelier B tool for the classical B). The Rodin platform is an eclipse-based open and extensible tool for B model specification and verification. It integrates various plug-ins: B Model editors, proof-obligation, generators, provers, model-checkers, UML transformers, …etc.

Our platform allows conducting linguistic pretreatment, modeling and formal validation activities for text in Arabic language. We note, through the state-of-the-art, that a direct transition from informal specifications to formal specifications is not possible [18]. The common solution would be the transition to a pivotal intermediate representation that would reduce the gap between the two types of specifications [2].

To conceive Ar2B, two questions arose:

1. How is the problem of linguistic pretreatment to be solved with all the difficulties of treatment that the Arabic language knows?
2. How are the specifications written in natural language to be formalized?

The solution that we adopted in response to these questions is to conceive three models: linguistic, conceptual and semi-formal models in order to finally lead to formalization. It can be summarized as follows:

1. It is necessary to treat the text by various linguistic analyses by integrating them into a platform in order to annotate them and to represent them by an intermediate model that can serve as a pivot for conceptualization: **Linguistic model** represented by XML.
2. It is imperative to use an intermediate representation to move from this linguistic model to a semi-formal specification: **Conceptual model** represented by semantic networks.

3. It is necessary to transform this conceptual model by means of conceptualization rules and to represent it by a rigorous representation model to make it relevant at the level of formal specification. It is represented by UML class diagram (semi-formal) then by event-B: **Formal model.**

As shown in Figure 2, the text constitutes the basis of modeling. From it, it is possible to extract a linguistic model that will contain the elements expressed in the text. The conceptual model corresponds to a modeling process whose automation is realized only by a linguistic model. The choices of modeling are important at this level. They strongly depend on the granularity of the desired description and the objectives of the modeling. The semi-formal model is based on the exploitation of the representation and formalization potential of the UML language. The lack of formal semantics from which UML suffers can lead to serious modeling problems [1], generating inconsistencies in the models developed. In addition, its simplicity has as a price, which is lack of precision. This led us to make a transition to the formal model in event-B. A formalization of the conceptual modeling is thus generated.

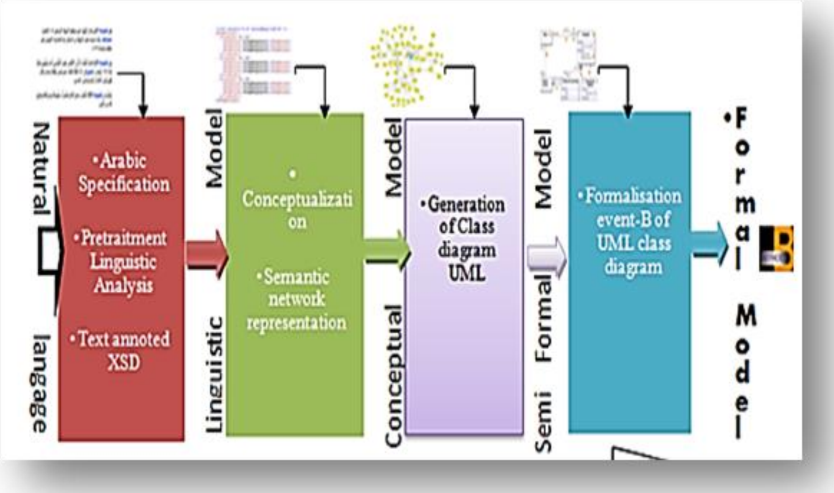

Natural langage
•Arabic Specification
•Pretraitment Linguistic Analysis
•Text annoted XSD
Linguistic Model
• Conceptualization
•Semantic network representation
Conceptual Model
•Generation of Class diagram UML
Semi Formal Model
•Formalisation event-B of UML class diagram
•Formal Model

Figure 2. Models of the platform Ar2B.

3.1 The Linguistic Model

To establish the linguistic model and proceed with the pretreatment (preprocessing) of the text of specification, we have designed the tool Alkhalil+ [19]. It does a considerable amount of work on MSA (Modern Standard Arabic), segmentation of the text in sentences, tokenization, morphological analysis, lemmatization, part-of-speech tagging (POS tag), disambiguation and diacritization. After the segmentation of the text into sentences, we use Alkhalil Morpho Sys (Alkhalil Morpho Sys, Version 1.3, http://sourceforge.net/projects/alkhalil/), a morphological online analyzer for Standard Arabic text. It is based in one part on the modeling of a large set of Arabic morphological rules and on the other part on the integration of linguistic resources that are useful to the other analysis. Next, we implemented a set of grammar rules as probabilistic contextual free grammar (PCFG) with a set of 87 ATNs (Augmented Transition Networks) applied to the pre-labelled text. Once the associated grammatical label for each word, a second disambiguation, is made, we apply a method based on decision theory to filter all successful applicant patterns and determine the final POS tag and diacritics of this word. The experiments have carried a disambiguation rate that is above 92.33%. The output is an XML file.

The XML schema consists of a set of sentences. Each word is described by its position in the sentence, its name, its lemma, its tag, a class of the verb, its multiplicity and its type.

The utilized tag set comprises the collapsed tags available in the Arabic TreeBank distribution {CC, CD, CONJ+NEG PART,DT, FW, IN, JJ, NN, NNP, NNPS, NNS, NO FUNC, NUMERIC,COMMA, PRP, PRP$, PUNC, RB, UH, VBD,VBN, VBP, WP, WRB} and we use 18 morpho-syntactic tags of Alkhalil {جا,نعا ,عا,نس,صن ,صه ,صر,صم,أص,وش ,فض,إض,نك ,زمك,آ,مفا,مف ,فا} to complete the description of verbs and nouns.

We have established a typology of verbs to determine the meaning of the sentence. We noted that the meaning of the verb gives us an indication of the semantics of the relationship in the future conceptual model. This typology is specific to the design of the IS. It is called class of verb.

Class of verb of state: includes the verbs that describe a state; these verbs express a hyponymy. Example: تنقسم, تتجزأ, تتفرع

Class of possession verb: introduces the concept of structural description. Example: يملك،يحوز،يتميز

Class of constraint verb: includes the verbs that are likely to define the constraints on attributes. Example: تفوق, تتجاوز, تنحصر, تتعدى

Class of action verb: includes the verbs that define an action; this action can be interpreted by a relationship. Example: يرسل، يكتب، يشارك، يزور

Class of composition verb: includes the verbs expressing the meronymy. Example: تحتوي، تتركب، تتشكل

Class of verb NULL (or sentiment): includes the verbs that signify no important action in the sentence. Example:.........يستطيع، يمكن، يقدر, يجب

About multiplicity, it is specific to the noun; it depends on its morphological nature; multiplicity = ‘2’ (dual), multiplicity = ‘*’ (plural). If the number is not specified in the text, it is equal to 1 by default.

Considering the type, it is an attribute to particles. It takes the value: NULL or NOT_NULL depending on whether or not these particles have a semantic meaning and a role in the design of our future IS. For example, particles ل' /lem' and في' fi' can in some sentences play an important role in the design; for example, in the sentence: للمعلم اسم و لقب و رقم شخصي / the teacher is characterized by a name, a surname and a personal number /, the particle 'ل' plays the role of a verb of possession. On the other hand, in the sentence:يدرس المعلم للطلاب مادتين إثنتين , / The teacher teaches two subjects to the students / particle 'ل' plays no role; its type will be set to NULL.

3.2 The Conceptual Model

The proposed approach starts by extracting terms and compound terms from the annotated XML file. The second step is the design of the chunker; we have defined a list of categories of chunks that were necessary for the classification of the sentences in order to extract the meaning; only the chunks pertinent to the structural description of the future IS are selected and we attribute to them different roles. Then, we proceed with the classification of these sentences according to sentence patterns that we have already determined. A semantic network represents the extracted information. A set of design patterns are applied, hence generating the corresponding UML class diagram.

3.2.1 Extraction of Simple and Compound Terms

To extract simple terms, our approach is based on weight calculation. We have not assigned a weight to every word in the text; it is only calculated for these whose tag is (NN, DT NN, NNS, NNP, NNPS), because there could be classes or potential attributes in the design of the future IS. This weight can be critical for the recognition of the nature of this concept. We calculated the frequency of the term using the lemma; the weight is calculated with the formula tf-idf. A list of term candidates is so extracted.

To extract word pairs, we have used a hybrid method. We have defined linguistic patterns to determine couples of candidates and then we have filtered them by using a statistical method based on mutual information (MI) in order to keep the couples of pertinent words. A third filtering is performed during the validation of the semantic network. If the chosen pair consists of two terms that have been identified as an entity, the couple is then rejected as a compound word and another treatment will be assigned to it.

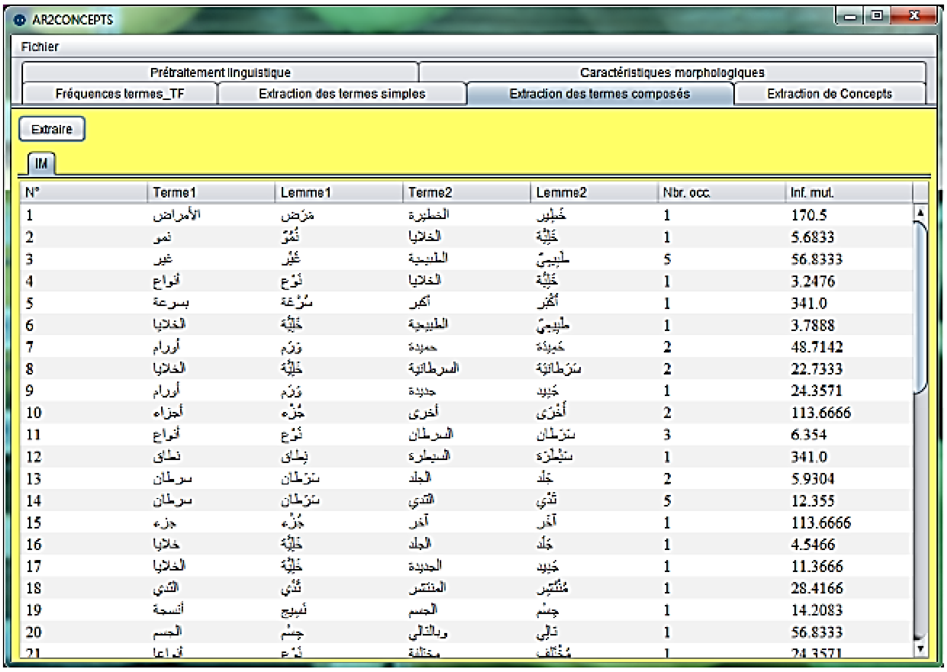
The linguistic pattern. For syntactic patterns, we have adopted the research work of [20]. We focus here on collocations consisting of two lexical units and respecting the following schemes: NN + JJ; (DT+NN) + (DT+JJ); NN + (DT+NN); NN + (DT+JJ); NN + NNP; NNP + NN……. (NN: indefinite noun; JJ: adjective; DT: definite noun).

Example: المدرسة الابتدائية, جامعة العلوم, الطالب الجامعي, طالب متفوق

The filtering calculation of mutual information (MI). For each pair of words learned previously, we calculate the MI. The MI is used to determine whether two words are closely related or not. Given two words designated by the variables x and y, the MI is calculated using the following formula (1):

$$MI(x,y) = \log_2 \frac{p(x,y)}{p(x)p(y)} \qquad (1)$$

where, p (x) and p (y) are the probabilities of observation of respectively words x and y; and p (x, y) is the probability to observe them together. Once the IM is calculated for each couple, we experimentally set a threshold for preferred pairs with strong cohesion. Figure 3 shows the compound terms extracted by Ar2B.


AR2CONCEPTS
Fichier
Prétraitement linguistique
Caractéristiques morphologiques
Fréquences termes_TF
Extraction des termes simples
Extraction des termes composés
Extraction de Concepts
Extraire
IM

N°	Terme1	Lemme1	Terme2	Lemme2	Nbr. occ.	Inf. mut.
1	الأمراض	مَرَض	الخطيرة	خَطِير	1	170.5
2	نمو	نُمُوّ	الخلايا	خَلِيَّة	1	5.6833
3	غير	غَيْر	الطبيعية	طَبِيعِيّ	5	56.8333
4	أنواع	نَوْع	الخلايا	خَلِيَّة	1	3.2476
5	بسرعة	سُرْعَة	أكبر	أَكْبَر	1	341.0
6	الخلايا	خَلِيَّة	الطبيعية	طَبِيعِيّ	1	3.7888
7	أورام	وَرَم	حميدة	حَمِيدَة	2	48.7142
8	الخلايا	خَلِيَّة	السرطانية	سَرَطَانِيَّة	2	22.7333
9	أورام	وَرَم	جديدة	جَدِيد	1	24.3571
10	أجزاء	جُزْء	أخرى	أُخْرَى	2	113.6666
11	أنواع	نَوْع	السرطان	سَرَطَان	3	6.354
12	نطاق	نِطَاق	السيطرة	سَيْطَرَة	1	341.0
13	سرطان	سَرَطَان	الجلد	جِلْد	2	5.9304
14	سرطان	سَرَطَان	الثدي	ثَدْي	5	12.355
15	جزء	جُزْء	آخر	آخَر	1	113.6666
16	خلايا	خَلِيَّة	الجلد	جِلْد	1	4.5466
17	الخلايا	خَلِيَّة	الجديدة	جَدِيد	1	11.3666
18	الثدي	ثَدْي	المنتشر	مُنْتَشِر	1	28.4166
19	أنسجة	نَسِيج	الجسم	جِسْم	1	14.2083
20	الجسم	جِسْم	وبالتالي	تَالِي	1	56.8333
21	أنواعا	نَوْع	مختلفة	مُخْتَلِف	1	24.3571

Figure 3. Extraction of compound terms by Ar2B.

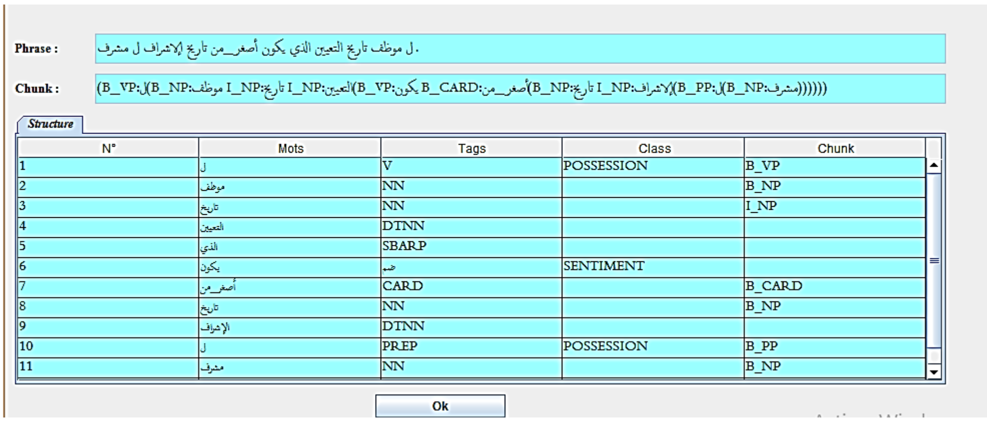
3.2.2 Base Phrase Chunking

Free-order language complicates the grammar construction. The basic order of Arabic words in a sentence is Verb–Subject-Object (VSO). However, other orders are possible: SVO and VOS. The idea of chunking was the solution to i) reduce the number of rules and thus improve the performance of the relationship extraction system and ii) eliminate the treatment of temporal and behavioral syntagm types in the structural description of the class diagram. Its use in the semantic analysis has made our system much more efficient in the classification of the sentences according to the established patterns of sentences. For this task, we use a setup similar to that of [21], with the BIO annotation representation: "Beginning”, "Inside" and “Outside" the chunk. Ten types of chunked phrases are recognized: {VP, NP, ADJP, PP, ADVP, CONJP, INTJP, PREDP, PRTP and SBARP}. We have added two other types: CD that indicates the cardinality necessary to determine in the design of the future IS and CARD to define static constraint. Chunking is introduced by the presence of words as indicated in Table 1.

Table 1. The chunk added by Ar2B.

Type of Chunk	Begin of the chunk	
CD	جميع، كل، وحيد، فقط........	number
CARD	أكبر من، يساوي، ما بين، أقل ، أصغر من، أصغر من أو يساوي، أقل من، على الأقل، في أحد، واحد من.........	constraint verb type

An example of chunking is given in Figure 4.



Phrase : ل موظف تاريخ التعيين الذي يكون أصغر_من تاريخ الإشراف ل مشرف.

Chunk : (B_VP:ل(B_NP:موظف I_NP:تاريخ I_NP:التعيين(B_VP:يكون B_CARD:أصغر_من(B_NP:تاريخ I_NP:الإشراف(B_PP:ل(B_NP:مشرف))))))

Structure

N°	Mots	Tags	Class	Chunk
1	ل	V	POSSESSION	B_VP
2	موظف	NN		B_NP
3	تاريخ	NN		I_NP
4	التعيين	DTNN		
5	الذي	SBARP		
6	يكون	ضم	SENTIMENT	
7	أصغر_من	CARD		B_CARD
8	تاريخ	NN		B_NP
9	الإشراف	DTNN		
10	ل	PREP	POSSESSION	B_PP
11	مشرف	NN		B_NP

Figure 4. Chunking by Ar2B.

3.2.3 Semantic Analysis

For semantic analysis, our approach is hybrid. It is based on Fillmore's case theory, combined with the verb-based and pattern-based approach. As shown in Figure 5, we begin with the assignment of roles to the various components in the chunks of the sentence. Then, we proceed with classifying these sentences into patterns. A semantic network is generated as output in order to represent the whole extracted information.

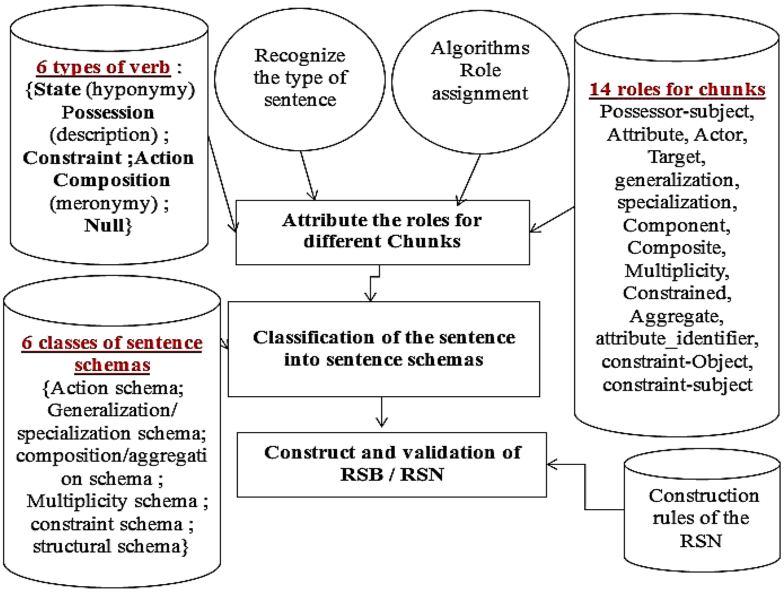

6 types of verb : {State (hyponymy) Possession (description) ; Constraint ;Action Composition (meronymy) ; Null}
Recognize the type of sentence
Algorithms Role assignment
14 roles for chunks
Possessor-subject, Attribute, Actor, Target, generalization, specialization, Component, Composite, Multiplicity, Constrained, Aggregate, attribute_identifier, constraint-Object, constraint-subject
Attribute the roles for different Chunks
6 classes of sentence schemas
{Action schema; Generalization/ specialization schema; composition/aggregation schema ; Multiplicity schema ; constraint schema ; structural schema}
Classification of the sentence into sentence schemas
Construct and validation of RSB / RSN
Construction rules of the RSN

Figure 5. Semantic analyzer design approach.

Step 1/ Recognition the type of sentence and attribution of roles: Fillmore accords that the verb is the central component of the sentence. He schematizes the sentence as distinction between modality (M) and proposal (P). The modality contains information about negation, time, mode and appearance. V is the verb. Each Ci is the name of a case (role) that will represent a name related to the verb by semantical case Ci.

$$S_V = \mathrm{M} + VP + C1 + C2 + \cdots . + Cn \qquad (2)$$

We have extended this definition to take into account the nominal sentence, knowing that it is non-existent in other languages, such as English and French. The formulae will then be written as:

$$S_N = \mathrm{M} + \mathrm{Pivot} + C1 + C2 + \cdots . + Cn \qquad (3)$$

$$Sv = M + P\,,\;\; PV + C1 + C2 + \cdots . + Cn \qquad (4)$$

For our design, Ci indicates the semantic case that binds a *chunk* with *verbal chunk (VP)* or with pivot.

We have identified 14 roles (*Ci)* (Figure 2). We were inspired to define our roles to those defined in Fillmore's causal theory. We adapted them to the Arabic language and to the purpose of our conception.

For S_V processing, it is recognized by the presence of a verbal chunk (VP) and not a verb. We were inspired by the research of [22] for detecting it. Our VP is recognized by the identification of a verb or verbal noun/ ((مصدر (صم, صر, صه, صأ)), active participle / ((اسم فاعل (فا)) or passive participle /((اسم مفعول مف)) or the particle 'ل' (the particle of possession/لام الامتلاك). We look for the verb class and the identification of the semantic relations that link the different chunks to the verbal chunk, in order to assign coherent roles to the different words and chunks composing this sentence.

We have defined for each class of verb an algorithm for assigning roles, except for the class of verb null considered as a stop word.

Algorithm *Class of action verb*

```
Begin
For i=1 to n do //n: number of sentences in text//
 Find_chunk_verbal (”VP”,i,exist,class) ;//research chunk VP (verbal chunk) //
  If exist then
   If class=”action” then
    If Find_chunk (i,”NP”) then  //research chunk NP (nominal chunk)//
    Role_chunk_NP ← “actor”; //For each term in this chunk NP attribute role “Actor”//
      While Find_chunk (i,”NP”) do //research another chunk NP//
        Role_chunk_NP ←”target” //For each term in this chunk NP, attribute role “target”//
        End;   End;
     While Find_chunk (i,”PP”) do     // PP: prepositional chunk//
        Role_chunk_PP ←”target”; end;
     While Find_chunk (i,”CD”) do  // CD: Cardinal Chunk//
              Role_chunk_PP ←”multiplicity”; end;
     end;end;end;
```

In Figure 6, an example is given of a sentence in which *Class of verb is State*. The algorithm assigns chunk roles.

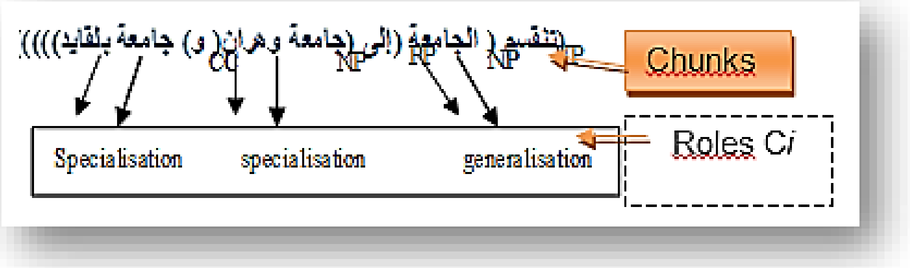


Figure 6. Example 'State' class verb=' ينقسم '.

Remark: جامعة وهران و جامعة بلقايد have been determined as compound words by our tool.

For S_N processing, usually the nominal sentence describes either an association relationship or an inheritance relationship. The relationship is deduced by a prepositional chunk (PP), nominal chunk (NP) or adjectival chunk (ADJP). It may even be implicit, in which case we look for a pivot element (the subject of the action) in the nominal chunk, identified by its Pos tag and by its position in this chunk. We have established algorithms that assign roles for each chunk of the sentence.

Step 2/ Classification of the sentence into sentence schemas: The sentence patterns allow us to stereotype the sentences; we have classified them according to six schemas (Figure 5). This classification determines the first interpretation of the specification text.

For example, all sentences resulting from one of this combination are classified as Structural, Action or

Generalization/ specialization schema.

Structural schema
Verbal Chunk in possession class + Role possessor subject+ Role Attribute+ constraint Role.

Action Schema
Verbal Chunk in Action class + actor Role+ target Role +multiplicity Role

Generalization/ specialization schema
Verbal Chunk in State class + Generalization Role+ specialization Role +multiplicity Role

The sentence shown in Figure 7 is in the structural schema:

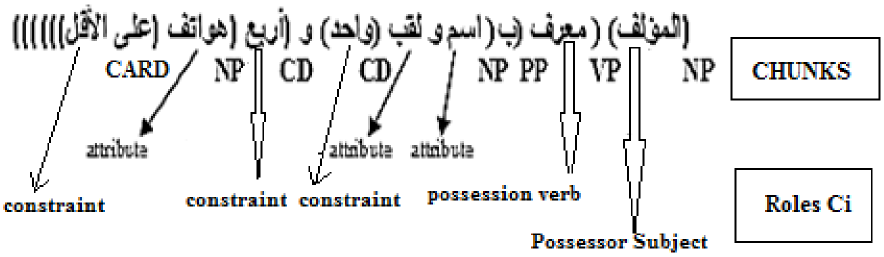


Figure 7. Example of classification in structural schema.

The constraint is recognized by the presence of a chunk CARD in the sentence. Once this chunk is detected, the processing of the constraint passes priority over the treatment of the different types of sentences.

<u>**Step 3/ Construction and validation of semantic network (RSB/RSN)**</u>: After classifying the sentences, we applied specific algorithms to extract the relevant information to represent them by a rigorous model to represent knowledge, which is the **S**emantic **N**etwork (RSN) in order to make it accessible.

The **r**aw **s**emantic **n**etwork (RSB) is the first network built; the RSN is the validated version of the RSB with pattern design. The RSN is characterized by a set of nodes and arcs. We have defined a total of 7 types of nodes and 10 types of arcs as shown in Figure 8, which allowed us to represent all relevant information in our corpus.

The nodes of the RSB: {entity, action, multiplicity, constraint, value, negation};
The nodes of RSN = RSB U {operation};
The arcs of the RSB = {poss, acti, mult, is-a, comp / arg, cti, not, val};
The arcs of RSN = RSB U {op, id}.

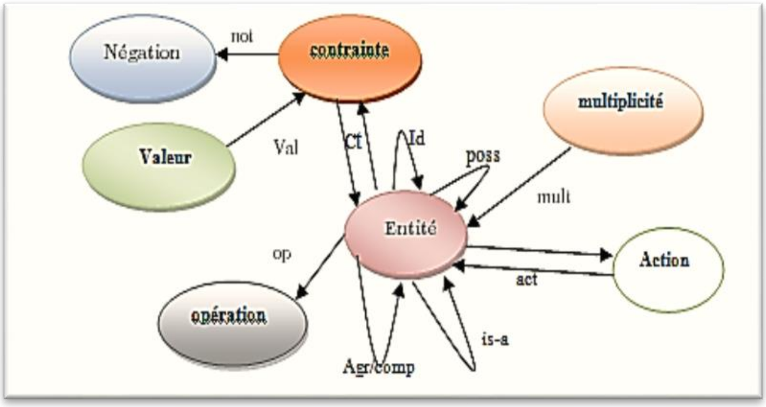


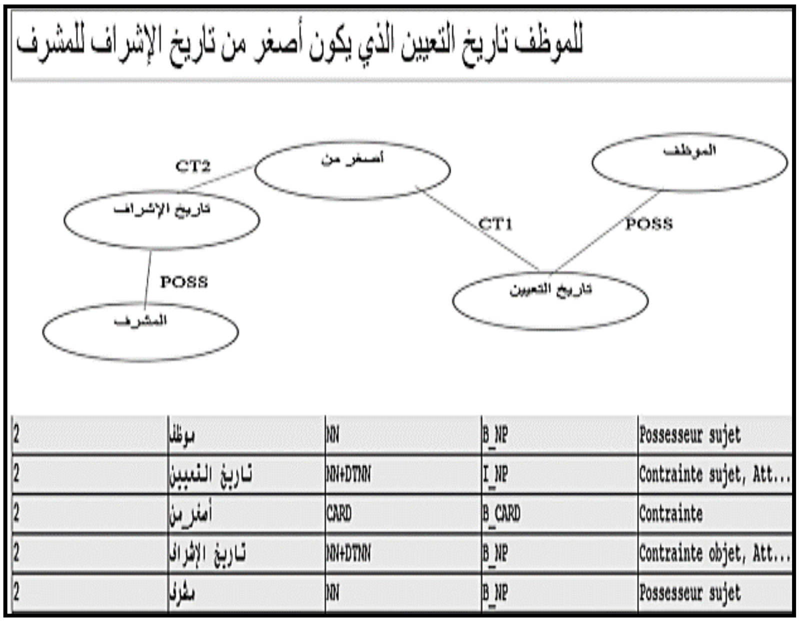
Figure 8. The meta-model of RSB / RSN.

For each sentence schema, we have applied specific algorithms for processing nodes and arcs.

An example of rule of structural schema:

- All terms having as Role “possessor_subject” and role “attribute” will all be transformed into entity nodes.
- They will be linked by an arc 'poss'. The source of the arc will be the term whose role is "possessor_subject".

In Figure 9, Ar2B treats a sentence. The second column contains and affichs the simple terms found: مشرف, موظف and compound terms: تاريخ التعيين, تاريخ الإشراف. The third column contains the Pos tag. In the fourth one, it presents the results of chunking. The roles assigned are displayed in the fifth column and the corresponding RSN is generate. We note that "poss" links were deducted automatically between تاريخ التعيين and موظف ; تاريخ الإشراف and مشرف. Although they were not specified as such in the sentence. Our approach allows for the detection of implicit arcs and nodes using Chunks role assignment algorithms and sentence classification algorithms.

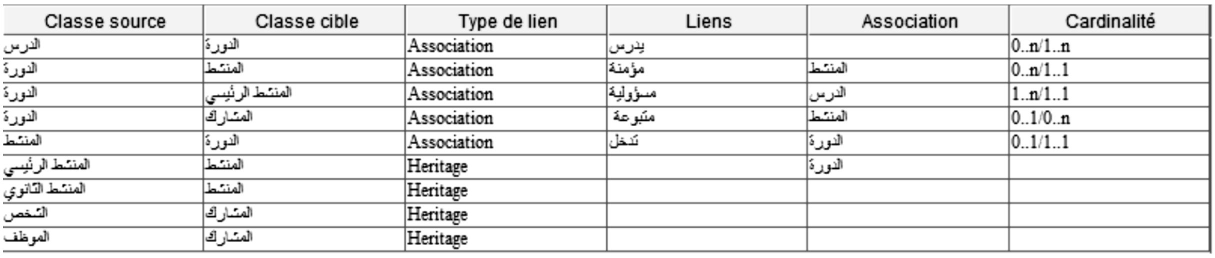


للموظف تاريخ التعيين الذي يكون أصغر من تاريخ الإشراف للمشرف

2	موظف	NN	B_NP	Possesseur sujet
2	تاريخ التعيين	NN+DTNN	I_NP	Contrainte sujet, Att...
2	أصغر_من	CARD	B_CARD	Contrainte
2	تاريخ الإشراف	NN+DTNN	B_NP	Contrainte objet, Att...
2	مشرف	NN	B_NP	Possesseur sujet

Figure 9. RSN of constraint schema sentence treated by Ar2B.

Once the network is generated, we proceed with its normalization for verification of the consistency of the network. To do that, we have established normalization rules inspired by design patterns of software engineering. These rules will help us remove some nodes and arcs and create others to validate coherence and compliance. Rule 5 is an example of normalization rules. Figure 10 shows the application of this rule to the specification text given at the top (المنشط الرئيسي والمنشط الثانوي).



Classe source	Classe cible	Type de lien	Liens	Association	Cardinalité
الدرس	الدورة	Association	يدرس		0..n/1..n
الدورة	المنشط	Association	مؤمنة	المنشط	0..n/1..1
الدورة	المنشط الرئيسي	Association	مسؤولية	الدرس	1..n/1..1
الدورة	المشارك	Association	متبوعة	المنشط	0..1/0..n
المنشط	الدورة	Association	تدخل	الدورة	0..1/1..1
المنشط الرئيسي	المنشط	Heritage		الدورة	
المنشط الثانوي	المنشط	Heritage			
الشخص	المشارك	Heritage			
الموظف	المشارك	Heritage			

Figure 10. Application of rule 5 by Ar2B.

Rule 5: Transformation in the entity node

Any entity node containing a compound word will be converted into 2 entity nodes connected by a link 'is_a' if one of the terms (or both) is a node entity.

Figure 11 represents the RSB of the specification given in Section 1.

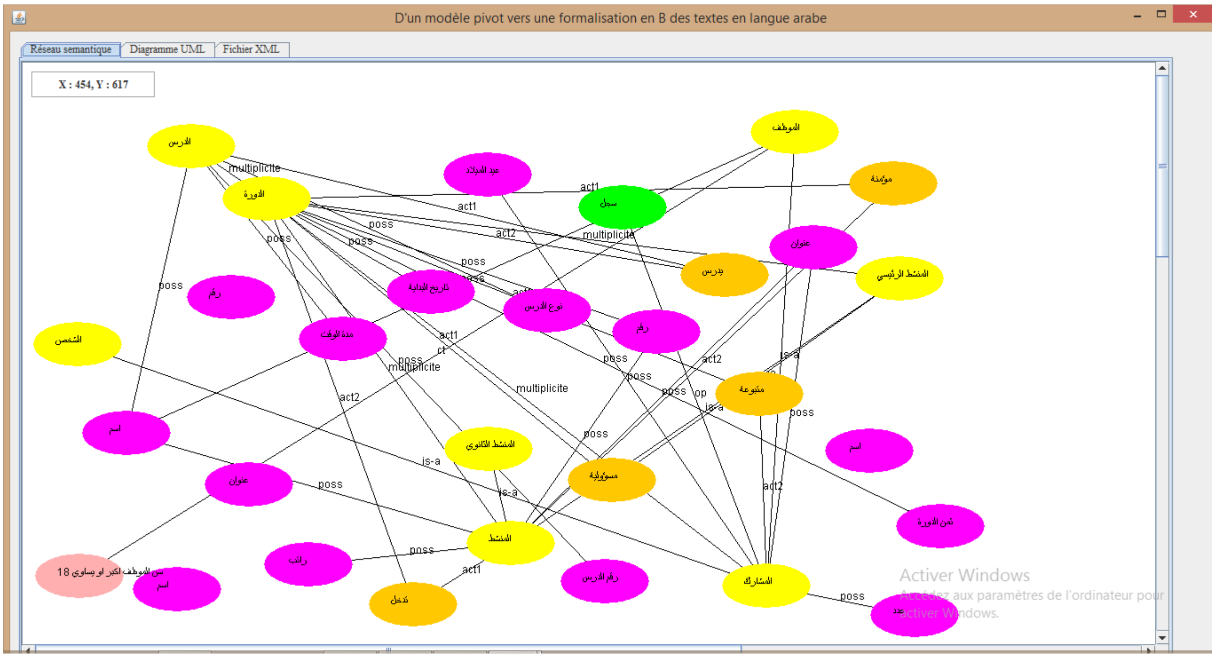


Figure 11. Generation of RSB by Ar2B.

3.3 The Formal Model

Ar2B generates the corresponding class diagram and then transforms it into an XML file for better interoperability. A second passage from UML to formal language Event-B is done. This formal method performs formal refinement in several steps until the final refinement contains enough details for an implementation.

Since we are the pioneers in the B-language formalization of the Arabic language, we were obliged to make an adaptation of the plug-ins for this language for the Rodin platform.
In the meantime, we have studied the specification of a class diagram under the RODIN platform and its implementation in EVENT-B. Machine and context will then be translated into XML (.bum: machine) and (.buc: context) files. To do this, we have deduced rules for switching from UML to EVENT-B.

3.3.1 Extraction of the Concepts of Class Diagram: Semi-formal Model

We have defined a rule database that we have applied to the standard semantic network (RSB) in order to extract the classes, the relations, the cardinalities, the operations and the static constraints of the class diagram. Once the model is generated, we apply to it a set of design patterns: creation design patterns, structure design patterns, in order to verify its conformity, consistency and completeness. Example of rules of transformation from RSN to class diagram

R1: Class concept identification

All the nodes of type "entity" sources of the arc "poss" will be transformed into classes.

R2: Identification of the relationship concept

- All nodes of type "action" will be transformed into relations. The role will be the name of the "action" node.
- All source/target "entity" nodes of an "act" arc will be classes linked by this relation.
- All the nodes of type "entity" linked by an arc "agr/comp" will become classes linked by a relation of composition / aggregation whose component is the target of the arc "agr /comp".
- All target "entity" nodes and "is-a" arc sources will be transformed into an inheritance relationship. The source nodes will be generic class and the target nodes the specialized class.

Figure 12 shows the class diagram generate by Ar2B of the specification given in Section 1.

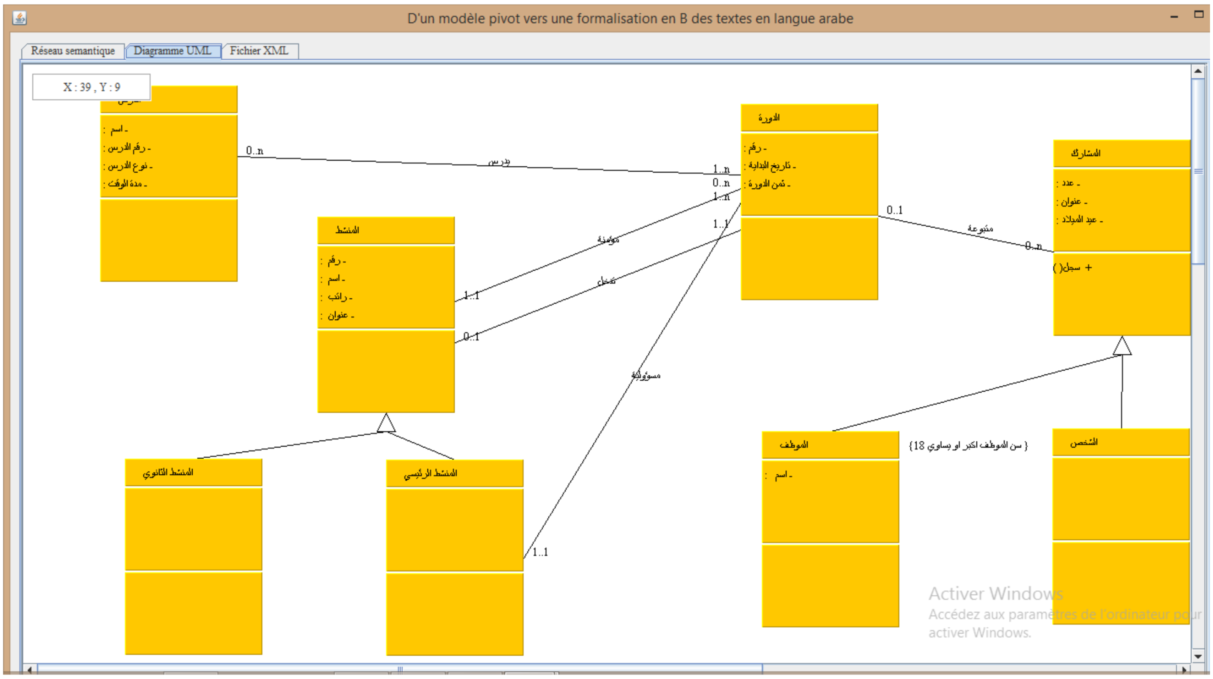
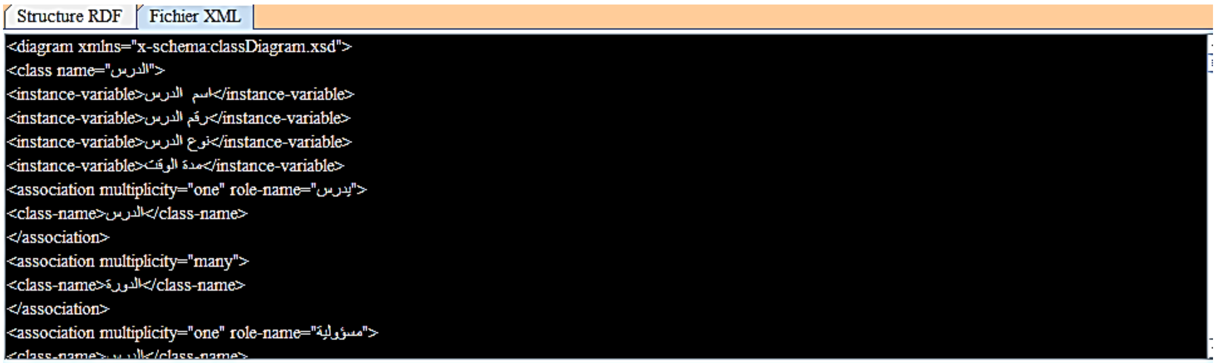


Figure 12. Class diagram generate by Ar2B.

3.3.2 Transformation of the UML Class Diagram to XML Schema

We built the XML schema from the UML model through the concept of Meta-data XML Interchange (XMI) specification, which defines a rigorous approach for generating an XML DTD from a meta-model definition expressed by UML to XML Schema. The transformation rules used in the mapping process are described as follows in [23]. Figure 13 shows the XML file generated by Ar2B of the specification given in Section 1.



```
<diagram xmlns="x-schema:classDiagram.xsd">
<class name="الدرس">
<instance-variable>اسم الدرس</instance-variable>
<instance-variable>رقم الدرس</instance-variable>
<instance-variable>نوع الدرس</instance-variable>
<instance-variable>مدة الوقت</instance-variable>
<association multiplicity="one" role-name="يدرس">
<class-name>الدرس</class-name>
</association>
<association multiplicity="many">
<class-name>الدورة</class-name>
</association>
<association multiplicity="one" role-name="مسؤولية">
```

Figure 13. XML file generated by Ar2B.

3.3.3 Extraction of Formal Specification with Event-B

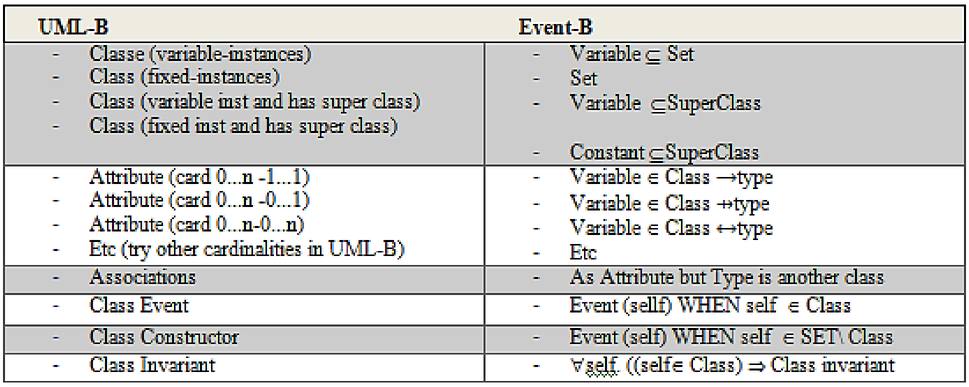
We try to apply the plug-in (XSLT Orange volt) Eclipse to translate our XML through the transformation rules proposed by XSLT and from research work [24] to automatically produce Event-B specifications (.bum / .buc) under the Rodin demonstrator.

Event-B is an extension of the B-method (J-R. Abrial). It uses set theory and logic, is relatively simple and has an extensive tool support. It comes with a new modelling framework called Rodin (like Atelier B tool for the classical B). The Rodin platform is an eclipse-based open and extensible tool for B-model specification and verification.

Unfortunately, forced by the transformation of our UML class diagram into a SOA pattern design diagram and the non-finalization of the transformation rules proposed by the authors, our choice then turned to the UML-B software. These specifications can also produce SQL or JAVA source code after a series of proven refinements proposed by the plug-in UML-B.

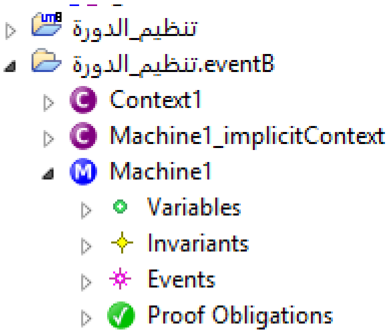
Table 2 summarizes the correspondence between a UML-B (context) specification and that of event-B. It is generated automatically.

Table 2. Correspondence UML-B/event-B.



UML-B	Event-B
- Classe (variable-instances) - Class (fixed-instances) - Class (variable inst and has super class) - Class (fixed inst and has super class)	- Variable ⊆ Set - Set - Variable ⊆SuperClass - Constant ⊆SuperClass
- Attribute (card 0...n -1...1) - Attribute (card 0...n -0...1) - Attribute (card 0...n-0...n) - Etc (try other cardinalities in UML-B)	- Variable ∈ Class →type - Variable ∈ Class ⇸type - Variable ∈ Class ↔type - Etc
- Associations	- As Attribute but Type is another class
- Class Event	- Event (sellf) WHEN self ∈ Class
- Class Constructor	- Event (self) WHEN self ∈ SET\ Class
- Class Invariant	- ∀self. ((self∈ Class) ⇒ Class invariant

Figure 14 shows the formal specification with event_B of the text in Section 1.



تنظيم_الدورة
تنظيم_الدورة.eventB
Context1
Machine1_implicitContext
Machine1
Variables
Invariants
Events
Proof Obligations

MACHINE
Machine1
SEES
Machine1_implicitContext
VARIABLES
منشط // class instances
مشارك // class instances
دورة // class instances
درس // class instances
شخص // class instances
موظف // class instances
رئيسي_منشط // class instances
ثانوي_منشط // class instances
يتدخل // attribute of منشط
المنشط_اسم // attribute of منشط
المنشط_رقم // attribute of منشط
المنشط_راتب // attribute of منشط
المنشط_عنوان // attribute of منشط
المشارك_اسم // attribute of مشارك
المشارك_رقم // attribute of مشارك
المشارك_عنوان // attribute of مشارك
الميلاد_عيد // attribute of مشارك
Annexe
مسؤولية // attribute of دورة
متبوعة // attribute of دورة
مؤمنة // attribute of دورة
الدورة_رقم // attribute of دورة
البداية_تاريخ // attribute of دورة
الدورة_ثمن // attribute of دورة
يدرس // attribute of درس
الدرس_اسم // attribute of درس
الدرس_رقم // attribute of درس
الدرس_نوع // attribute of درس
الوقت_مدة // attribute of درس
INVARIANTS
منشط _type : منشط ∈ ℙ (منشط _SET)
مشارك _type : مشارك ∈ ℙ (مشارك _SET)
دورة _type : دورة ∈ ℙ (دورة _SET)
درس _type : درس ∈ ℙ ((دروس)
شخص _type : شخص ∈ ℙ ((مشارك)
موظف _type : موظف ∈ ℙ ((مشارك)
رئيسي_منشط _type : رئيسي_منشط ∈ ℙ ((منشط)
ثانوي_منشط _type : ثانوي_منشط ∈ ℙ ((منشط)
يتدخل _type : يتدخل ∋ منشط ↔ دورة
المنشط_اسم _type : المنشط_اسم ∋ منشط → منشطين
المنشط_رقم _type : المنشط_رقم ∋ منشط → منشطين
المنشط_راتب _type : المنشط_راتب ∋ منشط → ℕ
المنشط_عنوان _type : المنشط_عنوان ∋ منشط → منشطين
المشارك_اسم _type : المشارك_اسم ∋ مشارك → مشاركين
المشارك_رقم _type : المشارك_رقم ∋ مشارك → ℕ
المشارك_عنوان _type : المشارك_عنوان ∋ مشارك → مشاركين
الميلاد_عيد _type : الميلاد_عيد ∋ مشارك → ℕ
مسؤولية _type : مسؤولية ∋ دورة ↔ رئيسي_منشط
متبوعة _type : متبوعة ∋ دورة ↔ مشارك

```
مؤمنة _type : مؤمنة ∋ دورة ↔ منشط
الدورة_رقم _type : الدورة_رقم ∋ دورة → دورات
البداية_تاريخ _type : البداية_تاريخ ∋ دورة → ℕ
الدورة_ثمن _type : الدورة_ثمن ∋ دورة → ℕ
يدرس _type : يدرس ∋ درس ↔ دورة
الدرس_اسم _type : الدرس_اسم ∋ درس → دروس
الدرس_رقم _type : الدرس_رقم ∋ درس → دروس
الدرس_نوع _type : الدرس_نوع ∋ درس → دروس
الوقت_مدة _type : الوقت_مدة ∋ درس → دروس
EVENTS
INITIALIZATION ≙
STATUS
ordinary
BEGIN
منشط _init : ∅ ≔ منشط
مشارك _init : ∅ ≔ مشارك
دورة _init : ∅ ≔ دورة
درس _init : ∅ ≔ درس
شخص _init : ∅ ≔ شخص
موظف _init : ∅ ≔ موظف
رئيسي_منشط _init : ∅ ≔ رئيسي_منشط
ثانوي_منشط _init : ∅ ≔ ثانوي_منشط
يتدخل _init : ∅ ≔ يتدخل
المنشط_اسم _init : ∅ ≔ المنشط_اسم
المنشط_رقم _init : ∅ ≔ المنشط_رقم
Annexe
المنشط_راتب _init : ∅ ≔ المنشط_راتب
المنشط_عنوان _init : ∅ ≔ المنشط_عنوان
المشارك_اسم _init : ∅ ≔ المشارك_اسم
المشارك_رقم _init : ∅ ≔ المشارك_رقم
المشارك_عنوان _init : ∅ ≔ المشارك_عنوان
الميلاد_عيد _init : ∅ ≔ الميلاد_عيد
مسؤولية _init : ∅ ≔ مسؤولية
متبوعة _init : ∅ ≔ متبوعة
مؤمنة _init : ∅ ≔ مؤمنة
الدورة_رقم _init : ∅ ≔ الدورة_رقم
البداية_تاريخ _init : ∅ ≔ البداية_تاريخ
الدورة_ثمن _init : ∅ ≔ الدورة_ثمن
يدرس _init : ∅ ≔ يدرس
الدرس_اسم _init : ∅ ≔ الدرس_اسم
الدرس_رقم _init : ∅ ≔ الدرس_رقم
الدرس_نوع _init : ∅ ≔ الدرس_نوع
الوقت_مدة _init : ∅ ≔ الوقت_مدة
END
END
```

Figure 14. Specification with event_B of class diagram obtained.

4. Experiments and Results

In order to evaluate Ar2B, experiments were performed on a corpus consisting of a collection of texts in Arabic by maintaining existing potential diacritics. The corpus contains about 51404 words, including 81 Arabic text, 899 paragraphs, 3871 sentences and 29188 words. The sentence can contain upto 25 words. We have taken texts from the practical exercises of the 'software engineering' course taught to 3rd year students in computer science of our university. We also translated specification texts that we took from other universities, websites and books. A list of text is available at: https://sites.google.com/site/kheirazinebbousmeha/corpus.

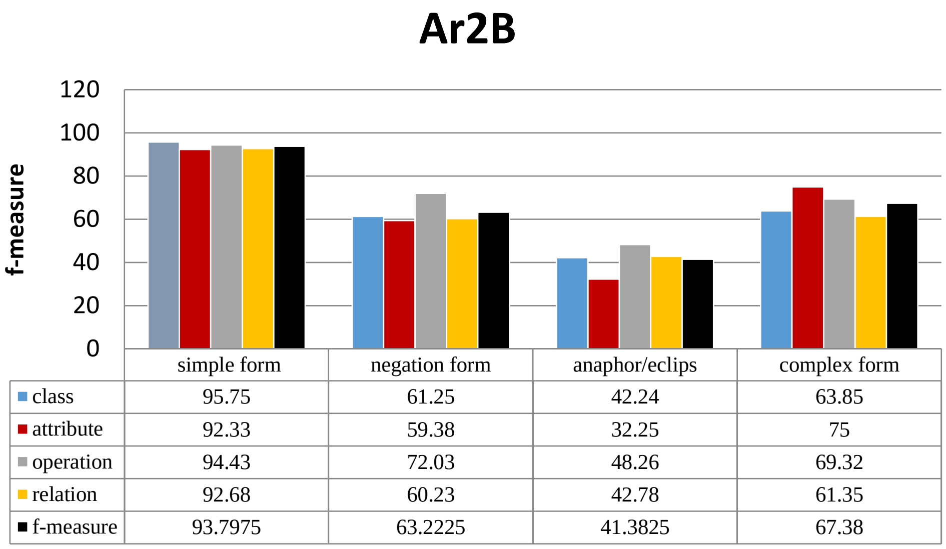
We have put our first results on the graph in Figure 15. We have, for a text comprising only simple sentences, an f-measure greater than 93 % in the generation of the class diagram. The f-measures of each concept were in the order of 95 % for the extraction of class and more than 92 % for the extraction of attribute, operation and relation. These f-measures would decrease as the sentence became more complicated, containing negative forms (f-measure=63.3825%), anaphoras and ellipses (f-measure=41.3825 %) or a complex formulation.

The greatest values were observed in the extraction of classes and attributes, because we used, in addition to the linguistic rules, statistical measures. The lowest rate was that of the generalization and specialization relation. This type of relationship has a schema similar to the action-type schema. It can be referred to in nominal and in verbal sentences. Sometimes, the verb type is ambiguous. This problem has been circumvented in other languages (English and recently in French) by the use of verbnet lexicon, where it is possible to use a syntactic construct to match an argument of a verb to semantic roles [25].

We use the confusion matrix in order to analyze the error rate of each concept. According to the values reported in this table, the overall error rate is:

$$E = 1 - \left(\sum_{i=1}^{m} n_{ii} / \sum_{i=1}^{m} n_{ij}\right) = 23.38\%. \qquad (5)$$

This matrix reveals the distribution of the error for each concept of the diagram on each rate found. We note that for the Attribute-identifier concept for example, 2.5% is erroneously classified as classes and 9.1% are classified, also by mistake, as attributes. This can be explained by a bad formulation of the rules concerning the extraction of the identifier attribute. This measurement allowed us to locate the error and review certain points of the design.



	simple form	negation form	anaphor/eclips	complex form
■ class	95.75	61.25	42.24	63.85
■ attribute	92.33	59.38	32.25	75
■ operation	94.43	72.03	48.26	69.32
■ relation	92.68	60.23	42.78	61.35
■ f-measure	93.7975	63.2225	41.3825	67.38

Figure 15. The f-measures obtained by Ar2B for each type of text.

5. Conclusions

We presented a platform for ANLP devoted to a text language processing treating the functional specifications and more specifically in the general functional specifications (GFSs).

Ar2B encompasses a set of coherent modules and an automatic event-B formalization of conceptual modeling based on hybrid approaches and reliable tools. It processes a large number of specifications in a shorter time and in a less subjective way than an expert.

The realization of the conceptual phase by "chunking" simplifies sentences. The idea of classifying sentences using Fillmore's case theory allowed us to disambiguate the different interpretations that a sentence may contain. The proposed hybrid approach based on linguistic rules and statistical methods allowed us to generate the relevant concepts of the future semi-formal model.

As for semi-formalization, the proposed approach makes it possible to take into account, through a set of heuristics and then design patterns, the automatic passage of the text represented by the standardized semantic network into a UML class diagram.

We used semantic networks for unambiguous semantic interpretation of specifications. The use of ontologies significantly improves the quality of the specified requirements. Their use as an intermediate representation in a process of automatic formalization of natural language specifications has been explored only recently [2]. The lack or absence of domain ontology devoted to the Arabic language has led us to use semantic networks that respond well to the specificities of our field of application.

Regarding the formalization in event-B and since there is no work on this formalization in Arabic, many difficulties were encountered among which we quote the installation of the platform Rodin and UML-B software as well as the adaptation of the plug-ins for the Arabic language.

We chose to expand our platform environment of the open source platform Rodin Version 3.2.0-ecacdcb; an IDE based on Eclipse for event-B provides an effective support for refinement and mathematical proof. The platform contributes through Eclipse and can be extended with plug-ins.

The results obtained are promising and have reached f-measures of around 70% for all types of sentences and up to 93% for the treatment of simple sentences. For the treatment of anaphoras and ellipses, we plan to integrate the research of our ANLP-RG and take into account the treatment of synonymy by using Ontology AWN as well as to complete the UML model by the OCL constraint expression language for more processing constraints.

ACKNOWLEDGEMENTS

The authors would like to thank the ANLP-RG research team as well as the master students from the Computer Science Department of the University Oran1 for their efforts in designing the Ar2B platform. They would also like to thank Mr. MEZIANE Farid, Professor at the University of Salford Manchester (United Kingdom) for his wise advice. He is the pioneer in the formalization of information system specifications from English texts. His great experience has been very useful in carrying out this work. It is noted through the examples given that the name of the application is different, because each student group of our masters had the programming of a module of the platform.

REFERENCES

[1] W. Chama, R. Elmansouri and A. Chaoui, "Modeling and Verification Approach Based on Graph Transformation," Lecture Notes on Software Engineering, vol. 1, no. 1, pp. 39-43, 2013.

[2] D. Sadoun, Des Spécifications en Langage Naturel aux Spécifications Formelles *vi*a une Ontologie Comme Modèle Pivot, Ph.D. Thesis, Diss. Université, Paris, Sud-Paris XI, 2014.

[3] M. Ilieva and H. Boley, "Representing Textual Requirements as Graphical Natural Language for UML Diagram Generation," Proc. of the International Conference on Software Engineering and Knowledge Engineering (SEKE'08), pp. 478–483, 2008.

[4] L. Kof, "Requirements Analysis: Concept Extraction and Translation of Textual Specifications to Executable Models," Proc. of the International Conference on Application of Natural Language to Information Systems, pp. 79-90, 2009.

[5] I. S. Bajwa, B. L. Bordbar and G. Mark, "OCL Constraints Generation from Natural Language Specification," Proc. of the 14th IEEE International Conference on Enterprise Distributed Object Computing (EDOC), pp. 204-213, 2010.

[6] O. Keszocze, M. Soeken, E. Kuksa and R. Drechsler, "Lips: An IDE for Model-driven Engineering Based on Natural Language Processing," Proc. of the 1st International IEEE Workshop on Natural Language Analysis in Software Engineering (NaturaLiSE), San Francisco, CA, USA, 2013.

[7] W. F. Tichy and S. J. Koerner, "Text to Software: Developing Tools to Close the Gaps in Software Engineering," Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research, pp. 379-384, 2010.

[8] R. Laleau and A. Mammar, "An Automatic Generation of B Specifications from Well-defined UML Notations for Database Applications," Proc. of the International Symposium on Programming and Systems (ISPS), Algers, Algérie, 2001.

[9] A. J. Fougéres and P. Trigano, "Rédaction de Spécifications Formelles: élaboration à Partir des Spécifications écrites en Langage Naturel," Cognito-Cahiers Romans de Sci. Cognitives, pp. 29-36, 1997.

[10] F. Mokhtari and M. Badri, "Generating Maude Specifications From UML Use Case Diagrams," Journal of Object Technology, vol. 8, no. 2, pp. 119–136, 2009.

[11] E. Mit, Developing VDM++ Operations From UML Diagrams, Ph.D. Thesis, School of Computing, Science and Engineering University of Salford, U.K, 2007.

[12] F. Meziane and S. Vadera, "Artificial Intelligence in Software Engineering: Current Developments and

Future Prospects," Artificial Intelligence Applications for Improved Software Engineering Development: New Prospects, pp. 24-29, 2010.

[13] H. H. Ammar, W. Abdelmoez and M. S. Hamdi, "Software Engineering Using Artificial Intelligence Techniques: Current State and Open Problems," Proc. of the IEEE International Conference on Communications and Information Technology, Al-Madinah Al-Munawwarah, Saudi Arabia, pp. 24-29, 2012.

[14] N. Arman and S. Jabbarin, "Generating Use Case Models from Arabic User Requirements in a Semi-automated Approach Using a Natural Language Processing Tool," Journal of Intelligent Systems, vol. 24, no. 2, pp. 277-286, 2015.

[15] N. Alami, N. Arman and F. Khamyseh. "A Semi-automated Approach for Generating Sequence Diagrams from Arabic User Requirements Using a Natural Language Processing Tool," Proc. of the 8th IEEE International Conference on Information Technology (ICIT), Amman, Jordan, 2017.

[16] I. Nassar and F. Khamayseh, "A Semi-automated Generation of Activity Diagrams from Arabic User Requirements," NNGT International Journal on Software Engineering, vol. 2, 2015.

[17] T. S. Hoang, "An Introduction to the Event-B Modelling Method," In Book: Industrial Deployment of System Engineering Methods, Publisher: Springer-Verlag, Editors: Alexander Romanovsky and Martyn Thomas, 2013.

[18] A. Guissé, F. Lévy and A. Nazarenko, "From Regulatory Text to BRMS: How to Guide the Acquisition of Business Rules," Proc. of the International Workshop on Rules and Rule Markup Languages for the Semantic Web, Springer, pp. 77-91, 2012.

[19] K. Z. Bousmaha, M. K. Rahmouni, B. Kouninef and L. Belguith Hadrich, "A Hybrid Approach for the Morpho-Lexical Disambiguation of Arabic," Journal of Information Processing Systems (JIPS), vol. 12, no. 3, pp. 358-380, 2016.

[20] S. Boulaknadel, B. Daille and D. Aboutajdine, "A Multi-word Term Extraction Program for Arabic Language," Proceedings of the International Conference on Language Resources and Evaluation (LREC), pp. 1485-1488, Morocco, 2008.

[21] M. Diab, "Second Generation AMIRA Tools for Arabic Processing Fast and Robust Tokenization, POS tagging and Base Phrase Chunking," Proc. of the 2nd International Conference on Arabic Language Resources and Tools, pp. 285-288, Egypt, 2009.

[22] F. Z. Belkredim and A. El Sebai, "An Ontology-based Formalism for the Arabic Language Using Verbs and Their Derivatoion," Communications of the IBIMA, vol. 11, pp. 44-52, 2009.

[23] J. Singh, Mapping UML Diagrams to XML, Doctoral Dissertation, University New Delhi, 2003.

[24] I. Tounsi, H. Zied, M. H. Kacem, A. H. Kacem and K. Drira, "Using SOAml Models and Event-B Specifications for Modeling SOA Design Patterns," Proc. of the International Conference on Enterprise Information Systems (ICEIS), 2013.

[25] L. Danlos, T. Nakamura and Q. Pradet, "Vers la Création d'un Verbnet du Français," Proc. of the 21ème Conférence sur le Traitement Automatique des Langues Naturelles (TALN), Atelier Fondamen TAL, 2014.

ملخص البحث:

إن تحويل متطلبات البرمجيات الطبيعية إلى مواصفات أكثر رسمية أمر ليس بالسهل، ويمكن أن يكون تطبيقاً ممتازاً لمعالجة اللغات الطبيعية. هذه المشكلة ليست حديثة، وقد أثارت وما زالت نثير اهتماماً كبيراً؛ لأنها تفتح الباب أمام العديد من التحديات في مجالات علمية متنوعة، مثل: المعالجة الآلية للّغات، وهندسة المتطلبات، وتمثيل المعرفة، والتحقق الرسمي. تقترح هذه الورقة منصّة واستراتيجية لتحويل متطلبات البرمجيات المعينة الى مواصفات رسمية باستخدام الحدث – ب (event-B). والجدير بالذكر أن النصوص المستخدمة في هذا البحث هي نصوص باللغة العربية، الأمر الذي يُعدّ تحدّياً حقيقياً. وقد تم بناء نظام (Ar2B) واختباره، وحققت التجارب نتائج جيدة بدقة بلغت 70%.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 02, June 2020.

TAG RECOMMENDATION FOR SHORT ARABIC TEXT BY USING LATENT SEMANTIC ANALYSIS OF WIKIPEDIA

Iyad AlAgha[1] and Yousef Abu-Samra[2]

(Received: 8-Dec.-2019, Revised: 22-Jan.-2020 and 16-Feb.-2020, Accepted: 24-Feb.-2020)

ABSTRACT

Text tagging has gained a growing attention as a way of associating metadata that supports information retrieval and classification. To resolve the difficulties of manual tagging, tag recommendation has emerged as a solution to assist users in tagging by presenting a list of relevant tags. However, the majority of existing approaches for tag recommendation have focused on domain-specific tagging and tackled long-form text. Open-domain tagging can be challenging due to the lack of comprehensive knowledge and the intensive computations involved. Furthermore, tagging of short text can be problematic due to the difficulty of extracting statistical features. In terms of the language, most efforts have focused on tagging text written in English. The tagging of Arabic text has been challenged by the difficulty of processing the Arabic language and the lack of knowledge sources in Arabic.

This work proposes an approach for tag recommendation for short Arabic text. It exploits the Arabic Wikipedia as a background knowledge and uses it to generate tags in response to input short text. Latent semantic analysis is exploited to analyze Wikipedia content and find articles relevant to the input text. Then, tags are selected from the titles and categories of these articles and are ranked according to relevance.

The approach was evaluated based on experts' ratings of relevance of 993 tags. Results showed that the approach achieved 84.39% mean average precision and 96.53% mean reciprocal rank. A thorough discussion of results is given to highlight the limitations and the strengths of the approach.

KEYWORDS

Tag recommendation, Arabic, Short text, Latent semantic analysis, Wikipedia, Apache Spark.

1. INTRODUCTION

With the massive daily increase of data on the internet, especially text, automatic tagging services that attach informative and descriptive tags to texts have become a necessity for information aggregation and sharing [1]. Tagging is the practice of creating and managing labels called tags that categorize or describe the content by using simple keywords [2]. Many social media platforms, such as Twitter, Facebook and Flicker, provide their users with functionalities for manual tagging to support content categorization and search. However, manual tagging has many documented limitations, including being laborious, ambiguous and error-prone [3]-[4]. In addition, users are often permitted to use their own conventions and interests when creating tags, a thing that makes tags noisy and sparse. Alternatively, automatic text tagging has been investigated in several studies to generate tags without or with minimal intervention from the user [5]-[6]. Automatic text tagging techniques can be classified into two categories based on the source of generated tags [5]: 1) content-based tagging, which extracts tags from the target content by employing information extraction or text categorization techniques; 2) knowledge-based techniques, which use external knowledge sources, such as ontologies [7], folksonomies [8], Wikipedia [9] or Linked Open Data [10] to recommend tags related to the target content. These knowledge sources can support the tagging process by disambiguating words, inferring relationships and leading to better understandability of the target content [11].

Most of the work related to tag recommendation has been applied to long-form text [5]. When it comes to social media, text often has unique characteristics that pose additional challenges. It is often extremely short, poorly composed and tend to be more informal [12]-[13]. These challenges can obstruct the extraction of textual features of short text by applying conventional statistical techniques that work with long text [14]. In addition, most existing efforts have focused on domain-specific

[1] I. AlAgha is with Department of Computer Science, Islamic University of Gaza, Gaza, Palestine. Email: ialagha@iugaza.edu.ps

[2] Y. Abu-Samra is with Department of Computer Science, Islamic University of Gaza, Gaza, Palestine. Email: y_samra@hotmail.com

tagging. Open-domain tag recommendation can be challenging due to the lack of comprehensive knowledge sources and the intensive computations involved [10]. From the perspective of the language, the majority of works have focused on tagging text written in English or Latin languages. These works benefited from the advancement in the processing of these languages and the presence of rich English -and Latin-based knowledge resources. However, there has been little effort to support tag recommendation for Arabic texts on social media [7]. This has been challenged by the difficulties associated with the processing of the Arabic language and the lack of comprehensive knowledge sources in Arabic [15].

Driven by the above discussion, this work proposes a tag recommendation approach that generates and recommends tags for short Arabic texts. It aims to support open-domain tagging by using the Arabic version of Wikipedia as background knowledge. The choice of Arabic Wikipedia is motivated by its large coverage of various subject areas, a thing that makes it adequate for open-domain text tagging. Given an Arabic short text as input, the proposed approach will suggest a ranked list of tags with high affinity for input text. These tags are selected from Wikipedia articles that closely match with the input text. To achieve that, a topic model for Wikipedia is first created by using Latent Semantic Analysis (LSA) [16]. Without yet delving into the underlying theory, LSA is a matrix-factorization method commonly used in natural language processing and information retrieval. It seeks to better understand a corpus of documents and the relationships between the words in those documents. LSA is used to distil the Wikipedia as a corpus into a set of relevant concepts, each of which corresponds to a topic that the Wikipedia discusses. It then captures the relationships between documents and concepts and between terms and concepts. This can create a simpler representation of Wikipedia that makes it easy to find the set of articles relevant to terms in the input text. LSA is used in this work for the following reasons: First, it can create a low-dimensional representation of the corpus and thus can effectively handle huge data volumes as with Wikipedia. Second, it produces results that are more robust indicators of meaning as compared to the traditional word co-occurrence models. This is due to its ability to extract features that capture underlying latent semantic structure in the term usage across documents[17].

To handle the heavy computations involved in LSA, a cluster of computers was constructed and operated by using Apache Spark [18] as a parallel processing framework. The proposed approach was evaluated by tagging a set of 100 tweets and then assessing the relevance of generated tags. In total, 993 tags generated by our approach were rated as being "relevant" or "irrelevant" by human experts. Results showed that the approach achieved 84.39% mean average precision and 96.53% mean reciprocal rank. Results were also discussed in detail to highlight the limitations and the strengths of the approach.

2. RELATED WORK

Tag recommendation methods can be classified into four categories based on the underlying technology [5], [19]. The first category is tag co-occurrence methods, which exploit tags previously assigned to a collection of objects to suggest candidate tags to new objects [20]-[23]. They often exploit metrics related to tag frequency to suggest related tags based on tags already associated with other texts. The limitation of these works is that they assume the existence of a tagged corpus.

The second group of methods is content-based. These works do not use external corpora, but exploit the textual features of the target text, such as TF-IDF and association rules, to extract candidate terms and phrases and use them as tags [24]-[27]. The main issue with content-based techniques is that they become ineffective when applied on short texts such as tweets. They also lack novelty, because they generate tags that are already part of the target content [5]. Supervised approaches for tag recommendation also fall in this category. As recommendation can be modelled as a ranking problem, supervised approaches often use training samples consisting of candidate tags to which relevance labels are assigned as ground truth. The aim is to generate a model that maps the tag quality attributes into a relevance score or rank. Several works tried to model the tag recommendation problem as a multi-label text classification task by using different classifiers, such as Naïve Bayes [14], [28] and deep neural networks [29]-[31]. However, supervised approaches are often applicable to restricted domains and are challenged by the difficulty of obtaining labelled data.

Another category of tag recommenders include matrix factorization-based methods under which this work falls. These methods use matrix factorization to model pairwise interactions between users, items and tags, such as the ranking preferences of tags for each pair user-item [32]-[33]. Latent Semantic Analysis (LSA) and Latent Dirichet Allocation (LDA) are often used to process and decompose the co-occurrence matrix [34, 35]. LSA learns latent topics by performing a matrix decomposition (SVD) on the term-document matrix. LDA is a probabilistic topic model, where the goal is to decompose a term by the document probability distribution into two distributions: the term by topic distribution and the topic by document distribution. This work uses LSA rather than LDA, because the low-dimensional representation generated by LSA enables to easily measure similarities and no further processing is needed once it is obtained. The separation between the term, document and concept spaces in the outputs of LSA makes it easy to calculate term-to-term, document-to-document and term-to-document relevance by using cosine measure. In addition, there is a lack of well-established methods to choose the number of topics in LDA and it is unrealistic to test different numbers of topics until the best result is achieved [17].

Another common method for tag recommendation is based on graph analysis. Graph-based methods extract tags by analyzing the neighbourhood of the target text or user [36]-[37]. These methods are commonly used for tag recommendation in social networks [38]-[39], where the nodes of the graph correspond to users and edges connecting users. Collaborative filtering techniques [40]-[41] fall in this category, because they exploit the tagging history of users who are similar to the target user. These methods require the presence of graph datasets that capture the tagging behaviour and links between users.

The fourth category of methods for tag recommendation includes clustering-based methods which recommend tags based on clusters or topics of objects [42]-[43]. Given a collection of documents, these method start by applying a clustering or a classification algorithm to divide documents into groups. Then, tagging a new document is performed by first classifying that document into one or more clusters and then selecting the most relevant tags from those clusters as recommended tags. Despite the potential of clustering in reducing dimensionality of the problem, generic tags that describe the whole cluster are often generated, but are less descriptive of the specific content being tagged. These methods also do not perform well with short texts.

Besides the aforementioned categories, some works have tried to combine methods from multiple categories to improve the performance. For example, P. Lops, M. De Gemmis, G. Semeraro, C. Musto and F. Narducci [44] proposed an approach that combined collaborative filtering based on community tagging behaviour and content-based heuristic techniques. P. Symeonidis [45] combined tag clustering with matrix factorization. M. Lipczak, Y. Hu, Y. Kollet and E. Milios [46] proposed a method that extracts terms from the title and description of the target object (a content-based technique) and then expands the set of candidate tags by exploiting tag co-occurrences. Several efforts have tried to overcome the challenges of short-text processing by exploiting complementary knowledge sources, such as ontologies [7], Wikipedia [9] and Linked Open Data [10] to generate tags.

In the domain of Arabic language, several studies have explored the use of matrix factorization techniques, such as LSA and LDA, to process Arabic texts for different purposes. For example, F. S. Al-Anzi and D. AbuZeina [47] used LSA for classifying Arabic documents. They compared LSA with other classification methods and found that LSA outperforms the TF-IDF-based methods. Some works used LSA for Arabic text summarization [48]-[50] and found that LSA improved the clustering performance and resolves issues related to noisy information. M. Naili, A. H. Chaibi and H. B. Ghézala [51] used LDA to identify topics in Arabic texts and examined the impact of using different LDA parameters and Arabic stemmers. R. Mezher and N. Omar [52] approached the problem of automatic Arabic essay scoring by exploiting both syntactic features of text and LSA and found that augmenting the similarity matrix of LSA with syntactic features could improve the results. Although our work is similar to the aforementioned efforts with regard to the use of LSA on Arabic text, it differs in two aspects: 1) it has a different objective, which is tag recommendation for short Arabic text, whose features cannot be easily captured as compared to the long-form text. 2) Previous works applied LSA on the target content, but we applied it on the Arabic Wikipedia as a complementary knowledge source. 3) We tackled issues related to the processing of the enormous content of

Wikipedia by using Apache Spark as a parallel processing framework and performing dimensionality reduction.

Recently, there has been a growing interest among Arab researchers to exploit the Arabic version of Wikipedia for different purposes in computer science. Some works exploited the semi-structured content of Wikipedia to construct ontologies [53]-[54]. Others used Wikipedia features and structure to build Arabic-named entity corpora [55]-[56] or for entity linking [57]. Wikipedia-based categories have been also exploited to improve the categorization of Arabic text [58]. Some works used the Arabic Wikipedia to expand queries submitted to search engines or question answering systems [59]. The work in this paper adds to previous knowledge by extending the use of Arabic Wikipedia to support open-domain text tagging.

3. Overview of the Approach

The overall approach for tag recommendation is depicted in Figure 1 and is summarized as follows: It starts by reading, cleansing and processing the Wikipedia content to create a document-term matrix. Then, LSA is applied by performing Singular Value Decomposition (SVD) on the document-term matrix. This creates a low-rank approximation of the original matrix that models concepts in Wikipedia as well as the pairwise relations between terms, documents and concepts. The outputs of SVD will form the core of the tag recommendation system that will serve user queries as shown in the bottom part of Figure 1.

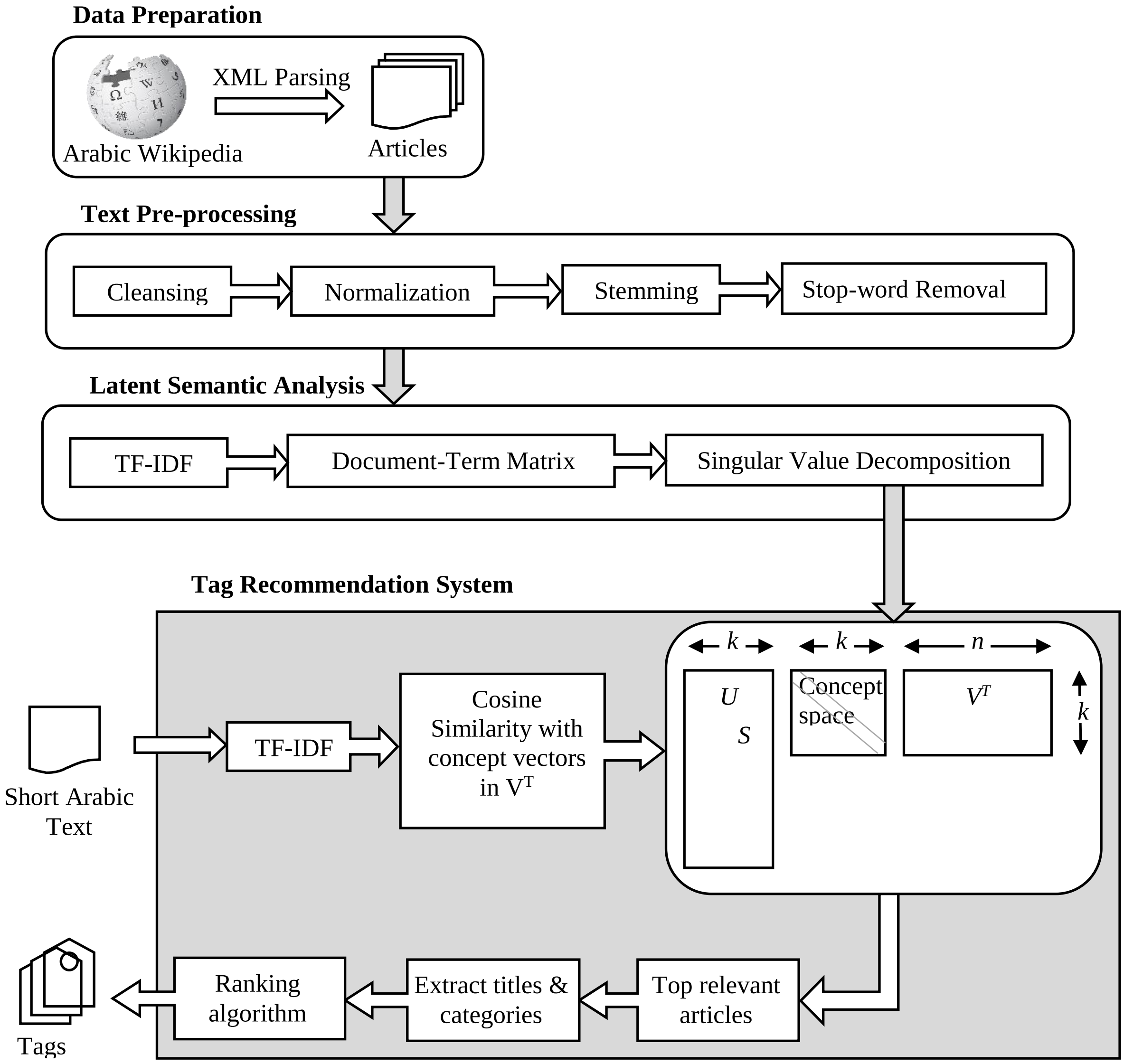


Figure 1. An approach for tag recommendation for short Arabic text based on LSA of Wikipedia.

It takes input text in Arabic, converts it into TF-IDF vector and compares it with concept vectors in SVD results. The aim is to find the concept that is most similar to the input text, which in turn will yield finding Wikipedia articles relevant to the input text. Finally, tags are selected from the titles and categories of these articles and are ranked according to their relevance scores. The approach is explained in detail in the following sections.

4. PRE-PROCESSING OF ARABIC WIKIPEDIA

Wikipedia makes its content available as XML dump files. For this work, we used the dump file of the Arabic version of Wikipedia published in October 2019. It contains 1,238,570 pages, including 435,672 actual articles and 267,580 categories. Pre-processing the Arabic Wikipedia, which is about 6.4 GB of raw text and performing the LSA computations demand a huge memory and processing power. Thus, we used a cluster of computers and Apache Spark as a cluster-computing framework. Apache Spark can distribute computational power to automatically parallelize and execute tasks on a large cluster of computers. It also provides a highly-optimized machine learning library called MLlib [60] which can perform matrix factorization on numerous datasets.

The dump file was first parsed to filter out non-informative pages, such as disambiguation pages, redirect pages, empty pages and templates. This has left only 435675 articles (about 35% of total articles) to be used for LSA. These articles were then processed to extract the textual content, the title and the associated categories. Table 1 presents some details of the pre-processed content.

Table 1. Information on the pre-processed content of Arabic Wikipedia.

Size of XML dump file	6.39 GB
No. of categories	267580
No. of pages	1238570
No. of redirect pages	437726
No. of disambiguation pages	10473
No. of template pages	345759
No. of discussion pages	181
No. of pages with empty body	8756
No. of articles used for LSA	435675

Articles were further processed by performing text pre-processing steps, including cleansing, normalization, stemming and stop-word removal. The cleansing step aims to remove words and phrases that increase the size of the corpus but do not affect the performance, such as the Latin alphabets, special characters, numbers and punctuations. This can both save space and improve fidelity. Normalization is then applied to convert the text into a more convenient and standard form. Normalization of Arabic text may be more complicated as compared to English text, because Arabic words are often connected to pronouns, prefixes and suffixes. In addition, Arabic letters such the 'أ' or 'ي' may be written in different ways. The Stanford Arabic Word Segmenter [61] was used to apply orthographic normalization to raw Arabic text. Afterwards, light stemming [62] was performed to reduce inflected or derived words to their word stem, base or root form. This is crucial, because different formations and derivations of the word may degrade the performance of LSA. We used Farasa [63] for light stemming of Arabic text.

After the pre-processing phase, each Wikipedia article was represented as a title, a list of tokens (cleansed, stemmed and non-stop-words) and a list of associated categories. The next step is related to the articles' details to vectors, which are necessary to perform SVD.

5. SINGULAR VALUE DECOMPOSITION (SVD)

Each article should be represented as a TF-IDF vector. This is done by computing the frequencies of each term within the document and within the entire Wikipedia. Since TF-IDF vectors are likely to have lots of zero values, they are converted into sparse vectors. A sparse-vector representation is more space-efficient, since it only stores the indices of the terms and non-zero values. The collection of sparse vectors form the document-term matrix, where each row corresponds to a document, each column corresponds to a term and each element indicates the importance of a term to a document.

With the document-term matrix in hand, the analysis can proceed to factorization and dimensionality reduction. MLlib, the machine learning library in Apache Spark, contains an implementation of the SVD that can process enormous matrices. SVD takes the document-term matrix and returns three matrices that approximately equal it when multiplied together, as shown in the following Equation.

$$M_{m\times n} = U_{m\times k}\, S_{k\times k}\, V^{T}{}_{k\times n}$$

where:

- M is the document-term matrix that is input to the SVD implementation.
- *m*, *n*, *k* are the number of documents, number of terms and number of concepts, respectively.
- U is an $m \times k$ matrix, where each row corresponds to a document and each column corresponds to a concept. Each element in U refers to the importance of a document to a concept. Thus, it defines a mapping between the document space and the concept space.
- V^T is a $k \times n$ matrix whose columns are basis of the term space. Each column corresponds to a term and each row corresponds to a concept. Each element in V^T refers to the importance of a term to a concept. Thus, it defines a mapping between the term space and the concept space.
- S is a $k \times k$ diagonal matrix, where each diagonal element in S corresponds to a single concept (and thus a row in V^T and a column in U). A concept captures a thread of variation in the data and often corresponds to a topic that Wikipedia discusses. Each element in S corresponds to the importance of a concept in the corpus.

Note that the three matrices are related so that each diagonal element in S corresponds to a column in U and a row in V^T. The decomposition is parameterized with a number *k*, less than or equal to *n*, which indicates how many concepts to keep around. *k* should be chosen to be less than *n* to create a low-dimensional approximation of the original document-term matrix. A key insight of LSA is that only a small number of concepts is required to ensure that the approximation will be the closest possible to the original matrix. Based on other studies that used LSA with Wikipedia [64, 65], *k* was chosen to be 1000 in our experiment, which is enough to represent the number of different topics discussed in the Arabic Wikipedia.

To illustrate how SVD outputs are interpreted in our approach to find Wikipedia articles that closely match an input text, consider the example shown in Figure 2. It shows the matrices generated after implementing SVD on five sample articles that contain seven unique terms in total. *k*, which denotes the number of concepts, is set to two. It is emphasized that this is a simplified example presented for the purpose of illustration only.

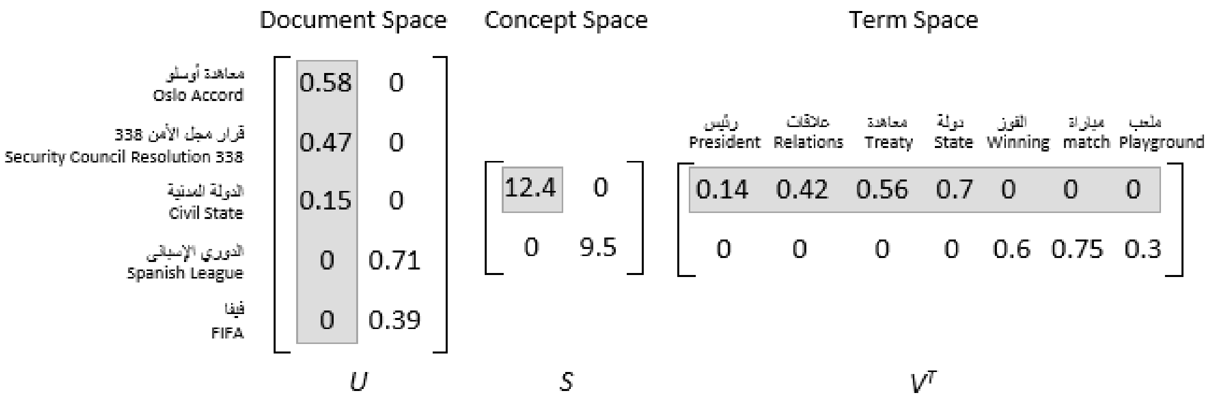


Figure 2. An example of SVD for five documents and seven terms.

Note that each diagonal element in S denotes the weight of a concept; i.e., how important the concept is to the corpus. In the given example, the shaded concept whose weight is 12.4 is the most important, because it holds the largest value. This concept is mapped to the first column in U and to the first row in V^T. Similarly, the second concept, whose weight is 9.5, is mapped to the second column in U and to the second row in V^T.

Each column in U indicates the degrees of relevance of each article to the corresponding concept in S. For example, the first column of U that is shaded in Figure 2 shows that the article titled: "معاهدة أوسلو (Oslo Accord) ", with the value of 0.58, is the most relevant to the first concept, followed by the article titled: "قرار مجلس الأمن.. (Security Council Resolution..) ". Likewise, the second column of U indicates that the article titled: " الدوري الإسباني (Spanish League)", with the value of 0.71, is the most relevant to the second concept. Zero elements in U indicate articles irrelevant to the corresponding concepts in S.

On the other hand, each row in V^T refers to the degrees of relevance of each term to the corresponding concept in S. For example, the first row in V^T that is shaded in Figure 2 reveals that the term " دولة (State)" is the most relevant to the first concept, because it has the highest value, while the term " مباراة (Match)" from the second row in V^T is the most relevant to the second concept.

Knowing this relationship between U, S and V^T matrices, SVD can tell which Wikipedia articles most closely match a set of query terms. Given a set of terms as input, the first step will be to create a TF-IDF vector of input query and find its representation as a new row of the low-rank document-term matrix approximation. Then, similar articles can be discovered by computing the cosine similarity between the new input vector and the other entries in this matrix.

6. MAPPING INPUT TEXT TO RELEVANT WIKIPEDIA ARTICLES

The implementation of SVD on the content of Arabic Wikipedia, as explained above, is performed only once and the SVD outputs are maintained in memory to form the core of the tag recommendation system shown in Figure 1. The system is now ready to take a short Arabic text as input and generate a ranked list of relevant tags as output. The input text will undergo the same text preprocessing steps applied on the Wikipedia content, including cleansing, normalization, stemming and stop-word removal. It is then converted into a sparse vector with TF-IDF weights of terms.

Let d be the TF-IDF vector of input query. We would like to map d into its representation in the SVD space, $\bar{d}$, by applying the following transformation [6]:

$$\bar{d} = S^{-1}\, U^T\, d$$

The next step is to find documents in U that are most similar to $\bar{d}$. This can be achieved by computing the cosine similarity between $\bar{d}$ and every row in U. The cosine similarity is employed, because it is simple, very efficient to evaluate especially for sparse vectors and gives normalized values in the range from 0 to 1. Documents that achieve highest cosine similarity scores refer to Wikipedia articles that are most relevant to input text. The next step will be to use these articles to generate recommended tags.

7. TAG GENERATION AND RANKING

Up to this point, input text should have been matched with relevant Wikipedia articles by using SVD outputs. These articles are ranked from the most to the least relevant based on the similarity to input text. Then, recommended tags are extracted from the titles and categories of these articles. While titles tend to be more specific and unique, categories are often more generic and referral to broader subject areas. Thus, tags selected from both titles and categories make a list of diversified and complementary descriptors covering both specific and broad subjects pertaining to the input text. However, number of titles and categories obtained from articles could be large. Thus, they should be filtered and ranked to get only the most relevant ones. The algorithm we use to filter and rank titles and categories can be explained as follows:

Let D = (d_1,d_2,..d_i,.., d_n) be an ordered set of documents obtained from SVD, where i indicates the rank of d; i.e., its relevance to the input text. Let A={a_1, a_2,..,a_m} be the set of terms in input text and B={b_1, b_2,..,b_k} the set of terms in the title of d. Then, each title of d_i is scored using the following Equation:

$$Score\ of\ t_i = \frac{1 + |A \cap B|}{1 + \log i}$$

where t_i is the title of d_i. $|A \cap B|$ is the number of terms occurring in both A and B. Based on this equation, each title is weighted based on criteria that consider both the overlap between the title and the input text and the rank of the document. That is, a title is scored higher if it shares more terms with input text and belongs to a highly ranked article based on SVD. Note that the above measure assures that each title has a non-zero score even if it has no overlap with the input text. This is necessary, because a title that does not overlap with the input text can still be relevant. Scores of titles are then normalized by being rescaled to have values between 0 and 1.

After scoring titles, we now move on to score categories. Since documents obtained from SVD can collectively have a large number of categories, it is necessary to choose only most relevant ones. In addition, we cannot rely on the overlapping of texts to filter categories as we did with titles. Categories are less likely to be included in the target text, because they often describe broader subjects or general classifications. Instead, categories are filtered based on their frequency in articles so that categories that occur more often are prioritized. The score of a category c is computed using the following Equation:

$$Score\ of\ c = \frac{frequency\ of\ c\ in\ D}{|D|}$$

where D is the set of documents obtained from SVD. Based on this equation, the score of a category ranges from 0 to 1, where the category gets the score of 1 if it appears in all documents in D.

Finally, titles and categories are grouped and ordered based on their normalized scores. In our experiment, the number of recommended tags for each input text was limited to the top ten tags. Table 2 shows the tagging results for a sample input tweet including the top scored titles and categories. Only shaded tags, which got the highest scores, are recommended to the end user.

Table 2. A sample input tweet with the tagging results as generated by our approach.

(The difference between the programmer and the graphic designer) الفرق بين المبرمج ومصمم الجرافيك	
Top titles	Top categories
(Graphic Design) تصميم الجرافيك	(Computer Science) علم الحاسوب
(Programmer) مبرمج	(Graphic Design) تصميم الجرافيك
فريق العمل لإنتاج برمجيات الوسائط المتعددة (Multimedia Production Team)	(Computer Professions) مهن الحاسوب
(Computer Science) علم الحاسوب	(Programmers) مبرمجون
(Graphic Designer) مصمم جرافيك	(Communication Design) تصميم الاتصال
(Graphics) رسوميات	(Computer Engineering) هندسة الحاسوب
(Information Design) تصميم المعلومات	(Media Careers) مهن وسائل الإعلام

8. EVALUATION

The objective of the evaluation is to assess the extent to which our tag recommendation approach can suggest tags relevant to the input Arabic text. Existing approaches for tag recommendation have been evaluated either by exploiting tags previously assigned by the users as a ground truth [66]-[67] or manually by relying on external users to evaluate the recommendations [57], [68]-[69]. In this work, we used the second approach, because we are not aware of any dataset of pre-tagged Arabic short texts that we can compare our results to. In addition, we emphasize that comparing LSA with other text-similarity measures is out of the scope of this work. The differences between semantic similarity measures have been experimentally explored in several studies [70]-[71]. Instead, we focus on the problem of short Arabic text tagging and use LSA as an unsupervised approach to achieve this purpose due to its output that facilitates similarity calculations.

Thus, we created a dataset consisting of 100 tweets selected randomly from three different domains: sports, technology and news. The tweets were classified as follows: sports: 36 tweets, technology: 41 tweets and news: 23 tweets. These tweets were used as input to the recommendation approach. The output for each tweet was a set of tags ordered by the system based on the relevance to the input tweet. Only top ten tags per tweet were considered in the evaluation. Thus, 1000 recommendations were collected in total for the 100 input tweets. These recommendations were then rated by human experts. As tweets were categorized into three distinct domains, each group of tweets was rated by two experts in each domain. Experts rated each tweet as either "relevant" or "irrelevant". Only tags that both experts agreed upon were considered in the evaluation. Finally, 993 tags rated by experts were considered for the evaluation process. Table 3 shows sample tweets from our dataset. The complete dataset including the tweets, the generated tags and the ratings of experts can be downloaded from: https://github.com/YousefSamra/ShortTextTagging and instructions can be found on: http://tiny.cc/op50iz.

Table 3. Sample tweets from the dataset.

Subject	Tweet
Sports	موقعة قويّة بين تشيلسي ومان سيتي وليفربول يترصّد A strong match between Chelsea and Man City and Liverpool stalks
Technology	سيانوجين مود رائدة تطوير رومات الأندرويد CyanogenMod is a pioneer in the development of Android ROM
News	فلسطين المحتلة: الصحفي محمد القيق يواصل إضرابه عن الطعام بسجون الصهاينة Occupied Palestine: Journalist Muhammad Al-Qeeq continues his hunger strike in Israeli prisons

8.1 Experimental Settings

The experiment was carried out in a computer lab consisting of 20 laptops, all with the specifications shown in Table 4. The laptops were all connected to a single LAN and controlled and scheduled by Apache Spark framework installed on a master computer. The cluster was used to operate the code for pre-processing the Wikipedia content and performing LSA. After getting the outputs of SVD, the system became ready to take a short text as input and to produce tags rapidly as outputs.

Table 4. Specifications of laptops used in the experiment.

Machine	HP laptop
CPU	Core i7 2.6 GHz
RAM	6 GB
OS	Windows 10, 64bit

8.2 Evaluation Metrics

In tag recommendation, the most important result for the end user is to receive a list of recommendations, ordered from the most to the least relevant. So, we used three metrics that are commonly used to evaluate recommendation systems [72]-[73]. These metrics are:

Precision at position k (P@k), where k denotes the number of recommended tags for each tweet. We aim to explore how the precision is affected when changing the number of tags to be examined. P@k is computed using the following equation:

$$P@k = \frac{number\ of\ relevant\ tags\ in\ top\ k\ positions}{k}$$

Mean Average Precision (MAP): The average precision for the query q is computed using the following equation:

$$AP(q) = \frac{\sum_{k=1}^{m} P@k(q)}{number\ of\ relevant\ tags\ for\ q}$$

where, m is the total number of results for the query q and P@k is the precision at position k. The mean average precision for a set of queries is the mean of the average precision scores for each query:

$$MAP = \frac{1}{N}\sum_{q=1}^{N} AP(q)$$

Mean Reciprocal Rank (MRR): While the first two metrics emphasize the quality of the top k tags, the MRR focuses on a practical goal, which is how deep the user has to go down a ranked list to find one useful tag [74]. MRR is the average of the reciprocal ranks of results for a sample of queries N and is calculated using the following Equation:

$$MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank_i}$$

where $rank_i$ refers to the rank position of the first relevant tag for the i-th query.

Average processing time: To assess the efficiency of the system, the average time required to tag each tweet was recorded .This is the time elapsed from inputting each tweet until the tags are generated.

The above evaluation metrics were then calculated according to the rates obtained from the experts. Table 5 shows how the metrics were calculated for a sample tweet. It shows the ordered list of recommended tags (10 tags), along with the experts' ratings of each tag and the calculated values of metrics.

Table 5. Expert ratings and evaluation metrics for a sample tweet.

RR	AP@k	P@k	Experts' rating (relevant =1, irrelevant = 0)	واتساب 2017: توقف خدمة واتس اب عن العمل على بعض الهواتف. اكتشف ان كان هاتفك من القائمة (WhatsApp 2017: WhatsApp has stopped working on some phones. Find out if your phone is on the list)	
1	0.82602	1	1	واتسآب (WhatsApp)	1
		0.5	0	سناب شات (Snapchat)	2
		0.666667	1	تراسل فوري (Instant Messaging)	3
		0.75	1	برمجيات آي أو إس (iOS Software)	4
		0.8	1	برمجيات أندرويد (Android Software)	5
		0.833333	1	برمجيات متعددة المنصات (Multi-platform Software)	6
		0.857143	1	برمجيات اتصال (Communication Software)	7
		0.875	1	مراسلة فورية (Instant Messaging)	8
		0.777778	0	برمجيات بلاك بيري (Blackberry Software)	9
		0.7	0	برمجيات سيمبيان (Symbian Software)	10

8.3 Results and Discussion

Table 6 shows the evaluation results. A total number of 933 tags were gathered and assessed by experts. 658 out of 933 were assessed as relevant, giving an AP@10 of 71.94%. Our approach also achieved a MAP of 84.39% and a MRR of 96.53%, indicating that the tagging approach achieved high precision.

Table 6. Evaluation results.

Number of generated tags @ k=10	933
Number of correct tags	658
AP@10	71.94%
MAP	84.39%
MRR	96.53%
Avg. processing time	2.54 sec.

In addition, the average processing time was 2.54 seconds. This result indicates that the approach is suitable for real-time usage, especially when considering the huge size of Wikipedia content and the intensive computations involved. When using sufficient computing and storage resources, LSA becomes an efficient text-mining technique, because it creates and uses a low-dimensional representation of the original document-term matrix [16].

The tagging performance was also explored across different subject domains. As shown in Table 7, the values of MAP and MRR for the three subject domains were close, indicating that the approach performed well in the three domains. This result indicates that the Arabic Wikipedia can be a reasonable choice as a background knowledge for open-domain tagging due to its generality and coverage of a wide range of topics.

Table 7. Results across different subjects.

Subject	No. of tweets	MAP	MRR
Sports	36	80.81%	95.46%
Technology	41	85.85%	96.83%
News	23	87.12%	97.83%

Figure 3 depicts how the average precision (AP@k) changes as k changes from 1 to 10. The precision is highest when k=1 and then declines consistently as k increases. This indicates that the approach often orders tags according to their relevance so that most relevant tags come on the top of the list.

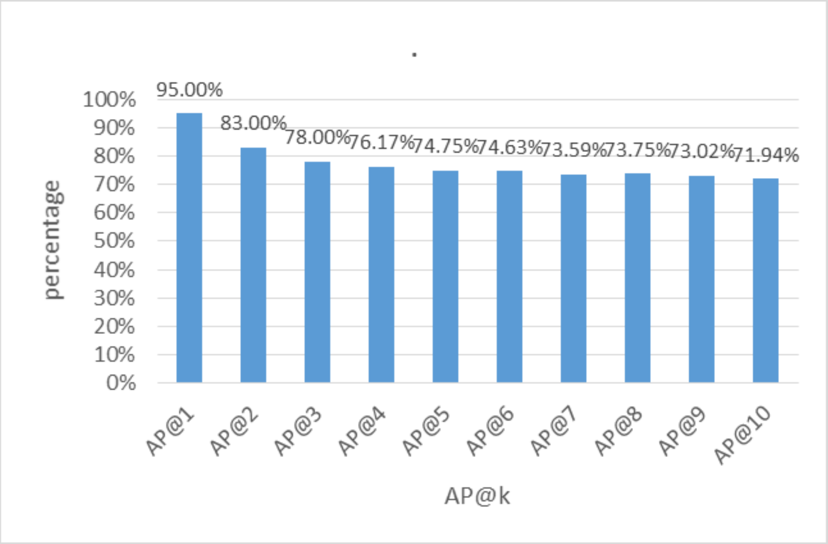


Figure.3 AP(1-100)@k(1-10).

To further explain our results, generated tags were inspected thoroughly to identify the main sources of strengths and weaknesses. The following discusses the main observations:

Term ambiguity: One challenge of any automatic tagging service is the ability to resolve word ambiguity and pick tags that conform to the context of the input text. The approach showed good performance with respect to handling polysemy; i.e., recognizing terms that have different meanings in different contexts, as was evident from several examples. Consider the following tweet: " زيدان: أخشى أن يسقط ريال مدريد مجددا - Zidane: I am afraid Real Madrid will fall again": The name " زيدان (Zidane)" could refer to many public figures, such as a French former player, a philosopher and an actor. However, the approach suggested the tag "زين الدين زيدان (Zinedine Zidane)" which refers to the intended person. In another example, the tweet " مواعيد العمل في معبر الكرامة بعد غد.. - Working hours at Al-Karamah crossing after tomorrow.." was associated with the tag "جسر الملك حسين (King Hussein Bridge)" which is the alternative name of Al-Karamah crossing between Palestine and Jordan. Tags in the previous examples were consistent with the context and the intended meanings of input terms. This can be attributed to the LSA's ability to base similarity scores on a deeper understanding of the corpus. LSA can capture Wikipedia concepts and associate articles with relevant concepts. When the input text is compared with articles, it will be able to recover the relationship between terms, such as " ريال مدريد (Real Madrid)" and " زيدان (Zidane)" based on the co-occurrence of both terms in articles associated with the same concepts. However, there were a few cases where the approach failed to resolve ambiguity and thus generated false tags. For example, the tweet: " ملعب كرة قدم في قطر على شكل بيت شعر..- A football stadium in Qatar in the form of Bedouin tent.." was tagged with: "قافية (rhyme)", "إيقاع شعري (Poetric rhythm)", "شعر (Poetry)" and "ملاعب كرة قدم في قطر (Football stadiums in Qatar)". While the first three tags are not related to the context, the fourth is relevant, but is ranked lower. This result can be explained by the lack of articles discussing both sports and football stadiums in Qatar. The ability of LSA to handle polysemy is often proportional to the number and depth of articles covering the ambiguous terms. In addition, our implementation of LSA does not consider the diacritization of Arabic[75], which is the process of restoring the diacritical marks, for handling morphological and syntactic ambiguity. Thus, the approach was not able to distinguish the difference between the words "شِعْر (Poetry)" and "شَعْر (Hair)".

Synonymy: One of the advantages of LSA is its potential to recognize synonyms and alternative words by condensing related terms [76]. This advantage was evident in many results, where the approach could recommend synonyms of terms from the input text. For example, the tweet " #بريطانيا عاجل | إغلاق محطة #لندن بريدج للقطارات والمنطقة المحيطة بسبب تحذير أمني - Britain urgent: London Bridge station and the surrounding area have been closed due to a security warning ", was tagged with the terms: "المملكة المتحدة (UK)","إنجلترا (England)". Similarly, the tweet "المعالج أكثر المنتجات تعقيداً اليوم.. - The processor is the most sophisticated product today.." was tagged with the following terms " وحدة المعالجة المركزية (CPU)" and " تصميم وحدة المعالجة المركزية (CPU Design)", which are all synonyms of the term " (The Processor) المعالج". A common limitation of content-based recommendation techniques is the lack of novelty, because they extract tags from the own content of the target text. This limitation significantly diminished in our approach, because tags were extracted from Wikipedia articles rather than from the target text.

Tag selection procedure: As explained earlier, the proposed approach uses a tag selection procedure that considers both titles and categories of articles. This combination of titles and categories often resulted in tags that varied in generality and covered both narrow and broad topics. For example, the tweet "غسان كنفاني: روائي وقاص وصحفي فلسطيني تم اغتياله على يد الموساد الصهيوني - Ghassan Kanafani: a Palestinian novelist, storyteller and journalist assassinated by the Mossad " had the following tags in order: "غسان كنفاني (Ghassan Kanafani)", "أدباء وكتاب فلسطين- Authors and writers of Palestine" and "الصراع العربي الإسرائيلي- Arab-Israeli Conflict". The first tag is a title of an article, while the rest are categories. In another example: " شركة أوراكل المتخصصة بحلول قواعد البيانات وتكنولوجيا المعلومات بلغت قيمتها 168 مليار دولار - Oracle, which specializes in database solutions and IT, valued at $ 168 billion " was tagged with the following terms in order: "أوراكل (Oracle)"," نظام إدارة قواعد البيانات العلائقية (Relational Database System)","تقنية" (Technique) and "بيانات ضخمة (Big Data)". The first two tags in the former example refer to titles, while the last two are categories. In both examples, titles are generally more concise and descriptive than categories, while categories are more generic and can give a broad insight into the subject area pertaining to the tweets. This can fulfil the interests of users that may vary with respect to the desired specificity of results.

9. CONCLUSION AND FUTURE WORK

This work presents an approach for tag recommendation of short Arabic text. It uses LSA to uncover the latent concepts within Wikipedia and to provide scores of similarity between documents, concepts and terms. The LSA model was used to find Wikipedia articles that best match with the target text. Tags are selected from titles and categories of retrieved articles to provide recommendations covering both specific and broad topics. In addition, selected tags are ranked based on several factors that include the overlap between the title and the input text, the rank of corresponding articles and the frequency of category in articles.

The evaluation of the approach by assessing resultant recommendations against expert judgments has proved the effectiveness and efficiency of the approach. In addition, the inspection of results has provided an insight into the strengths and weaknesses of the approach. The contribution of this work is two-folded: First, it tackles the problem of open-domain and real-time tag recommendation for short Arabic text, which is a problem that remains briefly addressed in the literature. Second, it exploits Wikipedia as a comprehensive source of tags and analyzes it by using LSA to match the input query with relevant articles. To our knowledge, little effort has been devoted to leveraging the Arabic version of Wikipedia with LSA for tag recommendation.

There are many directions to extend this work: First, we aim to improve the tagging results by testing techniques other than LSA, such as LDA and supervised approaches. Second, we may explore the unique challenges associated with the Arabic language, such as the diacritization of text and its impact on results. Third, we may explore the use of LSA with Wikipedia for other applications, such as question answering and text summarization. Third, we aim to deploy the proposed tagging service and integrate it with social media platforms in order to evaluate it at a larger scale.

REFERENCES

[1] V. Oliveira, G. Gomes, F. Belém, W. Brandao, J. Almeida, N. Ziviani and M. Gonçalves, "Automatic Query Expansion Based on Tag Recommendation," Proceedings of the 21st ACM International Conference on Information and Knowledge Management, pp. 1985-1989, 2012.

[2] O. Nov, M. Naaman and C. Ye, "What Drives Content Tagging: The Case of Photos on Flickr," ACM, pp. 1097-1100, 2008.

[3] M. R. Bouadjenek, H. Hacid and M. Bouzeghoub, "Social Networks and Information Retrieval, How Are They Converging? A Survey, a Taxonomy and an Analysis of Social Information Retrieval Approaches and Platforms," Information Systems, vol. 56, pp. 1-18, 2016.

[4] G. Sriharee: "An Ontology-based Approach to Auto-tagging Articles," Vietnam Journal of Computer Science, vol. 2, no. 2, pp. 85-94, 2015.

[5] F. M. Belém, J. M. Almeida and M. A. Gonçalves, "A Survey on Tag Recommendation Methods," Journal of the Association for Information Science and Technology, vol. 68, no. 4, pp. 830-844, 2017.

[6] O. Vechtomova, "Introduction to Information Retrieval," Proc. of the 40th European Conference on IR Research, 2009.

[7] I. Al-Agha and O. Abu-Dahrooj: "Multi-level Analysis of Political Sentiment Using Twitter Data: A Case Study of the Palestinian-Israeli Conflict," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 5, no. 3, 2019.

[8] "Latent Semantic Indexing," [Online], Availab.:https://en.wikipedia.org/wiki/Latent_semantic_analysis.

[9] H.-K. Hong, G.-W. Kim and D.-H. Lee: "Semantic Tag Recommendation Based on Associated Words Exploiting the Interwiki Links of Wikipedia," Journal of Information Science, vol. 44, no. 3, pp. 298-313, 2018.

[10] L. Jayaratne, "Content Based Cross-domain Recommendation Using Linked Open Data," GSTF Journal on Computing, vol. 5, no. 3, 2017.

[11] S. Vairavasundaram, V. Varadharajan, I. Vairavasundaram and L. Ravi, "Data Mining-based Tag Recommendation System: An Overview," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 5, no. 3, pp. 87-112, 2015.

[12] W. Guo, H. Li, H. Ji and M. T. Diab, "Linking Tweets to News: A Framework to Enrich Short Text Data in Social Media", ACL, vol. 1, pp. 239-249, 2013.

[13] S. Garcia Esparza, M. P. O'Mahony and B. Smyth, "Towards Tagging and Categorization for Micro-blogs," Paper presented at the 21st National Conference on Artificial Intelligence and Cognitive Science (AICS 2010), Galway, Ireland, 30 August-1 September, 2010.

[14] R. Dovgopol and M. Nohelty, "Twitter Hash Tag Recommendation," arXiv preprint arXiv:1502.00094, 2015.

[15] I. M. AlAgha and A. Abu-Taha, "AR2SPARQL: An Arabic Natural Language Interface for the Semantic Web," AR2SPARQL: An Arabic Natural Language Interface for the Semantic Web, vol. 124, no. 18, 2015.

[16] T. K. Landauer, D. S. McNamara, S. Dennis and W. Kintsch, Handbook of Latent Semantic Analysis, Psychology Press, 2013.

[17] T. Cvitanic, B. Lee, H. I. Song, K. Fu and D. Rosen, "LDA *v.* LSA: A Comparison of Two Computational Text Analysis Tools for the Functional Categorization of Patents", ICCBR Workshops, 2016.

[18] M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman and M. J. Franklin, "Apache Spark: A Unified Engine for Big Data Processing," Communications of the ACM, vol. 59, no. 11, pp. 56-65, 2016.

[19] R. Singh and A. Rani, "A Survey on the Generation of Recommender Systems," International Journal of Information Engineering and Electronic Business, vol. 9, no. 3, pp. 26-35, 2017.

[20] C. Wartena, R. Brussee and M. Wibbels, "Using Tag Co-occurrence for Recommendation," Proc. of the 9th International Conference on Intelligent Systems Design and Applications (ISDA), Pisa, Italy, pp. 273-278, 2009.

[21] R. Damaševicius, R. Valys and M. Woźniak, "Intelligent Tagging of Online Texts Using Fuzzy Logic," Proc. of IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1-8, 2016.

[22] G. V. Menezes, J. M. Almeida, F. Belém, M. A. Gonçalves, A. Lacerda, E. S. De Moura, G. L. Pappa, A. Veloso and N. Ziviani, "Demand-driven Tag Recommendation," Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part II, pp. 402-417, 2010.

[23] K. Yanai, "VisualTextualRank: An Extension of Visualrank to Large-scale Video Shot Extraction Exploiting Tag Co-occurrence," IEICE Transactions on Information and Systems, vol. 98, no. 1, pp. 166-172, 2015.

[24] M. P. Lipczak and E. Milios, "Efficient Tag Recommendation for Real-life Data," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 3, no. 1, pp. 1-21, 2011.

[25] Y. Wu, Y. Yao, F. Xu, H. Tong and J. Lu, "Tag2word: Using Tags to Generate Words for Content Based Tag Recommendation," Proc. of the 25th International ACM Conference (CIKM '16), pp. 2287-2292, 2016.

[26] J. Wang, L. Hong and B. D. Davison, "Tag Recommendation Using Keywords and Association Rules," RSDC'09, pp. 1-14, 2009.

[27] F. M. Belém, E. F. Martins, J. M. Almeida and M. A. Gonçalves, "Personalized and Object-centered Tag Recommendation Methods for Web 2.0 Applications," Information Processing & Management, vol. 50, no. 4, pp. 524-553, 2014.

[28] I. Katakis, G. Tsoumakas and I. Vlahavas, "Multi-label Text Classification for Automated Tag Suggestion", ECML/PKDD, pp. 1-9, 2008.

[29] Y. Gong and Q. Zhang: "Hashtag Recommendation Using Attention-based Convolutional Neural Network," Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI-16), pp. 2782-2788, 2016.

[30] Y. Wang, J. Li, I. King, M. R. Lyu and S. Shi, "Microblog Hashtag Generation *via* Encoding Conversation Contexts," arXiv preprint arXiv:1905.07584, 2019.

[31] H. T. Nguyen, M. Wistuba, J. Grabocka, L. R. Drumond and L. Schmidt-Thieme, "Personalized Deep Learning for Tag Recommendation", Proc. of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, pp. 186-197, 2017.

[32] Y. Yang, L. Han, Z. Gou, B. Duan, J. Zhu and H. Yan, "Tagrec-CMTF: Coupled Matrix and Tensor Factorization for Tag Recommendation," IEEE Access, vol. 6, pp. 64142-64152, 2018.

[33] C. Lu, B. Shen, L. Zhang and J. Allebach, "Tag Recommendation *via* Robust Probabilistic Discriminative Matrix Factorization," Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1170-1174, 2016.

[34] J. Yao, Y. Wang, Y. Zhang, J. Sun and J. Zhou, "Joint Latent Dirichlet Allocation for Social Tags," IEEE Transactions on Multi-media, vol. 20, no. 1, pp. 224-237, 2017.

[35] M. A. Masood, R. A. Abbasi, O. Maqbool, M. Mushtaq, N. R. Aljohani, A. Daud, M. A. Aslam and J. S. Alowibdi, "MFS-LDA: A Multi-feature Space Tag Recommendation Model for Cold Start Problem," Program, vol. 51, no. 3, pp. 218-234, 2017.

[36] T.-A. N. Pham, X. Li, G. Cong and Z. Zhang, "A General Graph-based Model for Recommendation in Event-based Social Networks," International Conference on Data Engineering, pp. 567-578, 2015.

[37] M. Hmimida and R. Kanawati, "A Graph-coarsening Approach for Tag Recommendation," Proc. of the International World Wide Web Conferences Steering Committee, pp. 43-44, 2016.

[38] Y. Chen, H. Dong and W. Wang, "Topic-graph Based Recommendation on Social Tagging Systems: A Study on Research Gate," ACM, pp. 138-143, 2018.

[39] M. Rawashdeh, M. F. Alhamid, J. M. Alja'am, A. Alnusair and A. El Saddik, "Tag-based Personalized Recommendation in Social Media Services," Multimedia Tools and Applications, vol. 75, no. 21, pp. 13299-13315, 2016.

[40] M. A. Chatti, S. Dakova, H. Thüs and U. Schroeder, "Tag-based Collaborative Filtering Recommendation in Personal Learning Environments," IEEE Transactions on Learning Technologies, vol. 6, no. 4, pp. 337-349, 2013.

[41] S. Panigrahi, R. K. Lenka and A. Stitipragyan, "A Hybrid Distributed Collaborative Filtering Recommender Engine Using Apache Spark," Procedia Comp. Science, vol. 83, pp. 1000-1006, 2016.

[42] Y. Song, L. Zhang and C. L. Giles, "Automatic Tag Recommendation Algorithms for Social Recommender Systems," ACM Transactions on the Web (TWEB), vol. 5, no. 1, p. 4, 2011.

[43] R. Krestel and P. Fankhauser, "Personalized Topic-based Tag Recommendation," Neurocomputing, vol. 76, no. 1, pp. 61-70, 2012.

[44] P. Lops, M. De Gemmis, G. Semeraro, C. Musto and F. Narducci, "Content-based and Collaborative Techniques for Tag Recommendation: An Empirical Evaluation," Journal of Intelligent Information Systems, vol. 40, no. 1, pp. 41-61, 2013.

[45] P. Symeonidis, "ClustHOSVD: Item Recommendation by Combining Semantically Enhanced Tag Clustering with Tensor HOSVD", IEEE Transactions on Systems, Man and Cybernetics: Systems, vol. 46, no. 9, pp. 1240-1251, 2015.

[46] M. Lipczak, Y. Hu, Y. Kollet and E. Milios, "Tag Sources for Recommendation in Collaborative Tagging Systems," ECML PKDD Discovery Challenge, vol. 497, pp. 157-172, 2009.

[47] F. S. Al-Anzi and D. AbuZeina, "Toward an Enhanced Arabic Text Classification Using Cosine Similarity and Latent Semantic Indexing," Journal of King Saud University-Computer and Information Sciences, vol. 29, no. 2, pp. 189-195, 2017.

[48] H. Froud, A. Lachkar and S. A. Ouatik, "Arabic Text Summarization Based on Latent Semantic Analysis to Enhance Arabic Documents Clustering," arXiv preprint arXiv:1302.1612, 2013.

[49] K. Al-Sabahi, Z. Zhang, J. Long and K. Alwesabi, "An Enhanced Latent Semantic Analysis Approach for Arabic Document Summarization," Arabian Journal for Science and Engineering, vol. 43, no. 12, pp. 8079-8094, 2018.

[50] H. Alazzam and A. Alsmady, "A Distributed Arabic Text Classification Approach Using Latent Semantic Analysis for Big Data," Proc. of the 12th IEEE International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT), pp. 58-61, 2017.

[51] M. Naili, A. H. Chaibi and H. B. Ghézala, "Empirical Study of LDA for Arabic Topic Identification," HAL Id: hal-01444574, 2016.

[52] R. Mezher and N. Omar, "A Hybrid Method of Syntactic Feature and Latent Semantic Analysis for Automatic Arabic Essay Scoring," Journal of Applied Sciences, vol. 16, no. 5, p. 209, 2016.

[53] N. I. Al-Rajebah and H. S. Al-Khalifa, "Extracting Ontologies from Arabic Wikipedia: A Linguistic Approach," Arabian Journal for Science and Engineering, vol. 39, no. 4, pp. 2749-2771, 2014.

[54] M. M. Boudabous, L. H. Belguith and F. Sadat, "Exploiting the Arabic Wikipedia for Semi-automatic Construction of a Lexical Ontology," International Journal of Metadata, Semantics and Ontologies, vol. 8, no. 3, pp. 245-253, 2013.

[55] F. Alotaibi and M. Lee, "Mapping Arabic Wikipedia into the Named Entities Taxonomy", Proceedings of COLING 2012, pp. 43-52, 2012.

[56] M. Al-Smadi, B. Talafha, O. Qawasmeh, M. N. Alandoli, W. A. Hussien and C. Guetl, "A Hybrid Approach for Arabic Named Entity Disambiguation," Proc. of the 15th International Conference on Knowledge Technologies and Data-drive, ACM, 2015.

[57] F. Fayad and I. AlAgha, Automatic Linking of Short Arabic Texts to Wikipedia, M.Sc. Thesis, Faculty of Information Technology, The Islamic University-Gaza, Palestine, 2013.

[58] A. Yahya and A. Salhi, "Arabic Text Categorization Based on Arabic Wikipedia," ACM Transactions on Asian Language Information Processing (TALIP), vol. 13, no. 1, p. 4, 2014.

[59] A. Mahgoub, M. Rashwan, H. Raafat, M. Zahran and M. Fayek, "Semantic Query Expansion for Arabic Information Retrieval," Arabic Natural Language Processing Workshop (EMNLP), Doha, Qatar, pp. 87-92, 2014.

[60] X. Meng, J. Bradley, B. Yuvaz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. Tsai, M. Amde and S. Owen, "MLlib: Machine Learning in Apache Spark," JMLR, vol. 17, no. 34, pp. 1-7, 2016.

[61] W. Monroe, S. Green and C. D. Manning, "Word Segmentation of Informal Arabic with Domain Adaptation," Proceedings of the 52nd Annual Meeting of the Association for Computational Sciences, vol. 2, pp. 206-211, 2014.

[62] L. S. Larkey, L. Ballesteros and M. E. Connell, "Light Stemming for Arabic Information Retrieval," Arabic Computational Morphology, (Springer, Dordrecht,), pp. 221-243, 2007.

[63] K. Darwish and H. Mubarak, "Farasa: Fast and Accurate Arabic Word Segmenter," [Online], Available: http://alt.qcri.org/farasa/segmenter.html, Accessed: 9 Feb. 2017.

[64] E. Gabrilovich and S. Markovitch, "Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis," Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07), PP. 1606-1611, 2007.

[65] D. Ștefănescu, R. Banjade and V. Rus, "Latent Semantic Analysis Models on Wikipedia and Tasa," Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC'14), pp. 1417-1422, 2014.

[66] F. M. Belém, C. S. Batista, R. L. Santos, J. M. Almeida and M. A. Gonçalves, "Beyond Relevance: Explicitly Promoting Novelty and Diversity in Tag Recommendation," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 7, no. 3, p. 26, 2016.

[67] J. Chakraborty and V. Verma, "Diversification in Tag Recommendation System Using Binomial Framework," Information and Communication Technology for Sustainable Development, Springer, pp. 423-430, 2018.

[68] B. Bi and J. Cho, "Automatically Generating Descriptions for Resources by Tag Modeling," Proceedings of the 22nd ACM International Conference on Information & Knowledge Management (CIKM '13), pp. 2387-2392, 2013.

[69] R. Prokofyev, A. Boyarsky, O. Ruchayskiy, K. Aberer, G. Demartini and P. Cudré-Mauroux, "Tag Recommendation for Large-scale Ontology-based Information Systems," Proc. of the International Semantic Web Conference, Springer, pp. 325-336, 2012.

[70] N. Niraula, R. Banjade, D. Ştefănescu and V. Rus, "Experiments with Semantic Similarity Measures Based on LDA and LSA," Proc. of the International Conference on Statistical Language and Speech Processing, Springer, pp. 188-199, 2013.

[71] C.-G. Chiru, T. Rebedea and S. Ciotec, "Comparison between LSA-LDA-lexical Chains," Proceedings of the 10th International Conference on Web Information Systems and Technologies (WEBIST), pp. 255-262, 2014.

[72] M. Allahyari and K. Kochut, "Semantic Tagging Using Topic Models Exploiting Wikipedia Category Network", 2016 IEEE 10th International Conference on Semantic Computing (ICSC), pp. 63-70, 2016.

[73] T. Bogers and A. van den Bosch: "Recommending Scientific Articles Using Citeulike," Proceedings of the ACM Conference on Recommender Systems, pp. 287-290, 2008.

[74] M. Sun, Y.-N. Chen and A. I. Rudnicky: "HELPR, A Framework to Break the Barrier across Domains in Spoken Dialog Systems," Dialogues with Social Robots, Springer, pp. 257-269, 2017.

[75] M. Maamouri, A. Bies and S. Kulick, "Diacritization: A Challenge to Arabic Treebank Annotation and Parsing," Proceedings of the Conference of the Machine Translation SIG of the British Computer Society, 2006.

[76] T. K. Landauer, "LSA as a Theory of Meaning," Handbook of Latent Semantic Analysis, vol. 3, 2007.

ملخص البحث:

لقد اكتسب موضوع عمل بطاقات اقتباس من النصوص اهتماماً متزايداً كطريقة للربط بين البيانات من شأنها أن تدعم استرجاع المعلومات وتصنيفها. ولحلّ المشكلات المتعلقة بالقيام بذلك يدوياً، ظهرت تقنيات لتسهيل الأمر على المستخدمين عن طريق توفير قائمة من البطاقات التي تقتبس من النصوص.

وتجدر الإشارة الى أن غالبية الطرق القائمة التي تستخدم لهذا الغرض إنما تركز على الحقل أو المجال، كما أنها تعالج نصوصاً طويلة. وتشكل الطرق المعتمدة على الحقل او المجال تحدياتٍ جمّة بسبب نقص المعرفة الشاملة والحسابات المعقدة التي تتضمنها.

علاوة على ذلك، قد ينطوي التعامل مع النصوص القصيرة على بعض الإشكاليات نظراً لصعوبة استخلاص السّمات الإحصائية منها. ومن حيث اللغة، انصبت الجهود المبذولة بهذا الخصوص على النصوص الإنجليزية. أما القيام بعمل بطاقات اقتباس من النصوص المكتوبة بالعربية فليس بالأمر اليسير؛ لصعوبة معالجة تلك النصوص وشُحّ مصادر المعرفة باللغة العربية.

هذا العمل يقترح طريقة للقيام بهذه المهمة بالنسبة للنصوص القصيرة بالعربية. وتستخدم الطريقة المقترحة موسوعة "ويكيبيديا" العربية كخلفية معرفية من أجل عمل بطاقات اقتباس مقترحة من نصوص قصيرة. ويستفاد من تحليل الدلالات الكامنة في الألفاظ في تحليل نصوص قصيرة من الموسوعة المذكورة وإيجاد فقراتٍ لها علاقة بالنصوص المدخَلة. بعدئذٍ يتم انتقاء البطاقات المتعلقة بالنصّ من العناوين والفئات الخاصة بتلك الفقرات ومن ثم ترتيبها وفق درجة علاقتها بالنصّ.

تم تقييم الطريقة المقترحة بناءً على التقديرات الممنوحة لها من الخبراء بتطبيق ذلك على (993) بطاقة. وأظهرت النتائج أن الطريقة المقترحة أحرزت معدّل متوسط دقة قدره (84.39%) ومتوسط رتبة عكسي قدره (96.53%) واشتملت الدراسة على مناقشة مستفيضة للنتائج التي توصلت إليها؛ لإلقاء الضوء على نقاط القوة والضعف للطريقة المقترحة.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 02, June 2020.

WEAVESIM: A SCALABLE AND REUSABLE CLOUD SIMULATION FRAMEWORK LEVERAGING ASPECT-ORIENTED PROGRAMMING

Anas M. R. AlSobeh[1], Sawsan AlShattnawi [1], Amin Jarrah[1] and Mahmoud M. Hammad[2]

(Received: 10-Feb.-2020, Revised: 16-Mar.-2020, Accepted: 4-Apr.-2020)

ABSTRACT

Cloud computing service-oriented simulation frameworks are very important tools for modeling and simulating the dynamic behavior of cloud-based software systems. However, the existing service-oriented simulation frameworks lack the ability to measure and control the rapidly changing (adaptive) requirements that span over many modules in cloud-based software systems, such as security, logging, monitoring, ...etc. To address these limitations, this paper presents an efficient framework for reducing the complexity of modeling and simulating the custom and dynamic behavior of cloud-based applications, called WeaveSim. WeaveSim utilizes the aspect-oriented programming (AOP) to encapsulate the complexity of developing the dynamic behavior of cloud-based applications by adding another abstract layer called Context-Aware Aspect Layer (CAAL). CAAL reduces the complexity of using CloudSim to simulate cloud-based software systems. Examples of cross-cutting concerns are data encryption, logging and monitoring. Since implementing a cross-cutting concern on a cloud-based simulator, such as CloudSim, requires modifications, from developers, to many core modules of that simulator. However, using WeaveSim, implementing cross-cutting concerns would be an easy task for developers, since they only need to reuse pre-defined joinpoints and pointcuts without modifying the underlying core modules of the simulator. We evaluated WeaveSim on an academically-scaled system. The results of our experimental evaluations show the benefit of WeaveSim in reducing the complexity of implementing cross-cutting concerns on cloud-based software systems. Hence, the reusability, scalability and maintainability of the cloud-based software systems are increased.

KEYWORDS

Model-based aspect-oriented programming, Cloud simulation, CloudSim, Dynamic modeling, Cloud architecture.

1. INTRODUCTION

Cloud computing provides easy access and reuse of shared resources anytime, everywhere. The appearance of cloud-based applications is often developed across multiple scattered units, some of which are called at run time. To achieve cloud collaboration services in cloud computing, cloud computing architecture must provide better-paired modules, such as cloud data access, monitoring, data compression, security and scheduling concerns [1]. These concerns about the cloud have increased the market value of many customers who want to use the cloud to achieve their organization's goals faster. This space is particularly complex, as these systems generally cost millions of dollars to develop and hundreds of thousands or more in annual cloud deployment costs. For example, security is a common secondary requirement, which is a comprehensive interest in cloud computing. The security application is required to interact with a set of scattered resources, contacts and nested context data objects stored in cloud instances.

By nature, a cloud-based application, or cloud app for short, is a complex and dynamic software system that consists of a set of contextual attributes and communicates with various services over the Internet. These attributes are scattered over different cloud levels: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) [2]. The required test of such cross-cutting concerns is very important for cloud users (e.g., service providers and consumers). Hence, the developer needs some crucial insight into how to implement these concerns, such as optimization of application efficiently. The implementation of cross-cutting concerns aims at improving the evolutionarily,

1. A. AlSobeh, S. AlShattnawi and A. Jarrah are with the Department of Computer Information Systems, Yarmouk University, Irbid, Jordan. Emails: anas.alsobeh@yu.edu.jo, Sawsankh@yu.edu.jo and amin.jarrah@yu.edu.jo

2. M. Hammad is with Software Engineering Department, Jordan University for Science and Technology (JUST), Irbid, Jordan. Email: m-hammad@just.edu.jo

evolvability, usability, understandability and efficiency of cloud apps.

This paper proposes a scheme that implements an efficient framework for reducing the complexity burden over cloud modeling and simulating the custom dynamic behavior of cloud-based solutions and applications. The proposed framework is structured into an abstract layer and aspects that are stored in the cloud by the simulator along with access to context-aware metadata. CloudSim simulator [3] is the most popular and extensible simulator that enables modeling, simulation and performance evaluation of emerging cloud computing applications [4]-[5] . CloudSim in its native state can be problematic to get up to use. Moreover, design choices and issues that were made and have evolved during its development have created significant issues in terms of its codebase as well as its efficiency and, with respect to certain aspects and details, its scalability and reusability. However, implementing and measuring the behavior of cloud apps against adaptive requirements are challenging tasks and are issues that have not been solved completely yet. Filho et al. [6] redesigned the CloudSim's core code components to increase its scalability and maintenance. In addition, they proposed features to enable dynamic monitoring behavior such as using listeners and performing dynamic operations such as arrival and destruction of virtual machines (VMs), horizontal and vertical VM scaling, fault injection and recovery, dynamic exchange of policies in runtime, …etc. However, this work did not solve the main contextual metadata complexity problem to offer scalability at runtime. In other words, it is not easy and clear to implement secondary requirements, since CloudSim does not provide clear insight for developers into implementing them. This dilemma prevents developers from evaluating the adaptive requirements of their cloud apps as a singular abstract module; i.e., separation of concern (SoC), such as monitoring, security, performance, cryptography, hive, map-reduce, throughput, …etc [1]. This highlights a general lack of sufficient care and accuracy in overreaching in terms of what value CloudSim would actually provide against the overall domain we target. Hence, value exists in creating overlay frameworks and/or abstraction layers so as to make CloudSim more accessible to its potential user community. Using Aspect-Oriented Programming (AOP) makes our work distinguished in that it does not require core code redesign or amendment and might be adaptable over any CloudSim architecture without key changes, viz. obliviousness process.

The significant concern about applying Aspect-Oriented Programming (AOP) into cloud application is dealing with an advanced access control distributed objects; for example, due to different issues such as modification in cloud behavior levels to access the shared resources. Using AOP provides a dynamic and flexible process to add probes for cloud-related cross-cutting concerns and to evaluate the system against related aspects; it encapsulates cross-cutting concerns in first high-level modules. AOP [7] is the appropriate approach for implementing cross-cutting concerns in separate modules in the CloudSim simulator dynamically. To add aspects into CloudSim, we have to extract relevant cloud-related contextual information, introduce helper characteristics and build utility functions, as a promising programming technique that promotes reusability and scalability of software systems through the separation of cross-cutting concerns [8].

The proposed simulation framework is a very abstract cloud collaborative scheme along with an efficient, expanded cloud application layer that leverages AOP to allow developers to efficiently implement and measure the dynamic and complex capabilities associated with cloud environments, viz. WeaveSim. It enables a developer to simulate the ability of the cloud-deployed cross-cutting concern to respond to time-domain variations in the volume and/or nature of the incoming workloads that it is servicing, e.g., which in turn is supported via elastic cloud services, …etc. WeaveSim deals with processing complex cloud- related cross-cutting concerns into separate first-class modules over the cloud. The key contributions of WeaveSim lie in refactoring CloudSim with:

- Supporting AOP-based application-level models for providing the cloud domain attributes,
- Processing cloud-based context-aware aspects using joinpoint-advice model, which's executed based on the creation of multiple VMs as well as a single VM function and
- Facilitating injection of cross-cutting concerns using an abstract pointcut-joinpoint model.

According to our conducted experiments, WeaveSim (1) demonstrated its ability to help developers evolve dynamic requirements of cloud apps with less complicated functionalities and (2) improved the reusability and scalability of cloud apps by creating better SoC efficiently.

The rest of the paper is organized as follows; various methods having been used in the literature as given in Section 2. The overall WeaveSim architecture is given in the methodology in Section 3. To evaluate the efficiency of WeaveSim, Section 4 provides an experimental case study in which we implemented a security cross-cutting concern scenario, then we compared the quality of WeaveSim with CloudSim in terms of scalablity factors: evolvability, functionality, usability, maintainability and efficiency in the results and discussion in Section 5. The conclusions with avenues of future work are depicted in Section 6.

2. RESEARCH WORK: A BRIEF REVIEW

2.1 Cloud Computing Simulation Frameworks

Simulators manage and control the infrastructure of the cloud hardware and software components. They provide information and key performance indicators for both cloud-based platform and applications. Those simulators vary in characteristics, result assessment, result validation metrics and symptoms of cross-cuttingness. In this section, we will introduce CloudSim simulator and the recent simulators built over CloudSim [9].

Kumar et al. [10] have provided a generalized and extensible simulator for modeling and efficient experimentation of cloud infrastructures and application services. It enables simulating the intrinsic distributed environment of cloud providers in a computer and provides a controlled environment that is easy to setup to test the performance of cloud applications. In their work, they formulated the CloudSim architecture to simulate different types of cloud environments (public, private, hybrid and multi-cloud environments) for key issues of cloud-based applications by creating cloudlet instances, which are submitted and processed by VMs deployed in the cloud [11]. CloudSim is built on top of the core engine of grid simulator, GridSim [12], and it is based on Java. It is an open source, event-driven extensible simulator that has diverse cloud features and capabilities and can be extended to include many plug-ins [13]. Due to its extendability, many authors proposed additional features and capabilities to the CloudSim, including [14]-[15]. The proposed CloudSim added many features, such as supporting modeling and simulation for large data center applications, the software architecture and simulation which are energy-aware computational resources, supporting both the network topology and message passing techniques to be compatible with a wide range of applications and supporting the ability to customize the polices for host resources to virtual machines. The existing proposed simulation models are not fully integrated into the requested design and implementation cycle for different applications. The code of some of them is complicated and very difficult to understand [13]-[14], which may prevent adding more functionality. In [15], the authors proposed a system that needs more time to manipulate such that each participant requests only partial information. The scalability of the proposed CloudSim is enhanced by combining increasing the volume of service request technique and deploying multiple instances of the service software. The proposed system maintains the quality of service in terms of average response time, where the scaling behavior is maintained over a long-time scale. Moreover, the proposed model extracts the information and constructs the initial simulation file in an efficient way, where the designer can automatically extract information into the targeted simulation model and run it remotely. This improved proposed system can be applied for many applications and decrease the challenges and drawbacks of the current techniques.

CloudSim has a layered architecture which is built on top of the GridSim [12] simulator. It consists of four layers [3]: (1) Simjava layer that implements the core functionalities required for higher-level simulation; (2) GridSim layer which supports high-level software components for modeling multiple grid infrastructure; (3) Management layer that manages the data centers, hosts, CPU units and VMs; (4) Application layer which allows users to write a code to configure the functionalities of a host, applications, VMs and broker scheduling policies.

Figure 1 shows the core components of the CloudSim, consisting of a base platform for simulation and research. The components form the infrastructure on which CloudSim is based and their relationships, which show the dependency arising from the interactions and overlap between these core components, as well as the oscillation resulting from the distribution of common objects between them [16] as follows: DataCenter component is a module that implements the core infrastructure level provided by the providers of cloud resources. DataCenterBroker is a class that models the relationship between end-users and service providers to allocate resources according to certain quality of service (QoS) requirements. Cloudlet contains a set of modules for cloud apps services deployed in data centers. VM is a module that runs different instances of VMs that are considered parts of the host. The host can instantiate several VMs and allocate the cores according to the internal scheduling policy, which is extended from the abstract component called VMScheduling. In [17] survey, the authors show that CloudSim emerges as a platform for simulation research and shows a lack of support for distributed implementation tools.

Despite CloudSim is written in Java, an object-oriented (OO) language, the framework barely relies on the intrinsic OO message-passing mechanism; i.e., the utilization of relationships between entities to perform method calls across such entities. CloudSim is an even-driven simulation framework that relies on custom message queue mechanism to enable communication between entities, such as Datacenters, Hosts, VMs, Brokers, …etc. The implemented mechanism transmits data inside events using the object raw type, instead of enabling type-safe message passing. Despite that, you can store anything inside an object variable, which is error-prone and usually leads to runtime exceptions, making tracing the root cause difficult.

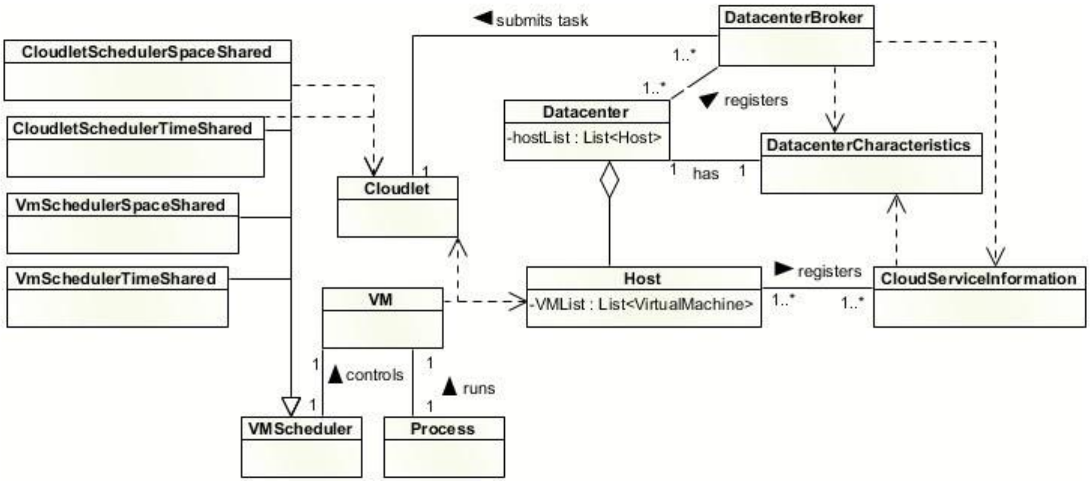


Figure 1. Key components of the CloudSim.

Despite all the aforementioned CloudSim features, the complexity of design scheme in master control and deep computation to control context attributes of cloud services makes it difficult to understand which type of context data each different event has to send. It allows the sender to transmit a data type different from that the receiver is expecting, leading to runtime exceptions if the receiver doesn't check whether or not the type of the received data is correct. Even if the receiver checks the data, there is a waste in processing to receive and discard an invalid event which shouldn't have been sent in the first place. This challenge prevents its scheme from being extensible and scalable effectively.

Wickremasinghe et al. [18] proposed CloudAnalyst simulator, which is derived from CloudSim, extended some capabilities and focused on evaluating performance and cost of large-scale Internet applications in a cloud environment. CloudAnalyst presents a significant challenge for a huge user workload. In attribute-based cloud applications, this issue is even more difficult, since each context data is conceivably scattered and tangled by multiple objects. This is why the current mechanism implemented in CloudSim is error-prone and not easy to understand and extend. This way, creating a data retrieval mechanism on top of this inappropriate CloudSim's message passing mechanism is questionable, mainly when one of your goals is to favor extensibility. Hence, it does not allow for separation of service abstractions and resources required by cloud applications. This presents a significant challenge for users who simulate cross-cutting concerns on the cloud. The next sub-section discusses how AOP copes with this challenge dynamically.

CloudSim4DWF is another simulator based on CloudSim to provide a new resources' provisioning policy for dynamic workflow applications. CloudSim4DWF added three modules to the CloudSim: (1) a graphical user interface (GUI) module that enables users to manage different types of VMs and to provide the inputs needed for simulation; (2) an event injection module aiming to trigger some events according to the dynamic workflow application; and (3) a resources' provisioning module for ensuring efficient resource provisioning for dynamic workflow. CloudSim4DWF injects some events that change the workflow during the runtime and adapts the actions to meet Quality of Service (QoS) requirements [19]. Dynamic CloudSim extends CloudSim to simulate instability and dynamic performance changes in virtual machines (VMs) during runtime. It determines whether a task succeeds or fails [20]. All of the presented simulation tools do not address cross-cutting concerns to improve cloud application evolvability. To address their limitations, we have designed and implemented WeaveSim framework that leverages AOP concepts to implement cross-cutting concerns efficiently.

Some simulators are presented in literature, such as GreenCloud [21] and CloudNetSim++ [22] for energy-aware cloud computing data centers. These simulators are designed to capture the energy consumed by data center components, such as servers, switches and links. GreenCloud is developed as an extension of a packet-level network simulator built on top of Ns2. The authors analyzed the network behavior of various data center network architectures over a designing data center simulator. They didn't consider the distributed data centers. CloudNetSim++ is built on the top of OMNeT++. It is designed to utilize the computing power of the data centers considering the distributed nature of data centers.

Other simulations rely primarily on CloudSim, as it is the basic infrastructure for other simulations (such as CloudSim plus, dynamic CloudSim, …etc.). These simulations suggest just reworking the code without showing any kind of updates with the same code limitations. So, our work is compared to become a competitor to CloudSim and is adapted to be more scalable in the future. Moreover, Byrne et al. in [23] reviewed 33 cloud-based tools. It identifies the emergence of CloudSim as a de facto base platform for simulation development and research. 18 of the platforms analyzed were derivatives or extensions of CloudSim. This is not surprising given the early mover advantage that CloudSim had, the eminence of the researchers involved and the quality and timeliness of the release of the simulator platform.

2.2 Cross-cutting Concerns in AOP Concepts

Cross-cutting concerns are aspects of a software system that span over many modules and affect the entire system. For example, monitoring, authentication, authorization and data encryption are crucial cross-cutting concerns that affect the entire cloud applications. The implementation of such concerns are either scattered (duplicated), tangled (significant dependencies between modules) or both.

In cloud application, the encryption concerns work as a service invoked when any message is sent or received [24]. Unfortunately, the above cross-cutting security concerns cannot be efficiently captured using the current implementation of CloudSim. To solve this problem, Filho et al. [6] proposed major restructuring and refactoring of the entire code base of the CloudSim, called CloudSim plus. They made a comprehensive re-engineering process to fix lots of existing issues in CloudSim. CloudSim plus was in fact focused on fixing lots of issues in CloudSim to promote extensibility. Changing core classes to create a customized framework could make it very difficult to incorporate improvements in new official versions of the base framework. Such a solution is costly and is not completely backward compatible with the cloud applications that have been built using CloudSim. In addition, there are major adjustments that are required to enable running a CloudSim's application in CloudSim plus. Due to those issues, they really re-structured CloudSim and provide a new general-purpose framework that can be easily extended (without forcing the researcher to change core classes). But, our framework intends to insert a layer over CloudSim instead of reforming the classes. Therefore, AOP concepts can solve this problem by implementing the aforementioned cross-cutting security concerns as separate aspects.

Even though AOP is a very powerful technique that can solve such a problem efficiently, no previous work in the literature has leveraged AOP for cloud simulation. WeaveSim is the first cloud simulation framework that leverages AOP to solve such a problem.

WeaveSim identifies a set of joinpoints in the CloudSim architecture and injects the code of each aspect module which changes the behavior of CloudSim at execution time without altering its core code. The added behavior; i.e., cross-cutting concern logic, is called advice [25]. These aspects encapsulate each concern in a special class, which alters the behavior of the base class by applying the advice at defined points on the original code. These points are called joinpoints; when a query matches at a point, this is called a pointcut [25]. Therefore, WeaveSim extends CloudSim by adding the necessary functions to support the separation of cross-cutting concerns across the core implementation of cloud apps in order to assess cloud services.

3. Overall Methodology: WeaveSim Model and Architecture

Changing the simulations' model of cloud applications changes the actual system's reusability or scalability. In particular, ensuring scalability is retained a critical concern within cloud-deployed software systems as cloud applications evolve and need to service expanding workload demands. This work addresses such issues and claims that it does. The proposed model presents a highly adaptive and collaborative scheme along with an efficient architecture in cloud computing based on CloudSim; i.e., WeaveSim. As described in Figure 2, WeaveSim shows the layered architectural design model of the WeaveSim framework. The architecture consists of six layers which can be applied to any data-driven cloud application that involves cloud-related cross-cutting concerns. The architecture includes: the CloudSim core layer, cloud resource layer, cloud service layer, user interface layer, cloud-aware aspect layer (CAAL) and application- level code layer. These layers are structured as a collection of loosely coupled services providing a solution for inter-service between cloud components in cloud-based applications.

The lower three layers (the CloudSim core, resource and user interface layers) are inherited from the CloudSim architecture. These layers provide infrastructures that facilitate communication, resources and services, respectively, needed to build cloud applications [4]. These layers depend on each other and are tightly coupled, which increases the complexity of implementing cloud-related cross-cutting concerns, e.g., monitoring, management, security, …etc.

The architecture involves an attributed-based context layer, where the context problem aspects from the analysis are embedded in the cloud-aware aspect layer (CAAL). This layer adds advice behaviors as structured activities in the CloudSim. The main reason for using the AOP is lying in making the objects' distribution and cross-cutting services encapsulated, extensible and accessible with less computational expenses with the layer that is executed into a machine.

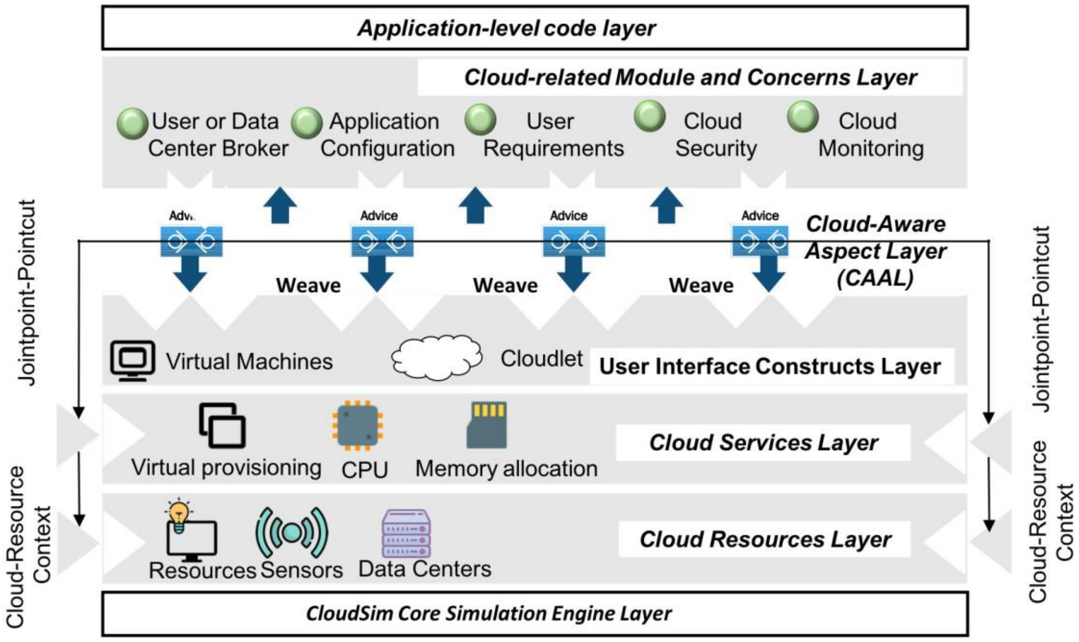


Figure 2. Layered architecture of WeaveSim.

CAAL includes an abstract library which offers services and interfaces to cloud apps. It integrates CloudSim APIs with contextual application-level components to make a cloud app more flexible and dynamic. It is modularized into small services that provide a well-defined functionality using well-accepted design principles. The layer offers the benefit of SoC to build scenarios that can handle the motioned challenge. In design, different packages have been used to make the design dynamic-oriented. Each package has aspects that are related to each cloud component. One of the interesting design decisions is that the system has an application-level code layer, the topmost layer, which contains aspects and methods for cloud objects. These aspects provide the ability to easier implement secondary functional requirements (i.e., cross-cutting concerns) necessary for cloud apps.

3.1 Cloud-Aware Aspect Layer (CAAL)

Cloud-Aware Aspect Layer is broken into a set of components (packages) communicated together. It is concerned with the aspects and mechanisms that are present in the cloud context data problem. It presents monitoring, WeaveSim and CloudService subsystems that monitor and simulate different cloud components in real time. They also make the CloudSim architecture and its functionalities available to developers as an abstract and reusable framework. CAAL is implemented pragmatically by extending the core functionalities of CloudSim. It allows users to select the cloud services and monitor the selected services based on user-defined characteristics. Technically, CAAL implements the cross-cutting concerns as aspects in the cloud architecture. It has been using different WeaveSim's joinpoints that are invoked/executed by aspects' pointcuts as shown in Figure 2. Each joinpoint has a crucial role in the simulation process. To simulate cloud-related cross-cutting concerns in a flexible manner, the CloudSim's meta-data components should be parameterized properly. CAAL extracts heterogeneous meta-data that supports comprehensive constructs of cloud-related cross-cutting concerns (e.g., managing cloud resource allocation and services). Such basic parameterization allows weaving of abstract cross-cutting concerns that might bind multiple cloud components together obliviously. Basically, CAAL implementation introduces a new abstraction level for aspects to overcome the dynamic weaving limitations on the cloud application code, recall Section 3, and it provides the adequate aspect representation in a woven structure based on context information, helper characteristics and utility function concepts.

The class diagram in Figure 3 depicts the structure of the cloud-based model of the dynamic weaving implementation within the CAAL layer. CAAL's structure is designed to embed the behavior of each advice with the weaving process as the structured event-based activities in the simulation process of the CloudSim. These events are being used in distribution service. They expose the context information about cloud components and are marked-up as general abstract classes and aspects to show how different values will be displayed at each joinpoint (e.g., BaseCloudServiceAspect, CloudServiceJP, CloudServiceJP). If new cloud services are added to the cloud application, those services' context information must be captured in such joinpoint which can improve the framework extensibility.

Through this layer, WeaveSim model leverages the intrinsic AOP communication mechanism to pass some context data around; i.e., method calls across a chain of objects. For instance, to pass a message to know in which data center a cloudlet is running, one has to simply cloudlet service joinpoint; i.e., CloudMessageService, which encapsulates message context data and provides a cheap message operation that returns immediately because it creates a SendEventJP and ReceiveEventJP. They cut-through simply call cloudlet.getVm().getHost().getDatacenter(), so there is no need to load a message to a queue, to be processed after a while and then return the response of the sender in an asynchronous manner.

This package is a library for services. It works as a monitoring component that performs low-level tracking such as instantiate, start, stop, resource allocation, management, scheduling, …etc. It automates the monitoring of a cross-cutting concern as a service. Moreover, its implementation determines the properties of various entities of cloud services, their communications and cloud applications to simulate the logical behavior of cross-cutting concerns. The issue with CloudSim is that any service that needs to be processed around goes into the CloudServiceRegistry, making the registry process really not expensive in large-scale simulations.

The package monitoring contains classes for both CloudSim components and cloud services. In Figure 3, CloudServiceJPTracer and BoundsimPoint are aspects extending the core components of the CloudSim by implementing specific new properties which are required to investigate and simulate cloud-related cross-cutting concerns. They cut-through the core components of the CloudSim to pull low-level contextual data characterizing the simulation of cloud-related concerns. Such context data includes services, network communication, VM, resources, device profile, location, calendar and time information.

Figure 4 shows that the CloudserviceJPTracer is an aspect object that cuts-through the CloudSim library components to connect CloudSim classes with defined distributed service (e.g., BaseCloudServiceAspect, ConfigureCloudlet, CloudSimBeginOnInitiator, CloudSimEndOn-Initiator, CloudMessageService, …etc.). It communicates with CloudSim's components via their joinpoints. It is responsible of exposing the spacious need for context data the effect of which manifests many separated parts of the cloud simulations. The collected service data is allocated in CloudsServiceRegistry. This repository uses the template parameter pattern to create a container for the advice methods for dynamically woven services.

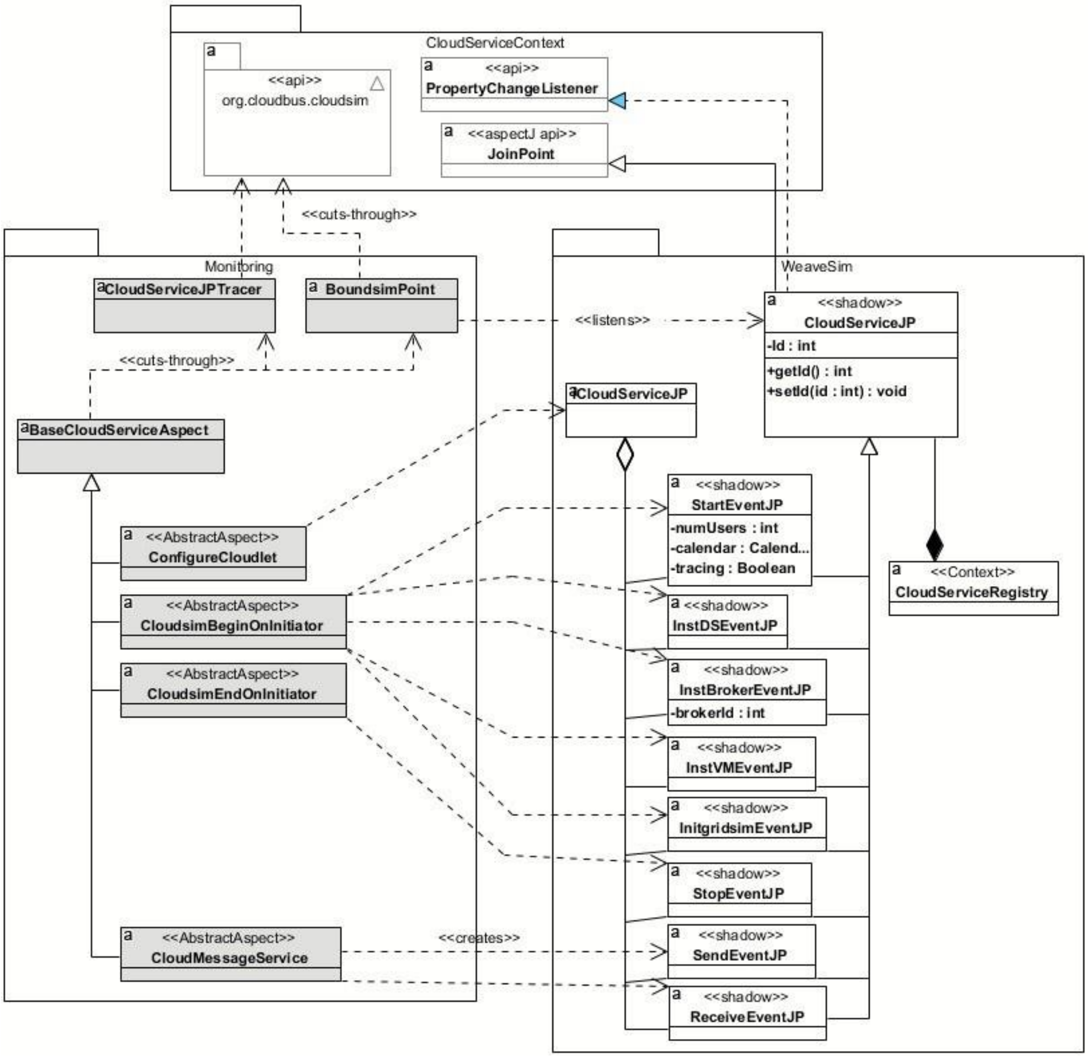


Figure 3. Structure of the cloud-based model of the dynamic weaving implementation within the CAAL layer.[1]

The BoundsimPoint is responsible for CloudSim's context which listens to parameter properties that hold a reference to an instance of CloudServiceJP, as shown in Figure 5.

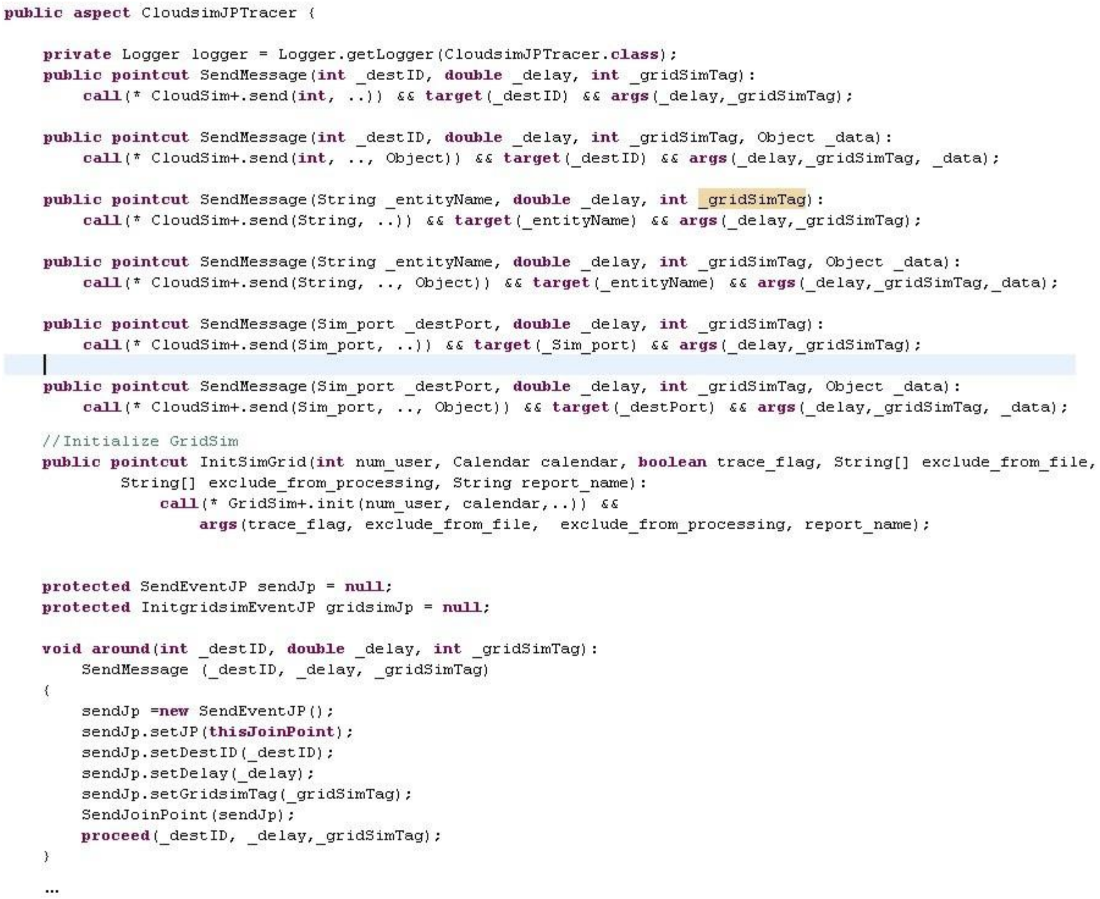
CloudServiceJP offers a direct mapping to the parameters of an advice for realizing joinpoint to add cross-cutting concerns related to a component-based cloud application. In addition, it supports the behavior of the aspects to work together, which can support the retrieval of the related context information and advice.

BaseCloudServiceAspect captures the auxiliary aspects which allow developers to handle interactions between different CloudSim components. It leads to the way in which different aspects cross-cut each other. All context parameters can be extended with additional properties to decide when and where crosscutting concerns are woven in order to perform the desired behavior.

The desired behavior is defined as a set of inherited subaspects, e.g., ConfigureCloudlet, CloudSimBeginOnInitiator, CloudSimEndOnInitiator and CloudMessageAspect. Such aspects can be woven before, after and during the calling and/or execution of joinpoints. In our approach, the binding of context parameters is represented in abstract pointcuts, which permits the reuse of pointcut signatures.

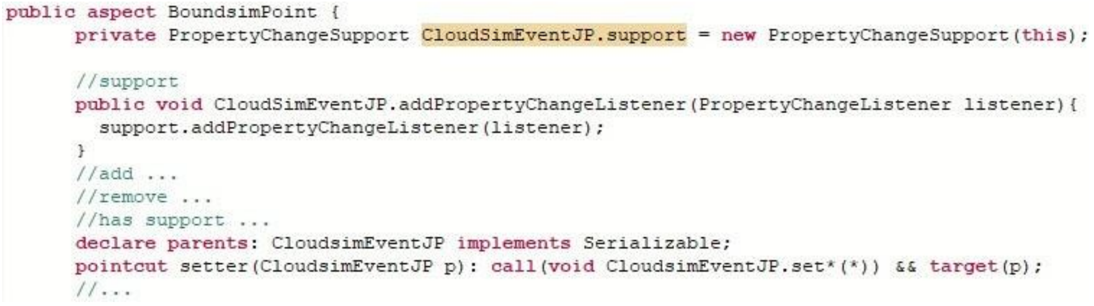
[1] Source code available on Github: https://github.com/aalsobeh .

We use explicit pointcut signatures which basically use generic constructs such as create, construct, initiate, start, stop, send, receive, …etc. The pointcuts pick out joinpoints and expose data from the execution context of joinpoints. The context provided by the selected joinpoints is validated regarding the general event handler class, the CloudServiceJP.



```
public aspect CloudsimJPTracer {

    private Logger logger = Logger.getLogger(CloudsimJPTracer.class);
    public pointcut SendMessage(int _destID, double _delay, int _gridSimTag):
        call(* CloudSim+.send(int, ..)) && target(_destID) && args(_delay,_gridSimTag);

    public pointcut SendMessage(int _destID, double _delay, int _gridSimTag, Object _data):
        call(* CloudSim+.send(int, .., Object)) && target(_destID) && args(_delay,_gridSimTag, _data);

    public pointcut SendMessage(String _entityName, double _delay, int _gridSimTag):
        call(* CloudSim+.send(String, ..)) && target(_entityName) && args(_delay,_gridSimTag);

    public pointcut SendMessage(String _entityName, double _delay, int _gridSimTag, Object _data):
        call(* CloudSim+.send(String, .., Object)) && target(_entityName) && args(_delay,_gridSimTag,_data);

    public pointcut SendMessage(Sim_port _destPort, double _delay, int _gridSimTag):
        call(* CloudSim+.send(Sim_port, ..)) && target(_Sim_port) && args(_delay,_gridSimTag);

    public pointcut SendMessage(Sim_port _destPort, double _delay, int _gridSimTag, Object _data):
        call(* CloudSim+.send(Sim_port, .., Object)) && target(_destPort) && args(_delay,_gridSimTag, _data);

    //Initialize GridSim
    public pointcut InitSimGrid(int num_user, Calendar calendar, boolean trace_flag, String[] exclude_from_file,
            String[] exclude_from_processing, String report_name):
                call(* GridSim+.init(num_user, calendar,..)) &&
                    args(trace_flag, exclude_from_file,  exclude_from_processing, report_name);


    protected SendEventJP sendJp = null;
    protected InitgridsimEventJP gridsimJp = null;

    void around(int _destID, double _delay, int _gridSimTag):
        SendMessage (_destID, _delay, _gridSimTag)
    {
        sendJp =new SendEventJP();
        sendJp.setJP(thisJoinPoint);
        sendJp.setDestID(_destID);
        sendJp.setDelay(_delay);
        sendJp.setGridsimTag(_gridSimTag);
        SendJoinPoint(sendJp);
        proceed(_destID, _delay,_gridSimTag);
    }

    …
```

Figure 4. A snippet of abstract CloudSimJPTracer module.



```
public aspect BoundsimPoint {
      private PropertyChangeSupport CloudSimEventJP.support = new PropertyChangeSupport(this);

      //support
      public void CloudSimEventJP.addPropertyChangeListener(PropertyChangeListener listener){
        support.addPropertyChangeListener(listener);
      }
      //add ...
      //remove ...
      //has support ...
      declare parents: CloudsimEventJP implements Serializable;
      pointcut setter(CloudsimEventJP p): call(void CloudsimEventJP.set*(*)) && target(p);
      //...
```

Figure 5. A snippet of BoundsimPoint code.

Finally, the CloudServiceJP allows for every CloudSim action to be validated according to the expected context parameters provided by advises and pointcuts explicitly. Indeed, events have to be extended to support new joinpoints without changing the validity of the CloudSim functionalities, as shown in Figure 3.

3.2 Joinpoint Shadows

Sub-aspects are structured for each CloudSim component to hold additional information for discharging and to be woven automatically. Depending on the level of a subaspect abstraction, the mapping refers to the execution of region range from simple joinpoints to complicated joinpoints. At any given point of time, only one abstract joinpoint can execute a region of a particular cross-cutting concern. This joinpoint contains a complete set of parameters for executing a cloud-related concern, where the advice associated with the aspect may be executed.

Shadows are places on the source code implemented as event handler classes to be responsible for the handling of joinpoint execution. The joinpoint context model is realized by defining a set of shadow points (reflective events) to which an advice can be bound (e.g., StartEventJP, InstDSEventJP, InstBrokerEvenJP, InstVMEventJP, InitgridsimEventJP, StopEventJP, SendEventJP and RecieveEventJP), as shown in Figure 3. In addition, with expanding the CloudServiceJP, the shadow classes provide utility functions related to encapsulate code constructs for each joinpoint simulator.

StartEventJP, InstDSEventJP, InstBrokerEvenJP, InstVMEventJP and InitgridsimEventJP encap- sulate the creation of the introduction region through initiating and starting of a simulation process. These classes added methods and properties to an advised cloud component to simplify tracking a target object at instantiating. StopEventJP encapsulates the creation of the region that spans after starting and before stopping a simulation. CloudletJP implements the creation of the entire region of a complete simulation service or task that spans before instantiating until after stopping. SendEventJP encapsulates the creation of the region that spans through establishing a message request containing a remote event to the end-user. The ReceiveEventJP encapsulates the creation of the region that spans through establishing a message response from an end-user. These shadows implement the entire region of the message-passing task that spans from before sending a request to after receiving a response.

In WeaveSim, each event class supports the shadow of those joinpoints and involves references to the CloudSim aspects' bindings to at least one pointcut that matches at a given event. In a nutshell, these references are used to pull the context information needed for weaving processes either before, after or around the joinpoint shadow.

4. EXPERIMENT: SIMULATION AND EVALUATION

Despite the successful practice in cloud-based systems, solutions are still unable to deal with the dynamic context changes. It is therefore necessary to provide context-aware innovative cloud services in which AOP-based services operate as a dynamic simulation service. To evaluate WeaveSim on real-world cloud apps, we have implemented the encryption service-oriented cross-cutting concern as a simulation service [26]-[27]. Figure 4 shows the quality of implementing cloud-related cross-cutting concern that begins from tracking states to the VM, broker and data center to ensure that the process of optimization becomes service-aware and well implemented. Adaptive WeaveSim based on the AOP increased the quality of the models regarding size, coupling, cohesion, obliviousness, complexity, response time separation of concerns and ease of change. Compared to Filho et al. [6], lots of issues in CloudSim are presented regarding these quality measurement features. One of the major issues in CloudSim and CloudSim plus is that some of these principles are not followed, such as correct inheritance and composition. Lots of subclasses in CloudSim just duplicate codes from the base class, what goes against inheritance. That increases code duplication, leading to software erosion and degrading maintainability. CloudSim 4.0 increased code duplication by 300%. That directly impacts all these quality measurement features.

WeaveSim ignores and improves some of these measurement features without adding additional code issues related to CloudSim and simply creates a separate layer over that original framework obliviously, as discussed in Section 5.

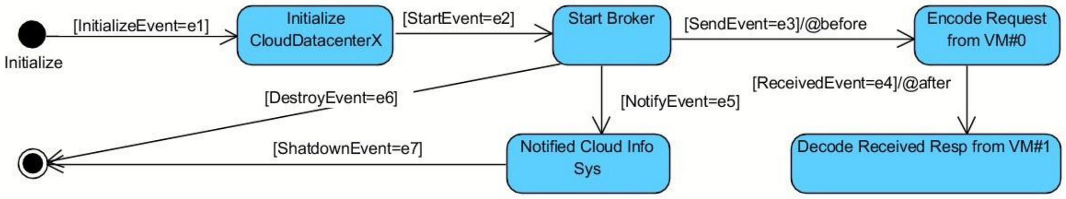


Figure 6. Example: the state machine causes weaving encryption solution.

Eventually, the simulation service using AOP-based service provides an integrated process simulation environment to optimize context-aware services for reduced time and resource consumption and increased quality and productivity. Hence, the extensibility, design stability, configurability, time behavior, resource behavior, code reducibility, understandability, complexity, learnability, reusability, changeability, modularity and scalability of platform-independent cross-cutting concerns could be improved in cloud application scenarios. In section 5, we examine such quality factors.

4.1 Cross-cutting Concern Case Study

In the proposed model, the central encryption cross-cutting concern is utilized to handle the communication security acting as a trusted application. Here, the encryption is significant to stay away from the security attacks amid the season of data modification. The access control policies are effectively taken care of in the proposed encryption feature. A real-world cloud-based weather system was used to conduct our experiments. This system indicates weather patterns and changes. The cloud application context data and encryption parameters are set up to the top application service level. The encryption parameters and collaboration services overhead increased the complexity of simulation by encryption and decryption message in real time. Therefore, the encryption process slows down the cloud's process significantly. Unfortunately, encryption is not well-structured into CloudSim components as a separate module; rather, it is intertwined into many CloudSim's components. This is the result of the interlocking process resulting from the implementation of such a concern. So, there is a need for a mechanism that allows developers to add such concern in the simulation process away from complexity and tangling caused by CloudSim.

Implementing such an aspect allows developers to evaluate the efficiency of various encryption algorithms. To show the benefit of WeaveSim, we have implemented the encryption service as a separate security aspect and compared that with a conventional implementation of the encryption service using CloudSim.

In our experiment, we expect the communication messages to be secured under attribute secret keys. The WeaveSim simulates encryption by injecting the secret keys on VMs at data centers. When a simulation process in WeaveSim starts, a Broker in the host requests, through a VM, a key to encrypt the Cloudlet message. The host is allocated in the datacenter and the VM encrypts the Cloudlet message data, creates a key, encapsulates the key in the Cloudlet response message and sends it back to the data center, as shown in Figure 6.

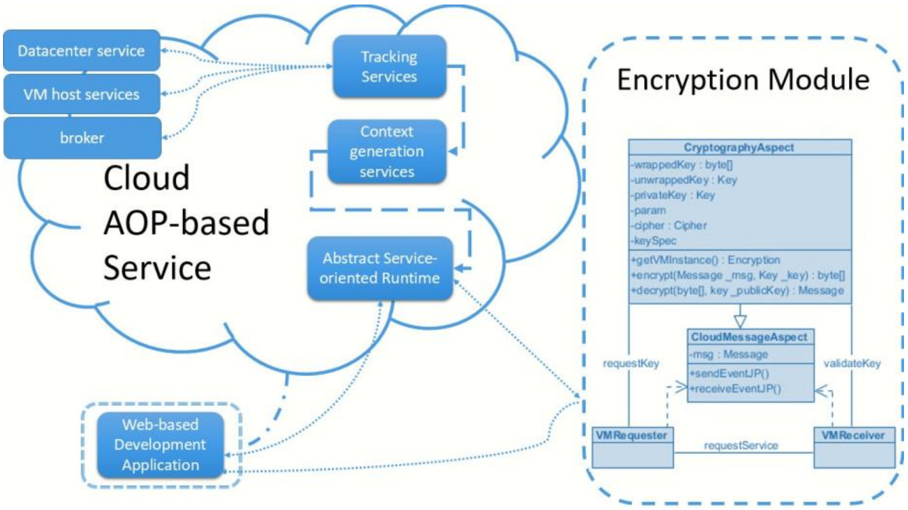


Figure 7. AOP-based model for the encryption service in the cloud weaving solution.

To exchange a secret key, Figure 6 shows a finite state machine for the encryption process. The behaviors of VM#0 and VM#1 are considerably simplified. The VM instance authenticates the requester instance, creates a key, encapsulates it in the response message and then sends it to the requester. The receiver instance also creates a new key, which sends the key request to the VM instance. The VM instance again authenticates the receiver instance, creates a key, encapsulates it in key Cloudlet message and sends it to the receiver instance. WeaveSim starts a VM process on Datacenter, which addresses the key requests from the requester instance before invoking the send method and after receiving a response from the receiver instance. In this approach, to ensure the integrity of the message data, we hash cloud message using the public key to obtain a new version with mapping all bits of the message data and offer an error-detection capability. Thus, encoding and decoding involve a combination of message data, a hash key and an encryption-decryption key.

To implement the encryption aspect in a way that would prevent the cloud's client from grabbing sensitive information, we extended CloudMessageAspect with CryptographyAspect.Cryptography-Aspect provides asymmetric key encryption constraints throughout the execution of the message; i.e., the send and/or the receive primitives. Figure 7 demonstrates the process of exchanging public-private key cryptography message codes, which apply to encrypt and decrypt advices at runtime, whereas AspectJ applies at sendEventJP and receiveEventJP, respectively.

5. Evaluation

We selected measurement features relative to the AOP and OO approaches, which effectively set up a strawman comparison of improvements on the basis of a selected set of measurements that are tuned to the software engineering approach. The measurements take a software code base, in CloudSim and WeaveSim and then overlays an ease-of-use methodology that an easier to evaluate system results.

We conducted a comparative experiment of the encryption cross-cutting concern implementation on both CloudSim and WeaveSim environments. Seven times of simulation runs were performed to measure such results. This is accomplished by comparing the result produced by the simulation weather application with data measured on the CloudSim and WeaveSim. The WeaveSim model has been desired with some specified degree of certainty (e.g., 95%) context parameters, as discussed below within acceptable confidence intervals for each hypothesis.

The evaluation process focuses on the quality principles of cloud applications. We mainly measure on these quality measurement features: (1) evolvability and functionality (Section 5.1) (2) usability and maintainability (Section 5.2) and (3) efficiency (Section 5.3). These quality measurement features are the most relevant measurement features for measuring the quality of cloud-based models [28]. These models are extended with further concepts, extra features and effective software quality measurements. To relate the results with the AOP concepts, we related the calculated results of each quality measurement feature with AOP concepts, such as aspect, pointcut, joinpoint and advice.

The specific set of concrete quality metrics to measure such attributes represents the number of AOP features, concerns and modules for cloud-related cross-cutting concerns. These metrics were adapted to measure the simulator application to be closer for the real-world implemented application. The simulation evaluation is completely isolated from the internal structure of the simulation itself. Our measurements measure the internal structure of the implemented applications. It's possible to implement a cloud application using a framework designed to run applications in a real cloud environment. These metrics will produce the same results in a simulated and actual cloud environment, because we measure high-level abstraction of applications for test results in the cloud.

5.1 Evolvability and Functionality Qualities

Evolvability is the ability of a cloud application to easily evolve in order to continue serving its users in a cost-effective manner. Since evolvability is a cross-cutting quality measurement feature of the system, it can also be considered as a non-functional requirement of the system. Thus, we treat evolvability as a functionality quality of the system [29]. To quantify these qualities, we considered a set of factors that cause the application to evolve. The identified factors are: extensibility, design stability, configurability, scalability and changeability. These factors can be measured by measuring how a system meets certain cross-cutting concerns' quality. Alongside, some of the related methods and advices might be adapted to measure the number of components (i.e., class or aspect) executed in response to a message received by a given feature (e.g., method and advice); these are triggered whenever a pointcut is matched. We define an application program as follows:

$$P = M_1(F_1, F_2, \dots, F_n), \dots, M_j(F_1, F_2, \dots, F_n) \quad (1)$$

where, P is a cloud app, Fi is a list of feature elements in a module Mj. M includes classes C, interfaces I, aspects A and cross-cutting concerns cc. F includes attribute/field/inter-type declaration field, advice adv, class shadow, method m, joinpoint and pointcut. These factors are measured by a set of respective measurement features.

- Degree Diffusion Pointcuts (DDPs): corresponding to the number of features defined in an aspect module. DDP measures the outgoing coupling connections (i.e., fanout) which helps the refactoring process and affects extensibility and scalability [30].

- In-Different Concerns (INCs): corresponding to the number of different concerns to which a module element is participating. INC is an AOP measurement feature for creating complex aspect code through extracting context information which is heavily dependent on the underlying code. This measurement feature affects the design stability and changeability factors [31].

- Feature Cross-cutting Degree (FCD): it corresponds to the number of modules that are crosscut by all elements of an advice in a concern and that are cross-cut by the inter-type declarations. It affects the application design stability and its configurability [32].

$$\begin{gathered} FCD = count(C \rightarrow m, \\ C \rightarrow constructors,\ C \rightarrow field, \\ C \rightarrow shadows(pointcuts(adv_{A(F_i)}))) \end{gathered} \tag{2}$$

Advice Crosscutting Degree (ACD): it corresponds to the number of classes that are exclusively crosscut by the method of advice in a concern. It affects the application design and stability [33]-[34].

$$\mathrm{ACD} = \mathrm{count}(\mathrm{Mj}(\mathrm{shadows}(\mathrm{pointcuts}(\mathrm{advA}(\mathrm{Fi}))))) \tag{3}$$

where A(Fi) represents the indices of aspect functions having been injected at the execution or call time.

- Program Homogeneity Quotient (PHQ): it corresponds to the summation of the homogeneity quotients of all features in a cloud app, divided by the number of features (FCD). It affects the application design stability and its configurability [35]-[36].

$$PHQ = \frac{\sum((\forall P,g.HQ(g,P)))}{FCD} \tag{4}$$

where,

$$HQ(Fi,P) = \frac{Count(ACD)}{FCD}$$

λ: is a computation based on aspect abstraction and application using AOP variable binding and execution.

- Class and Aspect Complexity due Number of Children (CACNoC): it corresponds to the number of inherited methods and advices of sub-classes and sub-aspects, respectively, of the parent class or aspect. It gives an idea regarding the effect of class and aspect on the overall design and implementation. The CACNoC value indicates the extensibility and the design stability, since inheritance is a form of code reuse [33].

$$\mathrm{CACNoC} = \sum_{\mathrm{j=0}}^{\mathrm{n}} \mathrm{Mj} + \sum_{\mathrm{i=0}}^{\mathrm{m}} \mathrm{Ai} \tag{5}$$

where, M_j and A_i are methods of class and aspect, respectively, at level *i,j.*

- Number of Methods (NoM) or Number of advices (NOA): it corresponds to the number of method and advice signatures in both classes and aspects in a cloud app, respectively. The complexity of (NoM) and (NoA) is obtained by counting the number of parameters in each operation and advice, assuming that an operation or advice with more parameters is a more complex than one with less parameters. They affect the code changeability level [37].

- Percentage of Advised Modules (ADM): it corresponds to the percentage of class and aspect modules that are interwoven, where a joinpoint shadow might be determined among all modules in the simulated cloud apps. It includes method-execution, method-call, constructor-call, constructor- execution, field-get and field-set pointcuts. It measures the extensibility and the scalability quality factors [38].

- Coupling on Advice Execution (CAE): it corresponds to the number of modules declaring fields that are accessed by a given module. It provides an overall estimate regarding the effects of aspects in other modules (classes or other aspects), in terms of how many modules an aspect affects and how many aspects affect a given module. It affects the capacity to extend and change a cloud app's components.
- Lack of Cohesion (LoC): it corresponds to the number of different methods and advices within a class and an aspect that refer to an invoked particular joinpoint. To reduce the possibility of errors during the development process, high cohesion value decreases complexity. LoC affects on the code design stability [39].
- Obliviousness (Obl): it corresponds to the number of inter-type declarations (ITDs) in the aspects and the number of times they are being used, which also includes the number of modules affected by pointcuts in a given aspect DDP. Moreover, it takes into account scattered aspects over cloud components INC. Obl indicates the tangling of aspects in a cloud app's components. It makes a simulated less reusable and less scalable cloud app.

$$\mathrm{Obl} = \mathrm{count}(\mathrm{IT\ D}) + \mathrm{count}(\mathrm{DDP\ }) + \mathrm{count}(\mathrm{IN\ C}) \qquad (6)$$

These aforementioned measurement features measure how much time and effort are required to evolve and maintain the functionality of a cloud app. The greater the value of each quality measurement feature, the more complex the program would be to evolve; i.e., the lower the better [16]. Figure 8 shows the values of these quality measurement features obtained to measure finer-grained constructs of cloud apps with different configurations on both CloudSim and WeaveSim.

The DDP, INC, FCD, ACD and PHQ measurement features measure the application tangling and scattering in AOP components [40]. Figure 8 clearly demonstrates that the number of changes required to evolve the functions of a cloud app is significantly reduced using WeaveSim in comparison to CloudSim. The results show that WeaveSim offers better encapsulating cross-cutting concerns with less scattering compared to CloudSim.

On the other hand, CACNoC, NOA, ADM and Obl provide additional insights into method and advice-level tangling of cloud applications inside aspect components [41]. As shown in Figure 8, WeaveSim's values are less than CloudSim's values, which indicates that the functions of cloud apps have lower coupling with the cross-cutting concerns. However, the figure shows that, using CloudSim, the concerns are tightly coupled with the cloud app's functions, increasing the complexity of evolving and reusing the functionalities.

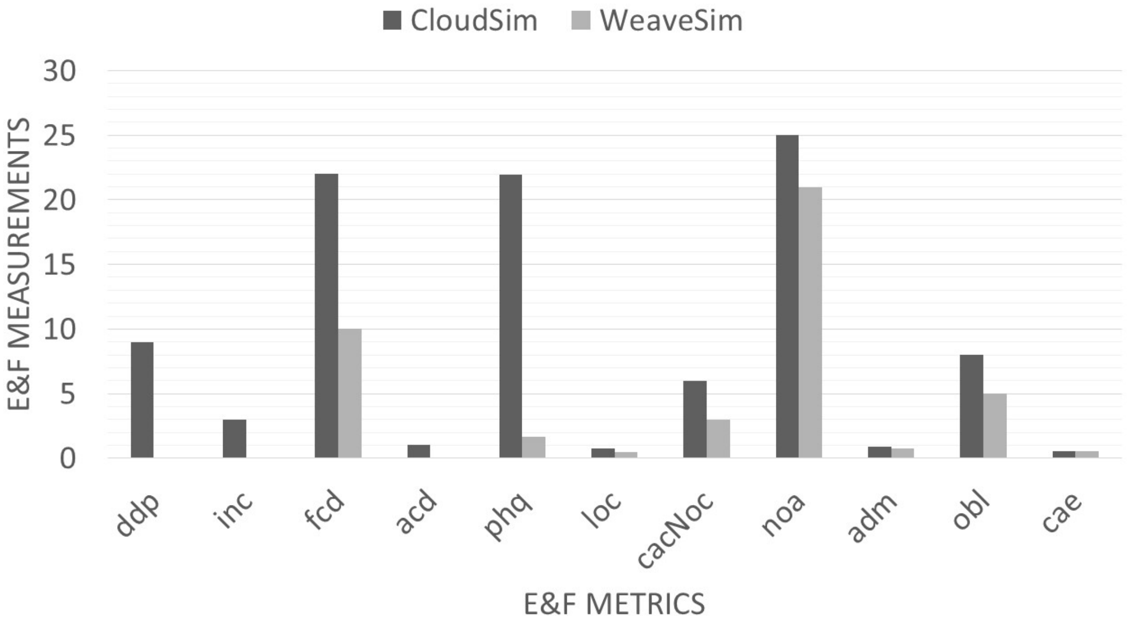


Figure 8. Evolvability and functionality measurement features for CloudSim and WeaveSim (the

lower the better).

CAE and LoC measurement features point out the complexity of cloud components in terms of advices, which should be tailored to the specific context information [30]. Figure 8 shows that CAE and LoC values for WeaveSim's application components have a higher coherence of a single cross-cutting concern logic compared to the tangled implementation in CloudSim.

Consequently, the presented result gives the confidence that in practice, the developer does not have a hard time trying to implement such concerns using WeaveSim compared to the extension built on top of CloudSim that was abandoned after a while. We can confidently conclude that WeaveSim provides a neater and cleaner set of pointcuts and joinpoints to cloud abstractions, which helps developers evolve the cross-cutting concerns with less complicated functionalities and implement some features on the top easily.

5.2 Usability and Maintainability Quality

Maintainability and usability measure the ability of a software product to be easily modified. Modifications may include corrections, improvements or adaptations of a software to adapt (change) to environments, requirements and functional specifications [42]. We measured these qualities with a set of factors, such as: code reducibility, understandability, complexity, learnability and reusability. These factors affect the SoC and ease of change, where SoC is an indication of concerns diffused in a cloud app and ease of change indicates the number of changes made to maintain a cross-cutting concern in the application [30].

- Line-of-Code (LoCC): it corresponds to the quantity of the executable source codes in terms of classes, interfaces and aspect elements in a correlative manner. The overall complexity of the simulated application will increase as the app's size grows. This affects the code complexity and understandability of the system [30]. LoCC can be computed as follows, where i is composed of several elements M1, ..., Mi of modules M.

$$\sum_{i=1}^{n} LoCC(Mi) \qquad (7)$$

- Classes, Interfaces and Aspects (CIA): it corresponds to the number of occurrences (NOOs) of classes, interfaces and aspects, as well as LoCC associated with each other. It points out whether aspects (as opposed to classes and interfaces) are a small or a large fraction of the modularization mechanisms used in a cloud app, then the implementation of cross-cutting concern will be a significant part or only a small part of the base code of the simulated application. It affects the code complexity and understandablity [43].
- Weighted Advice in Aspect (WAA): it corresponds to the number of adv and M's methods in a given aspect that indicates different weights to various advises with internal complexity. In other words, it is an indicator of how much effort is required to develop and maintain a particular cross-cutting concern. A high value denotes that the aspect is more complex and therefore harder to reuse and maintain, which may affect the code complexity and reusability.
- Code Replication Reduction (CRR): it corresponds to the reduction in the amount of LoCC when using homogeneous advises and inter-type declarations, rather than the amount of LoCC resulting from the use of traditional object-oriented approach. CRR is roughly the number of affected joinpoints, multiplied by the LoCC associated with them [35]. CRR affects the code reducibility, complexity and reusability [35].
- Degree of Focus (DoF): it corresponds to the variances of the dedication of a component to every concern with respect to the worse case. The average degree of focus gives an overall picture of how well concerns are separated in the program [43]. It affects the code complexity and learnability.
- Inter-type Declaration (ITD): it corresponds to the number of injected new data members into the core code to add states or behaviors to a particular class. A small number indicates less coupled modules increasing the code maintainability. It also affects the code reusability [30].

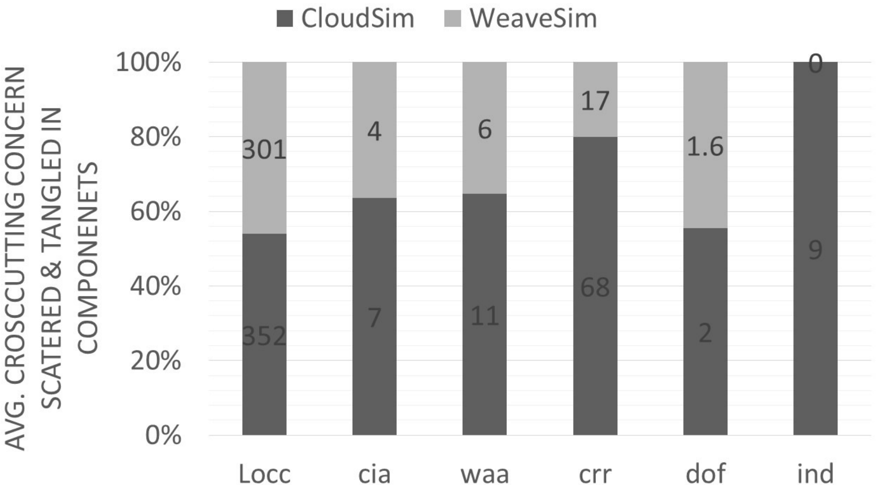


Figure 9. Usability and maintainability measurement features using CloudSim and WeaveSim (the higher the better).

An analytical evaluation for each of the aformentioned measurement features presents an overall sight of the size of the implemented cross-cutting concern in terms of modules. Figure 9 depicts the values of these quality measurement features for WeaveSim and CouldSim. LoCC and CIA are usually used as indicators of effort and productivity [44]. In the measurement of effort and productivity, Figure 9 clearly shows that the efforts for developing and maintaining a concern using WeaveSim are significantly less than those used to perform the same task with CloudSim. WeaveSim reduces the efforts, since it reduces the difficulty of redefining several core code behaviors in CloudSim. The flexibility of implementing such non-functional requirements in cloud apps is indicated by the high-level of SoC.

Figure 9 also shows that the measurements of WAA, CRR, DoF and ITD measurement features of the CloudSim app are higher than those of the WeaveSim app. The reason behind that is that CloudSim app contains tangled concern code with a few extension points which compromise ease of changes. In contrast, WeaveSim abstracts the aspects and provides context states that deal directly with cross-cutting concerns obliviously. Therefore, it is expected that the ability to modularize such concerns will improve the reusability and maintainability with a better SoC.

5.3 Efficiency Quality

The Response Time of Aspect (RFM) is a well-known factor that affects efficiency [45]. This measurement feature is used to measure the number of methods and advices executed in response to a message received by a given module triggered whenever a pointcut is matched. The results in terms of the change in response time for CloudSim and WeaveSim are shown in Figure 10.

To evaluate the performance of cloud apps, results were simulated on virtual instances configured with Window 10 (64-bit), 2.7, core i5 processor and 8GB RAM. The development environment was as follows: Java programming language (JDK 8), AspectJ to realize AOP concepts and Eclipse Oxygen IDE. As shown in Figure 10, we have conducted three rounds of execution where each round was run with different simulation parameters, as shown in Table 1. Table 1 shows the experimental design in terms of the involved userbases, user requests generated from each regional userbase and requests that are simultaneously processed by a virtual machine (VM).

As shown in Figure 10, CloudSim has a slightly higher efficient execution than WeaveSim, since CloudSim does not require any aspect. However, WeaveSim still provides an efficient weaving process for cross-cutting concerns. This Figure shows that WeaveSim extends the capabilities of CloudSim without degradation in efficiency. Based on our experimental results, WeaveSim provides reasonable efficient provisioning for large-scale cloud apps, allowing users to manage various types of VMs and provide the inputs needed for simulation without any knowledge of the core code of CloudSim.

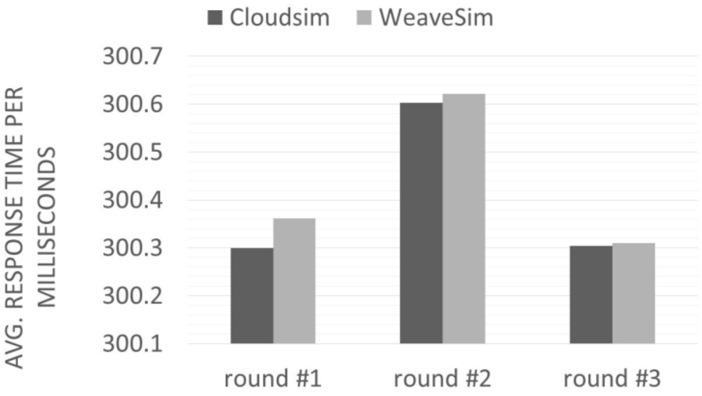


Figure 10. Overall average response time, in milliseconds, of encryption process using CloudSim and WeaveSim (the lower the better).

Table 1. The simulation experiment configuration.

Userbase	Region	TimeZone	PeakHours	Simultaneous
UB#1	0-N. America	GMT 6.00	08:00 - 10:00 pm	300,000
UB#2	Africa	GTM 4.00	10:00 - 14:00 pm	100,000
UB#3	Asia	GMT 2.00	10:00 - 16:00 pm	150,000

6. Conclusion

This paper delves into the factors that make cloud services largely scalable through bridging the gap between simulated cloud-based applications and real-time cloud applications. It introduces a novel cloud- based simulation framework called WeaveSim. WeaveSim extends CloudSim with AOP concepts to enable developers to simulate non-functional requirements for cloud computing applications easily. It extracts relevant contextual meta-data that can be applied to all cloud-related aspects. This framework effectively provides a high-level abstraction that encapsulates cross-cutting concerns in executable cloud structure in separate first-class aspects. It provides developers with a set of well-defined joinpoints and abstract pointcuts to pick the targeted cloud’s modules. In the future, we will conduct more experiments with different types of cross-cutting concerns, such as monitoring, transaction, quality of service, service level agreements and synchronization. It is sought to target the entire space inclusive of industry-scale cloud- deployed solutions. In addition, we will upgrade the base code of the AspectJ’s weaver which allows managing and handling the weaving process more efficiently.

References

[1] T. Grance and P. Mell, "The NIST Definition of Cloud Computing," NIST Special Publication, pp. 800–145, 2011.

[2] H. T. Dinh et al., "A Survey of Mobile Cloud Computing: Architecture, Applications and Approaches," Wireless Communications and Mobile Computing, vol. 13, no. 18, pp. 1587–1611, 2013.

[3] R. Buyya, R. Ranjan and R. N. Calheiros, "Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities," Proc. of IEEE International Conference on High Performance Computing & Simulation (HPCS’09), pp. 1–11, 2009.

[4] R. N. Calheiros et al., "CloudSim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services," arXiv preprint arXiv:0903.2525, 2009.

[5] H. Mei and X.-Z. Liu, "Internetware: An Emerging Software Paradigm for Internet Computing," Journal of Computer Science and Technology, vol. 26, no. 4, p. 588. DOI: 10.1007/s11390-011-1159-y, [Online], Available: http: //jcst.ict.ac.cn/EN/abstract/article_1768.shtml.

[6] M. C. Silva Filho et al., "CloudSim Plus: A Cloud Computing Simulation Framework Pursuing Software Engineering Principles for Improved Modularity, Extensibility and Correctness," IFIP/IEEE Symposium on Integrated Network and Service Management (IM), IEEE, pp. 400– 406, 2017.

[7] G. Kiczales et al. "Aspect-oriented Programming," Proc. of European Conference on Object-oriented Programming, Springer, pp. 220–242, 1997.

[8] R. S. Pressman, Software Engineering: A Practitioner's Approach, Palgrave Macmillan, 2005.

[9] S. K. Sood, "A Combined Approach to Ensure Data Security in Cloud Computing," Journal of Network and Computer Applications, vol.35, no. 6, pp. 1831–1838, 2012.

[10] R. Kumar and G. Sahoo, "Cloud Computing Simulation Using CloudSim," arXiv Preprint arXiv:1403.3253, 2014.

[11] P. Humane and J. N. Varshapriya, "Simulation of Cloud Infrastructure Using CloudSim Simulator: A Practical Approach for Researchers," Proc. of IEEE International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), pp. 207–211, 2015.

[12] R. Buyya and M. Murshed, "GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing," Concurrency and Computation: Practice and Experience, vol. 14, no. 13-15, pp. 1175–1220, 2002.

[13] R. Lakshminarayanan and R. Ramalingam. "Usage of Cloud Computing Simulators and Future Systems for Computational Research," arXiv preprint arXiv:1605.00085, 2016.

[14] F. Fakhfakh, H. Hadj Kacem and A. Hadj Kacem, "Simulation Tools for Cloud Computing: A Survey and Comparative Study," Proc. of the 16th IEEE International Conference on Computer and Information Science (ICIS), pp. 221–226, 2017.

[15] P. Kathiravelu and L. Veiga, "Concurrent and Distributed CloudSim Simulations," Proc. of the 22nd IEEE International Symposium on Modelling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), pp. 490–493, 2014.

[16] R. N Calheiros et al., "CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms," Software: Practice and Experience, vol. 41, no. 1, pp. 23–50, 2011.

[17] J. Byrne et al., "A Review of Cloud Computing Simulation Platforms and Related Environments," CLOSER, pp. 651–663, 2017.

[18] B. Wickremasinghe and R. Buyya, "CloudAnalyst: A CloudSim-based Tool for Modelling and Analysis of Large Scale Cloud Computing Environments," MEDC Project Report, vol. 22, no. 6, pp. 433–659, 2009.

[19] F. Fakhfakh, H. Hadj Kacem and A. Hadj Kacem, "CloudSim4DWf: A CloudSim Extension for Simulating Dynamic Workflows in a Cloud Environment," Proc. of the 15th IEEE International Conference on Software Engineering Research, Management and Applications (SERA), pp. 195–202, 2017.

[20] M. Bux and U. Leser, "Dynamic CloudSim: Simulating Heterogeneity in Computational Clouds," Future Generation Computer Systems, vol. 46, pp. 85–99, 2015.

[21] D. Kliazovich, P. Bouvry and S. U. Khan, "GreenCloud: A Packet-level Simulator of Energy-aware Cloud Computing Data Centers," The Journal of Supercomputing, vol. 62, no. 3, pp. 1263-1283, 2012.

[22] A. W.Malik, K. Bilal, K. Aziz, D. Kliazovich, N. Ghani, S. U. Khan and R. Buyya, "Cloudnetsim++: A Toolkit for Data Center Simulations in Omnet++," Proc. of the 11th IEEE Annual High Capacity Optical Networks and Emerging/Enabling Technologies (Photonics for Energy), pp. 104-108, 2014.

[23] J. Byrne, S. Svorobej, K. M. Giannoutakis, D. Tzovaras, P. J. Byrne, P. O. Östberg and T. Lynn, "A Review of Cloud Computing Simulation Platforms and Related Environments," CLOSER, pp. 651-663, April 2017.

[24] B. Fateh et al., "Secure Inverted Index Based Search over Encrypted Cloud Data with User Access Rights Management," Journal of Computer Science and Technology, vol. 34, no. 1, p. 133. DOI: 10.1007/s11390-019-1903-2,[Online],Available: http://jcst.ict.ac.cn/EN/abstract/article_2500, 2019.

[25] A. M. R. AlSobeh, R. Hammad and A.-K. Al-Tamimi, "A Modular Cloud-based Ontology Framework for Context-aware EHR Services," International Journal of Computer Applications in Technology, vol. 60, no .4, pp. 339–350, 2019.

[26] A. M. R. Alsobeh, A. A.-R. Magableh and E. M. AlSukhni, "Runtime Reusable Weaving Model for Cloud Services Using Aspect-Oriented Programming: The Security-related Aspect," International Journal of

Web Services Research (IJWSR), vol. 15, no. 1, pp. 71–88, 2018.

[27] A. A.-R. Magableh and A. M. R. AlSobeh, "Securing Software Development Stages Using Aspect-Orientation Concepts," International Journal of Software Engineering & Applications (IJSEA), vol. 9, no. 6, 2018.

[28] E. Kessler Piveta et al., "An Empirical Study of Aspect-oriented Metrics," Science of Computer Programming, vol. 78, no. 1, pp. 117–144, 2012.

[29] S. Ciraci and P. van den Broek, "Evolvability as a Quality Attribute of Software Architectures," The International ERCIM Workshop on Software Evolution, pp. 6–7, 2006.

[30] A. M. R. AlSobeh and S. W. Clyde, "Transaction-aware Aspects with TransJ: An Initial Empirical Study to Demonstrate Improvement in Reusability," ICSEA, p. 59, 2016.

[31] H. Ossher and P. Tarr. "Multi-dimensional Separation of Concerns and the Hyperspace Approach," Software Architectures and Component Technology, Springer, pp. 293–323, 2002.

[32] E. Figueiredo et al., "On the Maintainability of Aspect-oriented Software: A Concern-oriented Measurement Framework," Proc. of the 12th IEEE European Conference on Software Maintenance and Reengineering (CSMR), pp. 183–192, 2008.

[33] R. E. Lopez-Herrejon and S. Apel, "Measuring and Characterizing Cross-cutting in Aspect-based Programs: Basic Metrics and Case Studies," Proc. of International Conference on Fundamental Approaches to Software Engineering, Springer, pp. 423–437, 2007.

[34] E. Alemneh, "Current States of Aspect Oriented Programming Metrics," International Journal of Science and Research, vol. 3, no. 1, pp. 142–146, 2014.

[35] S. Apel, D. Batory and M. Rosenmüller, "On the Structure of Cross-cutting Concerns: Using Aspects or Collaborations," Proc. of GPCE Workshop on Aspect-Oriented Product Line Engineering (AOPLE), 2006.

[36] S. Garg, K. S. Kahlon and P. K. Bansal, "Testability Analysis of Aspect Oriented Software," International Journal of Computer Theory and Engineering, vol. 2, no. 1, pp. 119–124, 2010.

[37] G. Kiczales et al., "Getting Started with AspectJ," Communications of the ACM, vol. 44, no. 10, pp. 59–65, 2001.

[38] R. Akiladevi, "Aspect Oriented Refactoring for Software Maintenance," International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), vol. 2, pp. 79–84, 2013.

[39] L. Badri, M. Badri and F. Toure, "An Empirical Analysis of Lack of Cohesion Metrics for Predicting Testability of Classes," International Journal of Software Engineering and Its Applications, vol. 5, no. 2, pp. 69–85, 2011.

[40] A. Przybyłek, "An Empirical Study on the Impact of AspectJ on Software Evolvability," Empirical Software Engineering, vol. 23, no. 4, 2018.

[41] C. Driver and S. Clarke, "Distributed Systems Development: Can We Enhance Evolution by Using AspectJ?," Proc. of International Conference on Object-oriented Information Systems, Springer, pp. 368–382, 2003.

[42] I. Fleming, "Defining Software Quality Characteristics to Facilitate Software Quality Control and Software Process Improvement," Software Quality Assurance, Elsevier, pp. 47–61, 2016.

[43] F. E. Ritter, G. D. Baxter and E. F. Churchill, "User-centered Systems Design: A Brief History," Foundations for Designing User-centered Systems, Springer, pp. 33–54, 2014.

[44] K. Z. Ne Win, "Measuring and Evaluating Sustainability and Design Stability Software Qualities for Long-living Aspect-oriented Applications," Proc. of the 10th International Conference on ASEAN Community Knowledge Networks for the Economy, Society, Culture and Environmental Stability, Republic of the Union of Myanmar, pp. 8–12, 2014.

[45] M. Schalk, Response Time Measurement System and Method, US Patent 8,990,779, Mar. 2015.

ملخص البحث:

تُعدّ أُطر العمل الخاصة بالمحاكاة الموجّهة نحو الخدمات (service-oriented) في الحوسبة السحابية أدواتٍ مهمّةً جداً لنمذجة السلوك الديناميكي لأنظمة البرمجيات القائمة على الحوسبة السحابية ومحاكاته. ومع ذلك، فإن أُطر المحاكاة القائمة الموجّهة نحو الخدمات تحوزها القدرة على قياس المتطلبات سريعة التغير والتحكم بها؛ تلك المتطلبات التي تمتد لتشمل العديد من أنظمة البرمجيات القائمة على الحوسبة السحابية، مثل: الأمان، والتسجيل، والرّصد، ...الخ. ولمعالجة هذه المحدِّدات، تقدم هذه الورقة إطاراً فعالاً للحدّ من تعقيد النمذجة والمحاكاة فيما يتعلق بالسلوك الديناميكي للتطبيقات القائمة على الحوسبة السحابية. ويعرف الإطار المقترح بإسم (WeaveSim)، وهو يستفيد من البرمجة الموجّهة نحو المظاهر (AOP) للحدّ من التعقيد الذي يرتبط بتطوير السلوك الديناميكي للتطبيقات المرتكزة على السّحابة، وذلك عبر إضافة طبقة مجرّدة أخرى تسمى طبقة المظاهر الواعية للسياق (CAAL).

وتعمل الطبقة المضافة (CAAL) على التقليل من التعقيد المرتبط باستخدام إطار المحاكاة المعروف باسم (CloudSim) عند محاكاة أنظمة البرمجيات القائمة على الحوسبة السحابية. ومن الأمثلة على قضايا التعارض التي تتم معالجتها هنا: سرّية البيانات، وتسجيلها، ورصدها. ونظراً لأن تنفيذ مسألة من مسائل التعارض باستخدام نظام محاكاة قائم على السحابة (مثل CloudSim) يتطلب تعديلاتٍ من المطوِّرين على وحدات أساسية من ذلك النظام، فإن استخدام الإطار المقترح (WeaveSim) يسهّل مثل هذه المهمّة على المطوِّرين الذين لا يحتاجون في هذه الحالة سوى إلا إعادة استخدام نقط الوصل والقطع المعرّفة مسبقاً دون الحاجة الى إجراء تعديلاتٍ على الوحدات الأساسية المكوِّنة للمحاكي.

لقد تم تقييم الإطار المقترح (WeaveSim) للتحقق من أنه يحدّ من التعقيد المرتبط بتنفيذ مسائل التعارض، وجرى ذلك التقييم على نظام موزون أكاديمياً؛ إذ أثبتت النتائج فعالية الإطار المقترح في خفض التعقيد. وبذلك تم تحسين أنظمة البرمجيات القائمة على الحوسبة السحابية من حيث قابلية إعادة الاستخدام، وقابلية رفع الدرجة، وقابلية الاستمرار.

المجلة الأردنية للحاسوب وتكنولوجيا المعلومات (JJCIT) مجلة علمية عالمية متخصصة محكمة تنشر الأوراق البحثية الأصيلة عالية المستوى في جميع الجوانب والتقنيات المتعلقة بمجالات تكنولوجيا وهندسة الحاسوب والاتصالات وتكنولوجيا المعلومات.

تحتضن جامعة الأميرة سمية للتكنولوجيا (PSUT) المجلة الأردنية للحاسوب وتكنولوجيا المعلومات، وهي تصدر بدعم من صندوق دعم البحث العلمي في الأردن. وللباحثين الحق في قراءة كامل نصوص الأوراق البحثية المنشورة في المجلة وطباعتها وتوزيعها والبحث عنها وتنزيلها وتصويرها والوصول إليها. وتسمح المجلة بالنسخ من الأوراق المنشورة، لكن مع الإشارة إلى المصدر.

الأهداف والمجال

تهدف المجلة الأردنية للحاسوب وتكنولوجيا المعلومات (JJCIT) إلى نشر آخر التطورات في شكل أوراق بحثية أصيلة وبحوث مراجعة في جميع المجالات المتعلقة بالاتصالات وهندسة الحاسوب وتكنولوجيا المعلومات وجعلها متاحة للباحثين في شتى أرجاء العالم. وتركز المجلة على موضوعات تشمل على سبيل المثال لا الحصر: هندسة الحاسوب وشبكات الاتصالات وعلوم الحاسوب ونظم المعلومات وتكنولوجيا المعلومات وتطبيقاتها.

الفهرسة

المجلة الأردنية للحاسوب وتكنولوجيا المعلومات مفهرسة في كل من:

فريق دعم هيئة التحرير

ادخال البيانات وسكرتير هيئة التحرير	المحرر اللغوي
إياد الكوز	حيدر المومني

عنوان المجلة

الموقع الإلكتروني: www.jjcit.org
البريد الإلكتروني: jjcit@psut.edu.jo
العنوان: جامعة الاميرة سمية للتكنولوجيا، شارع خليل الساكت، الجبيهة، عمان، الأردن.
صندوق بريد: 1438 عمان 11941 الأردن
هاتف: +962-6-5359949
فاكس: +962-6-7295534

المجلة الأردنية للحاسوب و تكنولوجيا المعلومات

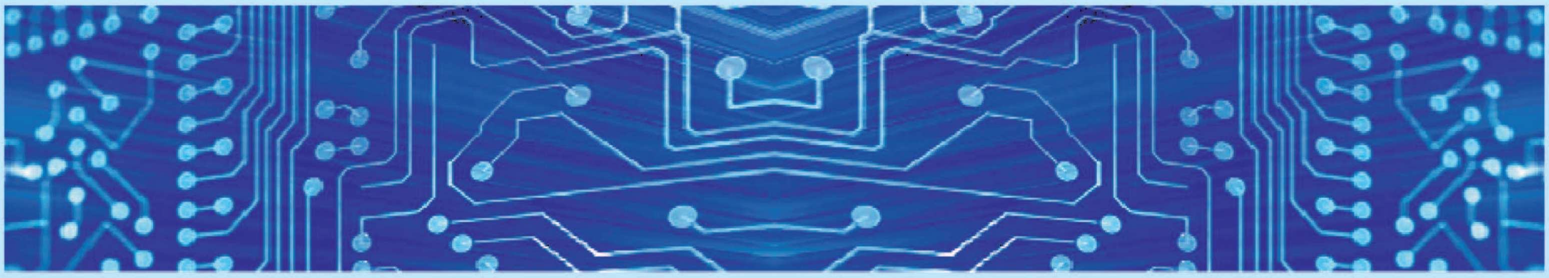
ISSN 2415 - 1076 (Online)
ISSN 2413 - 9351 (Print)

حزيران ٢٠٢٠ المجلد ٦ العدد ٢

JJCIT

الصفحات عنوان البحث

www.jjcit.org jjcit@psut.edu.jo



مجلة علمية عالمية متخصصة محكمة

تصدر بدعم من صندوق دعم البحث العلمي