The Computation of Assimilation of Arabic Language Phonemes

Ayad Tareq Imam
Faculty of Information Technology
Department of Software Engineering
Isra University
Amman, 11622, Jordan

Jehad Ahmad Alaraifi
School of Rehabilitation Sciences/Department of Hearing and Speech Sciences
The University of Jordan
Amman, 11941, Jordan

***Abstract*—The computational phonology is fairly a new science that deals with studying phonological rules under the computation point of view. Computational phonology is based on the phonological rules, which are the processes that are applied to phonemes to produce another phoneme under specific phonetic environment. A type of these phonological processes is the assimilation process, which its rules reform the involved phonemes regarding the place of articulation, the manner of articulation, and/or voicing. Thus, assimilation is considered as a consequence of phonological coarticulation. Arabic, like other natural languages, has systematic phonemes' changing rules. This paper aims to automate the assimilation rules of the Arabic language. Among several computational approaches that are used for automating phonological rules, this paper uses Artificial Neural Network (ANN) approach, and thus, contributes the using of ANN as a computational approach for automating the assimilation rules in the Arabic language. The designed ANN-based system of this paper has been defined and implemented by using MATLAB software, in which the results show the success of this approach and deliver an experience for later similar work.**

Keywords—Computational phonology; phonological rules; assimilation; phonological coarticulation; artificial neural networks; MATLAB

I. Introduction

Phonology is a branch of linguistics that studies the patterns' descriptions of speech sounds and the sound alternations in a language. The patterns are composed of abstract smallest units or sound types, which are called phonemes. Phonemes are the embedded abstract featured units that represent a meaning-distinguishing group of sounds in a language [1]. Each language has its own phonemes, and when phonology of a language is studied, it is actually addressing the phonemic inventory and how phonemes are organized and used [2].

A phonological representation is defined as the intellectual symbolizing of sounds and sounds' combinations that embrace words in a certain spoken language we have in our minds [3]. The physiological and physical features of sounds or speech are studied via a branch of phonology called Phonetics, which is divided into [4]:

- Articulatory Phonetics: depending on the production organs, each phoneme has it unique features that give the phoneme its distinctiveness among other phonemes. Sounds distinctive features result because of three main reasons, which are: the place of articulation (where the phoneme is produced), the manner of articulation (the way the phoneme is produced), and the voicing (whether there is a vibration of the vocal cords or not) [5]. Phonetics comes here in an Articulatory Form (ArtF), which specifying the distinctive features as criteria to classify phonemes [1].
- Acoustic Phonetics (the sound wave): concerns with discovering the physical properties of waveform like the mean squared amplitude, duration, fundamental frequency, and frequency spectrum. It also studies the relationship between these properties and the abstract linguistic concepts: phones, phrases, or utterances. Finally, acoustic phonetics investigates the relationship between waveform's physical properties and articulatory or auditory branches of phonetics [6].
- Auditory (Perceptual) Phonetics: The study of speech sounds from listener's point of view that focuses on the process of hearing and perception of a sound wave as much as the ears and brain do with the speech sounds reaching them. Phonetics here comes into an Auditory Form (AudF) [7].

The produced speech sounds, which made up words, are represented using alphabetic writing systems. The non-represented predictable phonological processes and the common of historical muddling of systems are two well-known shortcomings of alphabetic writing systems. The first one is an evolving standard, which is called International Phonetic Alphabet (IPA) and aims to transcribe the sounds of all languages of the human being. Advanced Research Projects Agency (ARPA) defined the second phonetic alphabet system, which is called ARPAbet, for American English using only ASCII symbols. Diacritic marks are also used to give an additional description of phonemes when they are produced as allophones. Aspiration (an additional amount of air follows the production of a sound), for example, is expressed using the diacritic mark [^{h}] as in the word tar which is transcribed as [t^{h}ar] [6], [8]. In all cases, there are three levels in which phonological representations are given, which are [9]:

- The acoustic level: pitch, loudness, and duration properties of signal form. These properties are used in this level of phonological representation of a spoken word. This level defines an Underlying Form (UF).

- The cognitive level: vowel phonemes and consonant phonemes classification is used in this level to describe the phonological representation of a spoken word. It is a type of Surface Form (SF).
- The linguistic level: the vocal tract and the ways that govern the production of speech sounds (like production's approach and articulation's place) are used to describe the phonological representation of a spoken word. This level encompasses morpheme level which connects the phonology to the syntax and the semantics in the lexicon.

For example, /t/ is considered as a voiceless sound because there is no vibration in the vocal folds while producing it, while /d/ is a voiced sound because the vocal folds vibrate while producing it. These two phonemes are the same with regard to their place of articulation (alveo-dental), and the manner of articulation (stops); actually, the voicing is what differentiates between them [9].

Researchers in [10]-[12] established Bidirectional Phonology and Phonetics (BiPhon) model, which is shown in Fig. 1. BiPhon model consists of five levels of representation and stored knowledge in a model of phonology and phonetics, which is a combination of phonological production model proposed by phonologists and comprehension and production models proposed by psycholinguists [13].

When the phonemes of a word are produced, they come in a form called allophones [1] [14], [6], which are the audible modification process applied to a phoneme. Using allophone instead of another allophone of the same phoneme results in different pronunciations of a word. For example, allophone /t/ phoneme in Eighth /eitθ/ is [t], in Writer /reiɾər/ is [ɾ], and in Tar /t^har/ is [t^h] [15].

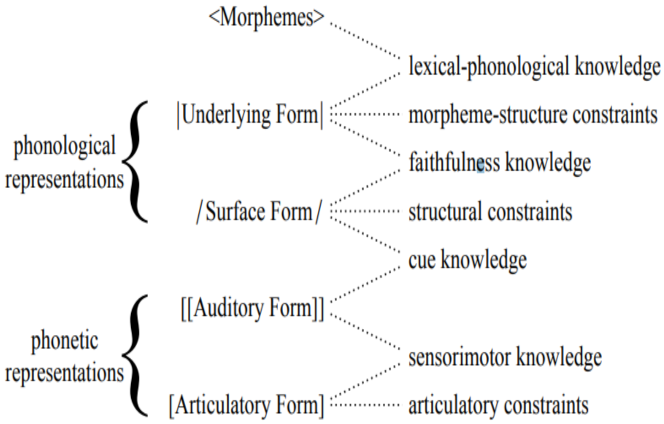


Fig. 1. Levels of representation and knowledge in a phonology and phonetics models [13].

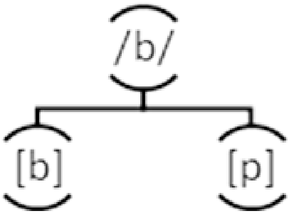


Fig. 2. Variations of the /b/ Arabic phoneme.

Phonetics' environment influences the way that the phoneme comes out. Phonord is the name of the outcome words of the production process, which includes the allophones, after applying the phonological rules. Phonemes are affected by the phonetic environment when they are produced in words, sentences, and connected speech. The permitted arrangements of sounds are called Phonotactics [16]. Coarticulation is a term used to describe the changes happened to the phoneme because of a specific phonetic environment, and assimilation is a consequence of coarticulation [4], [5]. For example, when the word /kabt/ is uttered, /b/ is affected by the following voiceless sound /t/ and it becomes voiceless and the speech outcome becomes [kapt] because of using the phonological rule: if voiced sound /b/ is followed by a voiceless sound, (such as /t/), then change /b/ to voiceless /p/ [17]. Fig. 2 shows the variations of the /b/ Arabic phoneme in the outcome of [kapt] Arabic word, which is termed as phonord. /p/ is an allophone for /b/ in Arabic language and has no effect on the meaning of the word. Note that /b/ and /p/ are not allophones in English since they are two different phonemes. For example, the English words /pa:t/, which means tapping, is different in meaning from the English word /ba:t/, which means wooden stick [18].

Basically, the substitution of a phoneme, with another one in the word, results changing the word's meaning. To illustrate that, consider the substitution the phoneme /d/ in the Arabic word /da:r/, which means house, with phoneme /s/ produces the word /sa:r/ (which means walked). Obviously, they are very different two words. The different meaning words that have one different phoneme in their utterances are called minimal pairs. Also, phonemes' order and place in the word are very important since they give the intended meaning. For example, the Arabic words /sa:r/ which means walked, and /ra:s/, which means head are two different words. They have the same phonemes but in a different order in each word, and consequently different meanings. When the person has an impaired phoneme pattern or impaired system of phonemes, he is considered to have a phonological disorder [16].

Phonological alternation is another type of changes happens to a phoneme, which is quite found largely in most natural languages. Phonological alternation is the gathering of multiple phonemes to produce morphemes. Diachronic sound change is the systematic phonological or morpho-phonological process in language, which is expressed by using phonological rule. Generally, the phonological rules are phonetic notations or distinctive features (or both) that describe sound-related operations and computations which are performed by human brain for generating or comprehending spoken language, which termed as generative phonology. The

[1] Phonemes are transcribed using phonetic transcription, in which symbols are put in virgules, while allophones are transcribed by phonemic transcription in which symbols are put in brackets [] [6].

following classifications of phonological rules come from the five phonological alternation forms, which are [19]:

- Assimilation: Change phoneme to allophone to make two adjust phonemes harmonic in their feature (elision).
- Dissimilation: Changes one of the sound's features to reduce its similarity to an adjacent sound in order to differentiate the two adjacent sounds.
- Insertion: Adding an additional sound between two adjacent sounds.
- Deletion: The omission of pronouncing a sound, for instance, a weak consonant or a stress-less syllable.
- Metathesis: Changing places of sounds within the same word.

The goal of this paper is to test the using of ANN for computing assimilation rule of Arabic phonemes. An overview over phonological and computational phonology has been made, focusing on the assimilation phonological rules in the Arabic language.

This paper is divided into the following main sections: The Arabic Language section (to describe its phonemes, phonemes alternation, and assimilation), Computational Phonology section (to describe computational models used in this field), section of Related Works and Approaches (to get benefit of previous work and experience, which is guiding the selection of a suggested approach), the Suggested Approach section (to handle the problem of computation of Arabic assimilation process), Results section (to make verification and validation of the proposed approach), Discussion section, and finally the Conclusions and Findings section (to discuss the suggested approach and its results).

II. Arabic Language

The Arabic language is a Semitic language spoken by 27 countries [20]. The main problem with the Arabic language is the range of assorted dialects, each of different phonology. However, it is worth to mention here that there is Modern Standard Arabic (MSA), which is used only in formal occasions and settings, such as literature and religious ceremonies, and Educated Spoken Arabic (ESA) spoken by educated people and it is not as formal as MSA [21]. The Modern Standard Arabic (MSA) consists of 26 consonants (b t d k ʒ q l m n f θ ð s Ṣ z ʃ x ɣ ḥ h r ς ŧ đ ∂ ʡ b), 2 semi-vowels (w j), and 6 vowels (ɪ i ə a ʊ u), according to (Sabir & Alsaeed, 2014). Arabic language has some phonemes that are not present in English, such as the emphatic sounds: /t/, /d/, /s/, /ð/. It also has pharyngeal sounds, such as / ç / and /ħ/, and uvular sounds, such as /q/ /χ/, and /ɤ/. Some phonemes, such as /q/ are not used in everyday colloquial Arabic (e.g., Jordanian Arabic), but they are used in Modern Standard Arabic (MSA), which is the formal shape of Arabic [21]. Arabic is a unique language in its sounds because they spread all over the tongue starting from the tip of the tongue and ending to the root of the tongue. It also has the glottal stop /ʔ/, which is considered as a phoneme. As for vowels, Arabic has three short vowels (harakat), which are /a, u, i/ (inflections) and three long vowels: /a:/, /u:/, /i:/ [22].

Like other natural languages, the Arabic language is governed by phonological alternations rules, which relate the phonemic level to the phonetic level and they show that the changes which occur to phonemes are not random; but are deliberate. Fig. 3 shows the phonological rule using distinctive features (description of phonemes using symbols), where (+) means that the feature is present and (–) means that the feature is absent. The condition of the phoneme is mentioned before the arrow and the changes are mentioned after the arrow. Fig. 3 explains symbols (color of explanations is the same color of symbol/s).

To illustrate how rule relates underlying representation to surface representation, consider the rule that says that voiced consonant becomes voiceless when it is followed by a voiceless sound. Let us look closely at the example of /b/ (which is a voiced consonant[2]) when it changes into /p/ (which is a voiceless consonant) in a specific environment (when it is followed by a voiceless sound). This example is illustrated in Fig. 4, in which the underlying representation /b/ (phonemic level) is the abstract form in one's mind, and /p/ (phonetic level) is considered as the surface representation that is produced by the speaker. What you have stored in mind is different from what you produce due to the ability of the brain to ease sound production (it is easier to produce voiceless sound proceeded by another voiceless sound rather than a voiced sound).

For example, we can see such this change practically in the Arabic word /kabt/ (which means "inhibition"). This word includes two consonants following each other: /bt/ and is mentally stored /kabt/. This word is going to be produced as [kapt] in some dialects. Note that the underlying representation of the word is /kabt/, and the surface representation of the word is [kapt]. In other words, what is in the mind is presented orally different (based on the phonetic environment).

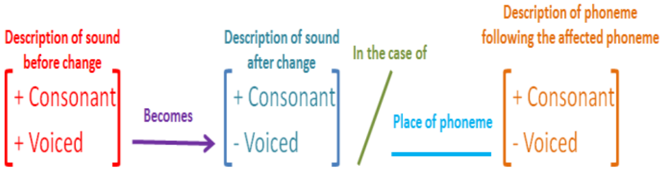

Description of sound before change
+ Consonant
+ Voiced
Becomes
Description of sound after change
+ Consonant
- Voiced
In the case of
/
Place of phoneme
Description of phoneme following the affected phoneme
+ Consonant
- Voiced

Fig. 3. Phonological rule using distinctive features.

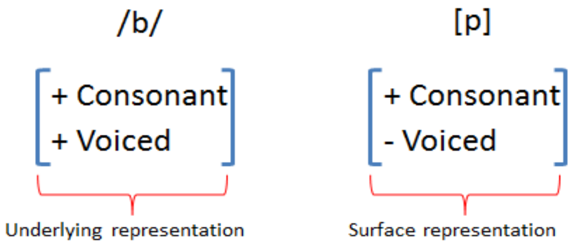


Fig. 4. Speaker's surface representation.

[2] Voiced sounds are produced with the vocal fold vibration while voiceless sounds are produced with no vocal fold's vibration.

Dissimilation is another phonological rule that is applied in some dialects where one sound from two identical sounds is changed to be different. The word /finʤa:n/ is produced as [finʤa:l], where /n/ is changed into /l/ [23]. Epenthesis -vowel insertion- is also another rule. When there are two consonants [3] following each other in a cluster, they are separated by a vowel. The example is employed in the word /kabt/, which is produced in some dialects as [kabit] [24]. Deletion of sounds is another way to ease speech. The word /zarqa:ʔ/ is produced as [zarqa:]. In this example, the glottal stop at the end of the word is deleted [25]. Another Arabic phonological rule is found in sound displacement in words (metathesis). For example, the word /malʕaqa/ “spoon” is produced as [maʕlaqa] [23]. This change of places might be due to the presence of two adjacent[4] sounds produced from the back of the mouth (/q/ and /ʕ/) and the goal of displacement is to separate the adjacent sounds from each other. Phonological rules are not applied on consonants only, but also on vowels. There are different variations of vowels in Arabic and some of them are presently based on dialects [26]. An example of phonological rules in Arabic vowels is the emphatic assimilation. As mentioned, there are three vowels in Arabic and three inflections (phonetically transcribed as short vowels). When the vowel is preceded or followed by an emphatic sound, it turns into a vowel that has some emphatic features. For example, /bata:ta/ (which means potato) is produced as [bɑtɑ:tɑ]. The underlying representations of the long vowel /a:/ and short vowel /a/ “inflection” in /bata:ta/ are not emphatic. However, the surface representations are [ɑ] and [ɑ:], both which are emphatic. The changes that occurred to the vowel are due to the effect of the emphatic sound beside it. However, certain authors applied the use phonetic transcription including vowel variations [27].

TABLE I. CLASSIFICATIONS OF ASSIMILATION

Criteria	Classifications
The amount of assimilation	Complete assimilation: the sound becomes exactly the same neighboring phoneme sound that affects it. Partial assimilation: the sound takes one neighboring sound features, which are the place of articulation, the manner of articulation, and / or the voicing.
The direction of assimilation	Progressive assimilation: when the previous sound affects the following sound Regressive assimilation: when the following sound affects the previous sound.
The distance between the sound that affects and the affected sound	Connected assimilation: If the two sounds follow each other Separate assimilation: If the two sounds are separated by sound /s/
The distinguishing features of sounds*	Place of articulation Manner of articulation Voicing

(* Some resources add two features: emphatic assimilation, and lip rounding [17].)

Assimilation is resulted from syntagmatic constraints which are even adjustments of articulatory productions to be acceptable perceptual demands of the listener like place assimilation [28], [29]. Assimilation is one of the phonological rules that occur when speaking. The main purpose of assimilation is to ease speech and make it more cohesive with less muscular effort [30]. Assimilation, in general, is classified in different ways since there are different criteria used for classification of Assimilation. Table I illustrates some of the classifications of Assimilation [31], [32].

TABLE II. ASSIMILATION RULES OF ARABIC LANGUAGE

Assimilation Rule	What happens ?	In what cases?	Word	Substitute
The identifier assimilation	/l/ changes to /ʃ, s, s, r, θ, n, d, d, t, t, ð, z/ or /ð/	/l/ is followed by:/ʃ, s, s, r, θ, n, d, d, t, t, ð, z/ or /ð, /.	/alsajja:rah/	[assajja:rah]
Deglottaliz-ation	/ʔ/ becomes a vowel	/ʔ/ is preceded by "harakat": /a/ "fathah" /u/ “dammah" /i/ "kasrah"	/faʔs/ /muʔmin/ /biʔr/	[fa:s] [mu:min] [bi:r]
Inflections assimilation in the pronoun	/u/ "dammah" becomes /i/ "kasrah"	/h/ of the pronoun is preceded by the /i/ "kasrah".	/ҫalaji:hum/	[ҫalaji:him]
Imalah	/a:/ becomes /e:/ [5]	/a:/ vowel is followed by a sound that has the inflection /i/	/sala:mih/	[sale:mih]
Lip rounding	/c/[6]. becomes /cʷ/	/c/ is followed by "dammah" /u/.	/kul/	[kʷul]
Labialization	/n/ becomes /m/	/n/ is followed by /b/ or /m/	/minma:/	[mimma:]
Emphatic assimilation	/s/ (non imphatic) becomes /s/ (emphatic)	/s/ is followed by an emphatic sound.	/sater/	[sater]
Voicing	/s/ becomes /z/ /t/ becomes /d/	/s/ or /t/ is preceded by a voiced sound.	/muhandis/ /ʔidtaҫa:/	[muhandiz] [ʔiddaҫa:]
Devoicing	/dʒ/ becomes /ʃ/ /d/ become /t/	/dʒ/ and /d/ are followed by /t/.	/ʔidʒtamaҫa/	[ʔiʃtamaҫa]

[3] Consonant cluster is a string of consonants without a vowel between them.

[4] Adjacent sounds are sounds that are produced from two closed places of articulation and the tongue needs to move very precisely to produce them.

[5] /e:/ is part of the inventory of some dialects, such as Lebanese [26].

[6] /c/ means any consonant.

Worth to mention here, the first three classes could be combined into one class. For example, the assimilation in the word [ʔiddaça:] (which is originally /ʔitdaça:/ is called complete, connected, and regressive assimilation [30]. There are several kinds of assimilation process in the Arabic language, in which the rules of these kinds are summarized in Table II [6], [17], [23], [26], [30], [33].

III. Computational Phonology

Computational phonology is a computer science field that concerns with developing a set of computational models for both the patterns and alternations of speech sounds. These computational models are to be used for [34]:

1) Phonological parsing using finite-state phonology and optimality theory computation approaches: This is the mapping of a surface phonological shape to its underlying phonological structure.

2) Syllabification is an opposed phonological function that is used for mapping a syllable structure to phones' sequences.

3) Computational morphology or computational orthography to differentiate it from text morphology.

Computational phonology is fairly a new area of the computational linguistic branch and is getting fast growth results from applying computational linguistics' theories, approaches, and technologies to phonology. Computational phonology describes computational models of phonological representation, computational models of sound alternations in a language defined by phonology, and using phonological models to map from surface phonological forms to underlying phonological representation. Thus, computational phonology is viewed as the application field of formal computational approaches that aim to handle the representation and processing sound patterns (phonological information) required when words and phrases are either built or recognized. As it implies, this field of science is cooperation between both phonological analysis, which describes the formal models and tests it against data, and computer science, which implements these formal models as computational models. Attain this goal will certainly extend the use of these formal and computational models to the computer as well as human beings [35]. But what are the tasks that computational phonology should handle? The tasks of computational phonology, which are illustrated in Fig. 5 are: [34]

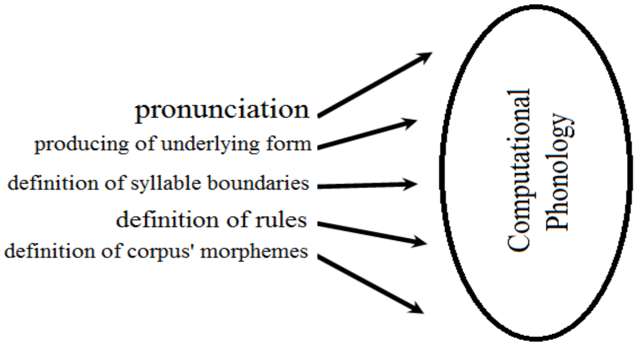

pronunciation
producing of underlying form
definition of syllable boundaries
definition of rules
definition of corpus' morphemes
Computational Phonology

Fig. 5. Tasks of computational phonology.

1) Producing surface form (pronunciation) of a given underlying form using phonological and morphophonological rules relate to that underlying form.

2) Producing of underlying form of a given surface form (pronunciation).

3) Definition of syllable boundaries of a given underlying (or surface) form.

4) Definition of rules that relate a given a database of underlying and surface forms.

5) Definition of morphemes exist in a given transcribed (or written) unannotated corpus.

This list of tasks imposes defining the rules' types required for modeling NL phonological systems, and the computational approaches required to implement these rules.

Due to the nature of the phonological problem, the approaches of Artificial Intelligence (AI), which is a field of computer science, are the most suitable ones that able to implement phonological rules.

Two tasks should be handled by computation phonology, which are phonological representation and sound alternations in language. The computational models of phonological representation aim to convert these environments to computational models its three levels: linguistic level, acoustic level, or cognitive level. The computational models of sound alternations in language are the computational models of sound alternations rules defined by phonology in language. These models are required for syntactically analysis and synthesis of a spoken word or statement.

Generally, the phonological parsing is more interested in using phonological models to map from surface phonological forms (linguistic) to underlying phonological representation (acoustic). A related kind of phonological parsing task to be handled by computational phonology is the syllabification, which is used for speech synthesis and defined as the assigning of syllable structure to sequences of phones [36]. Major models defined by computational phonology that used for phonological parsing task are finite-state phonology and optimality theory, which both use finite-state automaton. Certain research related with computation of assimilation used ANN. Both of finite-state automaton and ANN (also called Connectionist approach) are considered as the main methods used in computational phonology [37].

IV. Related Works and Approaches

Searching about related works leads us to find that there are four key approaches that had been followed to handle problems of computational phonology. All of these approaches belong to AI discipline of computer science. This is very normal since phonology topic is considered as an application that required AI techniques to deal with it. The works with computational phonology are of two types, either with phonological data or with the rules of phonology. Documentation, description, exploration, and analysis (sorting, searching, tabulating, defining, testing, and comparing) are some examples of previous work types. The following subparagraphs categories the previous work depending on the computational approach.

The rule-based approach is one of AI approaches used in computational phonology. A set of if-then statements forms rule-based system, which can be used to create a program that will deliver a solution or decision to a problem, much like a human expert. These systems may also be called an expert system and generally implemented using Prolog programming language [34]. Bobrow and Fraser's [38] Phonological Rule Tester is an earliest computational phonology research work developed to alleviate rule evaluation problem. We can mention here the work of [39], who proposed declarative phonology and ensuing work with a mathematical groundwork in first-order logic. J. Coleman [40] proposed phonetic interpretation relating to speech synthesis and Firthian Prosodic Analysis (FPA).

Finite State Transducers (FST) is another approach used in computational phonology. There are two types of FST, which are deterministic and non-deterministic. In Deterministic Finite State Transducer (DFST), only one state transition for every input state and it not allowed to move to a new state without consuming an input. NFST is a 7-tuple ($Q, \Gamma, \Sigma, \omega, \mu, q0, F$), where [34], [41]:

1) Q: a finite set called the states
2) Γ: a finite set called the alphabet
3) Σ: a finite set called the output alphabet
4) ω: $Q \times \Gamma \cup \{\varepsilon\} \rightarrow P(Q)$ is the transition function
5) : $Q \times \Gamma \cup \{\varepsilon\} \times Q \rightarrow \Sigma^*$ is the output function
6) $q0 \in Q$ is the start state
7) $F \subseteq Q$ is the set of accept states

Non-Deterministic Finite State Transducer (NFST) allows the normal transition state, transition without consuming input, and no-transition for an input state, which in the last case means no processing for the current input or the input is not accepted. DFST is a 7-tuple ($Q, \Gamma, \Sigma, \omega, \mu, q0, F$) where [34], [41]:

1) Q: a finite set called the states
2) Γ: a finite set called the alphabet
3) Σ: a finite set called the output alphabet
4) $\omega : Q \times \Gamma \rightarrow Q$ is the transition function
5) $\mu : Q \times \Gamma \rightarrow \Sigma$ is the output function
6) $q0 \in Q$ is the start state
7) $F \subseteq Q$ is the set of accept states

Example for this approach of computation the phonology can be seen in the work of Kaplan and Kay, who proposed the using of Finite State Transducers (FST) to implement the rules of generative phonology as a computerized system in the early of 1980s. Since that time, FST was a method for many research phonological works. The role of FST can be understood as computing of relation between two sets [42]. A type of weighted automata called Markov models had been used also by many researchers in speech recognition, and other related applications, which used phonetically annotated corpora (TIMIT for example) as training data [43].

Among the wide range of different application areas, the use of ANN in computational linguistics has been proven through several developed applications. ANN becomes popular processing approach for phonologically based applications. The abilities of modeling gradient behavior and training (self-learning) of ANN were motivations of using this approach. The self-learning is achieved via using a training database, which is to be observed and used to update the weights and biases parameters to reach a good classification ability of the ANN. The lower error in the classification during training phase results better network's architecture. ANN's architecture encompasses the number of layers, the number of neurons in each layer, and the selected input and output processing functions [44].

ANN can be seen in many phonological different applications, in which thus the inputs and output will vary accordingly. M. Gasser [45] used Recurrent Neural Network (RNN) to recognize syllables and to repair ill-formed syllables. Imam et al [46] used Feed Forward Neural Networks (FFNN) for recognizing distorted speech. There are so many other examples that use different types and architectures of ANN in the different computational phonology based applications.

Optimality Theory (OT) is a finite-state model that considers a finite upper bound on the number of violations and used to solve the problems of phonology. OT was firstly proposed in 1993 by Alan Prince and Paul Smolensky [47]. While phonology was the main area that most OT has been applied and associated with, OT has been applied and used also in other subfields of linguistics like syntax and semantics. OT can be used to explain variation among world's languages. In OT, universal tendencies, which are called constraints, are to be formalized in abstract form instead of defining new languages' rules using the observations' set of theoretical phonological rules. However, there are two things to consider, firstly constraints conflict each other because they can be false from time to time, and secondly languages differ in both: the values held by constraints and the ranking of constraints, in which this ranking is used to grade and thus make more accurate selection for possible pronunciations (that is the outputs) result from certain input [48]. OT consists of three basic portions, which are: Generate (GEN) that generates a list of potential outputs from certain input, Constraints (CON) that are the rules used to select an alternative from defined possible outputs, and finally, Evaluate (EVAL) that aims to pick up the optimal candidate using the defined CON which is the output [34]. Machine Learning is an interesting approach that used also in computational phonology. Given certain domain data accompanied with other potential information, these systems are able to automatically develop a computational model for these data. There are two learning approaches. First one is the supervised algorithms, which uses input data engaged with its correct answers to induce generalization model to be employed with further data. The second one is the unsupervised algorithms that use data and learning biases [49].

V. Suggested Approach

Following the standard steps for computation, computational phonology addresses assimilation phonological rules in three steps: Input, Processing Rules, and Outcome (output). To illustrate this, Fig. 6 shows the application of phonological rules on /kabt/ (the formerly mentioned example).

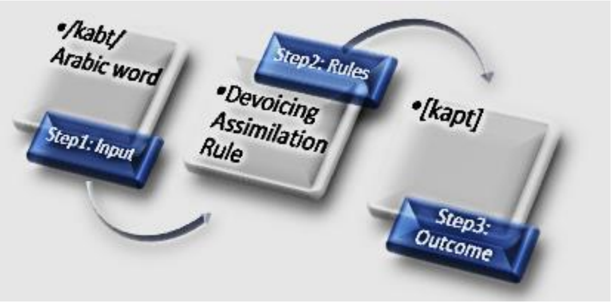


Fig. 6. Phonological rules applied to the word /kabt/.

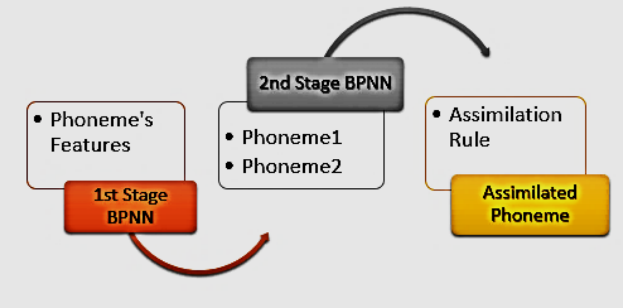


Fig. 7. Suggested approach of computation the assimilation rules of Arabic phonemes.

The example in Fig. 6 is used to define an approach for identifying and applying an assimilation rule considering an input phoneme. The steps followed for designing ANN are Data set collection, Creating, Configuring, and initializing of weights and biases of Network, Training of the Network, and the Using of the Network. Note that two ANNs, denoted as 1st stage BPNN[7] and 2nd stage BPNN, are going to be used in this research work. The function of the 1st stage BPNN is to recognize a phoneme. The function of the 2nd stage BPNN is to select the assimilation rule to be performed. This approach is illustrated in Fig. 7.

The assimilation process, in this work, is applied on phonemes of the Arabic language. The phonemes as data are recorded as signals and saved on files to be used as input to the 1st stage BPNN. Due to the huge size of the phoneme's signal, it is common that the phoneme's signal is going to be converted to a sort of representation that used the phoneme's features instead of phoneme's signal. The process of extracting phoneme's features is considered as a part of a pre-processing step that prepares the input data of the 1st stage BPNN as a pattern. The corpus used in this work is the MSA, which is illustrated earlier.

Different spectral analysis techniques are defined like Cepstral Analysis, Mel-scale Frequency Cepstral Coefficients (MFCC) Analysis, Linear Predictive Code (LPC), Perceptually Based Linear Predictive (PLP) Analysis, and Critical Band Filter Bank Analysis. In this work, LPC technique is used to represent the features of Arabic phoneme. According to LPC technique, the signal is composed of multiple parts. A mathematical representation of the signal is illustrated in (1) [50]:

$$s(n) = \alpha_1 s(n-1) + \alpha_2 s(n-2) + \ldots + \alpha_x s(n-x) \qquad (1)$$

Where:

s(n): the sample of speech at time n,

α_1, α_2 and α_x: constants over the frame of speech analysis

x: LPC's order.

As it is recommended by previous works, we define the number of LPC's coefficients to be 16 considering minimizing the error possibly appeared between the original signal and the one represented by LPC's coefficients. MATLAB's library of functions contains LPC function that determines the coefficients of a forward linear predictor as a function of time [51]. Algorithm for developing training pattern is:

1) Read from phoneme_database, the first Arabic phoneme file using.
2) Extract phoneme's features using LPC function.
3) Save the features in a vector.
4) Normalize the pattern.
5) Repeat 1- 4 for the second Arabic phoneme.

The function of 1st stage BPNN is to recognize Arabic phoneme by using its features. As illustrated in Fig. 8, the input layer of the 1st BPNN is of size 16-same as the size of the input pattern that is the LPC's coefficients. The size of the output layer is 8, the number of the phonemes that could be affected by the assimilation rules (see Table II). The number of the hidden layers and the size of each hidden layer are engaged to trial and error principle performed during the training stage.

The size of the input layer of the 2nd sage BPNN is two nodes (neurons) that indicate a phoneme and its neighbor one. Their values are of integer type, represent the codes of the input phonemes, each one is ranging from 0 to 8. The size of the output layer is 9 nodes (neurons), each one represents an assimilation rule of Arabic phoneme as listed in Table II. Due to the limited number of the input data and output targets, there was no difficulty to get a good mapping of the input phonemes to their corresponding output assimilation rule. It means that was no tries to get an identical number and size of the hidden layer. Fig. 9 illustrates the architecture used to develop the 2nd stage BPNN.

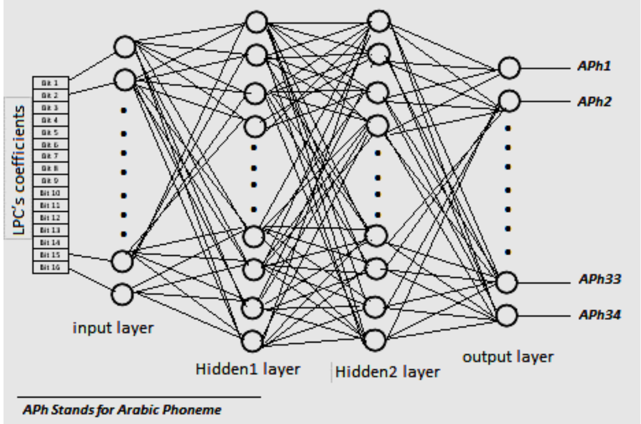


Fig. 8. Possible architecture of the 1st Stage BPNN.

[7] BPNN stands for Backpropagation Neural Network.

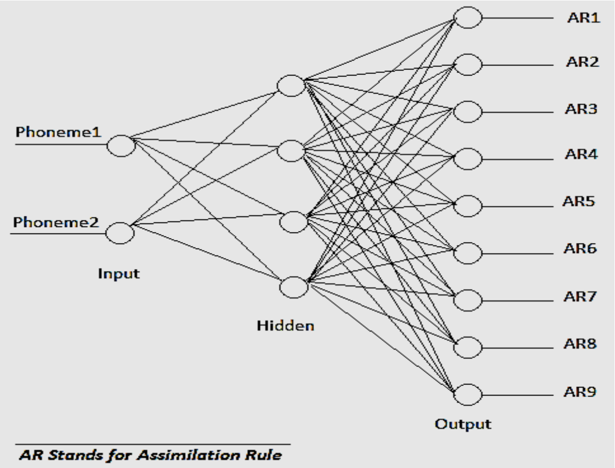


Fig. 9. Architecture of the 2nd Stage BPNN.

There is an intermediate space between 1st BPNN and 2nd BPNN. This space is a pattern of two cells that are filled by a process, which takes the outputs of 1st BPNN and fills this pattern with them. Thus, the intermediate pattern represents the codes of Arabic phoneme1 and Arabic Phoneme2 that are under Assimilation process and it is used as input to the 2nd BPNN. Several BPNN of different architectures were developed for each stage, where each stage is trained separately. The MATLAB's commands used to create a neural network are [51]:

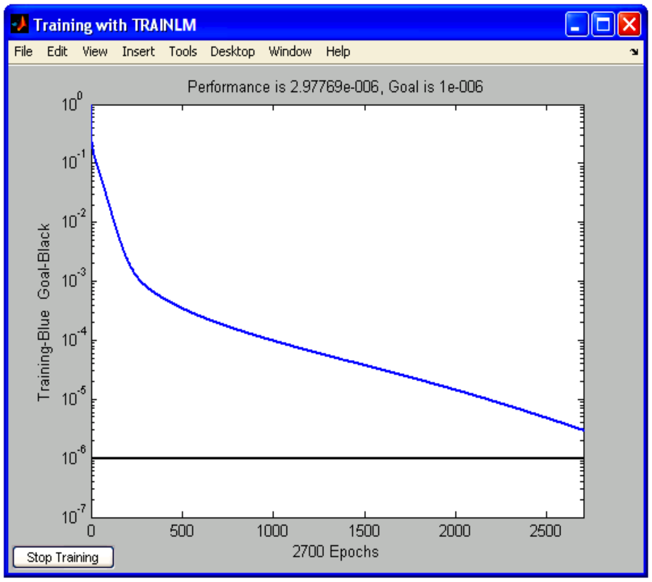
Mynet= network InputS,LayerS,biasC,inputC,layerC,outputC)

The next step is the training stage that adjusts the weights of the connections among the neurons in the network. The supervised learning algorithm is the one used to achieve this task. The epoch of learning is the term used to indicate the performing of training procedure once. As many epochs performed as more generalization ability of classification gained by the ANN, which makes it able to recall these pattern-categories, and hence leads to correctly classification of any unknown/untrained input pattern. In our work, the number of samples of Arabic phoneme, which was between 1200-10000 determines the number of epochs. We set the stop-training condition to be either reaching error of 1x 10-6 or completing the specified epochs. MATLAB's command used to perform training is [51]:

Mynet = train (Mynet, InputPattern, TargetPattern);

Following the step of training each network's architecture under development, a measuring of the accuracy of the suggested BPNN is performed by using 10 samples of each phoneme (possibly affected by assimilation rules), which were randomly selected from the phoneme_database. Note that training data is used to train all possible architecture of BPNN as well as the using of testing data. This is significant criteria to assure the trusty of the results yielded.

Proper grouping of transfer functions, learning and training play a critical role in the success of any designed ANN. In this work, the BPNN of the two stages had been trained using back propagation learning algorithm of supervised training strategy with the following parameters [51]:



Performance is 2.97769e-006, Goal is 1e-006

Fig. 10. MATLAB's training phase of a 1st Stage BPNN.

- Trainlm function for updating weights and bias values according to Levenberg-Marquardt optimization.
- Gradient descent method (GDM) with momentum weight and bias learning function, namely learngdm, for reducing the mean squared error between ANN's output and the actual error rate.
- The mean squared errors (mse) for measuring the network's performance together with the rate of convergence and the number of epochs taken to converge the network.

However, Fig. 10 illustrates a MATLAB's training phase used to train 1st Stage BPNN.

The last stage showed in Fig. 7, which is called Assimilated Phoneme is a process that generates the utterance of the assimilated phonemes. The utterance of each assimilation rule is stored and indexed in a file, where each one has a unique identification number opposite to the assimilation rule. Simply, Assimilated Phoneme process retrieves this utterance by its index number. This is much like a database of utterances to be retrieved by their number, which is resulted from the 2nd stage BPNN.

VI. Results

A set of test cases is used to verify the function of the proposed computation approach. Table III illustrates the achievements of multiple experienced architectures of 1st stage BPNN.

This work, up to our knowledge, is the first of its type, which makes the using of comparison approach to evaluate the results, is a difficult task.

Confusion matrix technique is used to validate the achievement of the proposed computation approach. Principally, a confusion matrix is used to evaluate the performance of a classification system. The confusion matrix encompasses two dimensions, one for the actual (negative and positive) classifications, and the other for the classification system predicts (negative and positive). Table IV illustrates the confusion matrix, which records the quantified relations between actual and predicted types of classifications by using the symbols a, b, c, and d [52].

The records of confusion matrix are used to measure several elements of performance like the accuracy, the recall, and the precision, where [52]:

- The accuracy (AC): the ratio of the correct (negative and positive) predictions to its total number, which is calculated as:

$$AC = \frac{a+d}{a+b+c+d} \quad (2)$$

- The recall or true positive rate (TP): the ratio of correctly identified positive cases (actual positive vs. predicted positive) to total actual positive cases, which is calculated by:

$$TP = \frac{d}{c+d} \quad (3)$$

- The precision (P): the ratio of correctly identified positive cases (actual positive vs. predicted positive) to total predicted positive cases, which is calculated by:

$$P = \frac{d}{b+d} \quad (4)$$

Using the results illustrated in Table III, the accuracy, recall, and precision are calculated for each of the experienced architecture reported in Table III. These calculations are shown in Table V, and illustrated as a graph chart in Fig. 11.

TABLE III. EXPERIENCED ARCHITECTURES OF 1ST STAGE BPNN AND THEIR ACHIEVEMENTS

ANN's Architecture (I-H_1-H_2-O)	Learning parameters			Testing using Training Data (error rate)	Testing using Testing Data (error rate)
	Maximum Number of Epochs	Tolerance of Minimizer	Cost Function		
16-20-20-9	100,000	1e-15	mse	6 / 50 (12 %)	38 /80 (48%)
16-20-25-9	100,000	1e-15	mse	5 / 50(10 %)	39 /80 (49%)
16-25-25-9	100,000	1e-15	mse	5 /50 (10 %)	36 /80 (45%)
16-30-25-9	300,000	1e-15	mse	6 /50 (12 %)	37 /80 (46%)
16-30-30-9	350,000	1e-15	mse	5 /50 (10 %)	33 /80 (41%)
16-33-30-9	350,000	1e-15	mse	4 /50(8 %)	31 /80 (39%)
16-33-30-9	400,000	1e-15	mse	4 /50 (8%)	30 /80 (38%)
16-35-30-9	500,000	1e-15	mse	5 /50 (10 %)	29 /80 (36%)
16-35-35-9	450,000	1e-15	mse	5 /50 (10 %)	28 /80 (35%)
16-30-38-9	250,000	1e-15	mse	6 /50 (12%)	28 /80 (35%)
16-30-40-9	250,000	1e-15	mse	5 /50 (10 %)	27 /80 (34%)

TABLE IV. ACTUAL VS PREDICTED CONFUSION MATRIX

		Predicted	
		Negative	Positive
Actual	Negative	a	b
	Positive	c	d

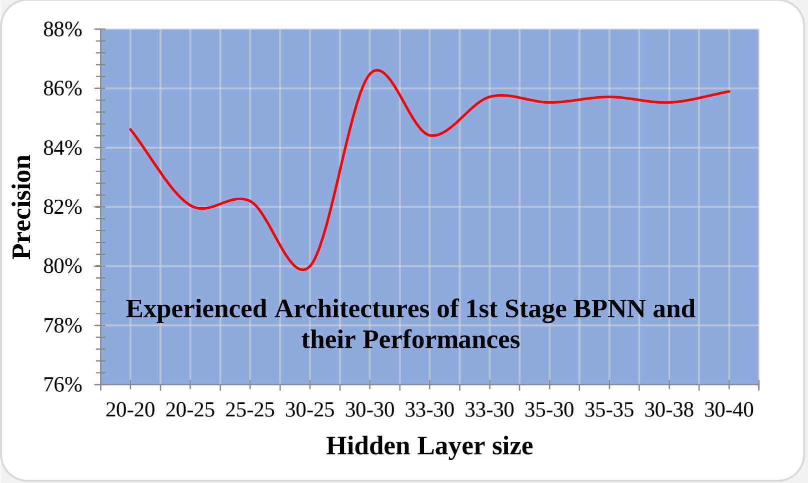


Fig. 11. Performances of experienced architectures.

TABLE V. EXPERIENCED ARCHITECTURES OF 1^{ST} STAGE BPNN AND THEIR PERFORMANCES

ANN's Architecture (I-H$_1$-H$_2$-O)	NN (Value a in Table IV)	NP (Value b in Table IV)	PN (Value c in Table IV)	PP (Value d in Table IV)	Total	Accuracy (%)	Recall (%)	Precision (%)
16-20-20-9	20	12	32	66	130	66.15	67.35	84.62%
16-20-25-9	22	14	30	64	130	66.15	68.09	82.05%
16-25-25-9	29	13	28	60	130	68.46	68.18	82.19%
16-30-25-9	27	15	28	60	130	66.92	68.18	80.00%
16-30-30-9	28	10	28	64	130	70.77	69.57	86.49%
16-33-30-9	30	12	23	65	130	73.08	73.86	84.42%
16-33-30-9	30	11	23	66	130	73.85	74.16	85.71%
16-35-30-9	31	11	23	65	130	73.85	73.86	85.53%
16-35-35-9	31	11	22	66	130	74.62	75.00	85.71%
16-30-38-9	31	11	23	65	130	73.85	73.86	85.53%
16-30-40-9	31	11	21	67	130	75.38	76.14	85.90%

(***NN***: Actual Negative-Predicted Negative, ***NP***: Actual Negative-Predicted Positive, ***PN***: Actual Positive-Predicted Negative, ***PP***: Actual Positive-Predicted Positive)

In addition to the success of using ANN in the application of phoneme assimilation, the results also show that the performance of phonemes' recognition can be raised up as shown by accuracy, recall, and precision factors in Table V. In Fig. 11, however, there is a clear relationship between the size of the hidden layer and the precision performance of the proposed system. It is, as known, hard to form such relationship otherwise one can rely on such formula to design an optimal ANN without needing to practice trial and error approach in the developing of ANN.

By analyzing these results, we may suggest number factors that influence the enhancement of the recognition performance. The first one is relating to LPC coding of the phoneme signal; unfortunately, while the using of LPC coding approach results operative secure communication for sounds of low bandwidth, the quality is not in that goodness and it possibly be intolerable in both the fail to meet the molds of the filter model (like fricative or nasal sounds) and in case that decision of voiced/voiceless is error tolerant [50].

Another reason is that the ANN's recognition's performance is subjected to trial and error principle in the selecting of the number and size of the hidden layers. Keep trying to design different architecture could lead to better achievements. Obviously, this tactic is time consuming.

Not to forget the goodness of recording of the speech (phoneme signals). This factor is of important role that impact the coding/decoding of the phoneme and hence the accuracy of recognition. As much noise exist as more complexity and negative achievement gain.

VII. DISCUSSION

Computation of Arabic Assimilation rules handled by this paper, results a mapping process that locates the proper Assimilation rule of a certain written phonemes combination. The importance of such process is shown in the applications that utter a text (text to speech applications), where the mapping process is a stage in these applications.

A question may come that why mapping nature process and not generation nature process. Actually, the limited number of Assimilation rules supports the developing of mapping nature process. Also, the missing of rules at the level of speech signal that governs the process of assimilation obscures the developing of generation nature process. This challenge is considered as a problem to be solved by future works.

Another question that may come is why the using of ANN as an approach for computation the Assimilation rules. The answer for this question comes from the property of the ANN itself that is the self-learning. This property helps controlling those cases that are not follow the standard defined rules, which comes from different Arabic language dialects. This phenomenon is handled by using MSA, which ignores the existence of accents and limits the process on the standard Arabic. We may suggest here the using of other advanced processing approach like Genetic Algorithm (GA) to compute Assimilation rules.

VIII. CONCLUSIONS AND FINDINGS

In this paper, a BPNN based complex system is proposed to automate the assimilation process of Arabic phonemes. BPNN is used to identify the index of assimilation rule that is required to be applied for a certain combination of Arabic phonemes, and the resulted index of the assimilation rule is used to retrieve the opposite utterance of these assimilated Arabic phonemes.

The theoretical background of the assimilation rules is given and presented as a condition/action form. This description is used for developing a system to automate the assimilation rules of Arabic phonemes. This system encompasses the representing of a phoneme, the developing of a processing machine that yields assimilated phoneme. This processing machine is a form of ANN.

Both traditional LPC approach (for representing phoneme signal) and the MATLAB software toolkit are quite helpful for such kind of applications. The processing machine is composed of two stages. The 1st stage is a BPNN that is responsible for recognizing a phoneme. The 2nd stage is a BPNN also that is responsible for determining the assimilation rule related to the recognized phonemes. The reason that we select ANN as an approach for determining the assimilation rule instead of the simpler technique that is the rule-based is to create more harmonics between 1st stage and 2nd stage from designing point of view. However, while ANN is used previously for voice recognition mainly, here ANN is used for selecting proper assimilation rule for the input Arabic phonemes. This might open the door for researchers in computational phonology in speech composing and generating.

The confusion matrix is used here to measure the performance of the proposed system, which has (in a best-suggested architecture) 75.38% for accuracy, 76.14% for recall, and 85.90% for precision. Although there is no such previously work to make a comparison with, we can, however, and due to many reasons, report the need to enhance the way of design ANN using controlled approach instead of trial and error one used currently. Nevertheless, the experiments show the success of using ANN and the defined approach for handling duties of phonological alternations' process.

References

[1] R. J. Lowe, Phonology Assessment and Intervention Applications in Speech Pathology, Williams & Wilkins, 1994.

[2] ASHA, "Arabic Phonemic Inventories across Languages," American Speech-Language-Hearing Association (ASHA), 2014. [Online]. Available: http://www.asha.org/practice/multicultural/Phono/.

[3] J. Pierrehumbert, "Phonological and Phonetic Representation," *Journal of Phonetics*, vol. 18, no. 3, pp. 375-394, 1990.

[4] J. Bauman-Waengler, Articulatory and Phonological Impairments: A Clinical Focus, Pearson, 2011.

[5] A. B. Smit, Articulation and Phonology Resource Guide for School-Age Children and Adults, San Diego, California, USA: Singular Publishing Group , Inc., 2003.

[6] L. D. Shriberg and R. D. Kent, Clinical Phonetics, Pearson, 2002, p. 116.

[7] A. Lotto and L. Holt, "Psychology of auditory perception," *Wiley Interdisciplinary Reviews: Cognitive Science*, pp. 479-489, 2011.

[8] I. P. A. IPA, "Full IPA Chart," 2015. [Online]. Available: https://www.internationalphoneticassociation.org/sites/default/files/IPA_Kiel_2015.pdf.

[9] U. Goswami, "Phonological Representation," in *Encyclopedia of the Sciences of Learning*, Springer International Publishing AG, 2012, pp. 2625-2627.

[10] P. Boersmá, "Functional Phonology Formalizing the interactions between articulatory and perceptual drives," University of Amsterdam, Amsterdam, Holland, 1998.

[11] P. Boersma, "Some Listener Oriented Accounts of h-Aspire in French," Lingua, vol. 117, no. 12, p. 1989–2054, 2007.

[12] D. Apoussidou, " The learnability of metrical phonology," University of Amsterdam, Amsterdam, Holland, 2007.

[13] P. Boersm, T. Benders and K. Seinhorst, "Neural network models for phonology and phonetics," Unpublished manuscript, Amsterdam, The Netherlands, 2012.

[14] I. Biblawi, Speech Disorders: A Guidance for Speech Pathologists, Teachers, and Parents, Cairo, Egypt: Maktabat Annahda Almasriyah, 2003.

[15] D. Eddington, "Flaps and other variations of /t/ in American English: Allophonic distribution without constraint, rules, or abstractions," *Cognitive Linguistics*, vol. 18, no. 1, pp. 23-46, 2007.

[16] M. Gordon-Brannan and C. E. Weiss, Clinical Management of Articulatory and Phonologic Disorders, USA: Lippincott Williams & Wilkins, 2003.

[17] M. Alkhouli, The Linguistic Sounds, Amman, jordan: Dar Alfalah for Publication, 1990, pp. 219-221.

[18] H. B. Klein, T. M. Byun, L. Davidson and M. I. Grigosa, "A Multidimensional Investigation of Children's /r/ Productions: Perceptual, Ultrasound, and Acoustic Measures," *American Journal of Speech Language Pathology*, vol. 22, no. 3, pp. 540-553, 2013.

[19] J. Panevov´a and J. Hana, "Intro to Linguistics – Phonology," 2010.

[20] J. Yasheng and M. Li, "Acoustic analysis of standard Arabic plosives," in *National Conference on Information Technology and Computer Science*, Lanzhou, China , 2012.

[21] M. Amayreh, "Completion of the Consonant Inventory of Arabic," *Journal of Speech, Language, and Hearing Research*, vol. 46, no. 3, pp. 517-529, 2003.

[22] Amayreh, Mousa M.; Dyson, Alice T., "The Acquisition of Arabic Consonants," *Journal of Speech, Language, and Hearing Research*, vol. 41, no. 3, pp. 642-653, 1998.

[23] I. Khalil, Introduction to Phonology of Arabic Language, Amman, Jordan: Abdelkareem Isma'el Press-Amwaj, 2013, pp. 104-114.

[24] M. Gouskova and N. Hall, "Acoustics of epenthetic vowels in Lebanese Arabic," in *Phonological argumentation: essays on evidence and motivation*, London, UK, Equinox, 2009.

[25] B. Al-Shawabkeh, "Dictionary of Everyday Language in Jordan in the Light of Sociolinguistics," Middle East University, Amman, Jordan, 2013.

[26] G. Khattab, "A phonetic Study of Gemination in Lebanese Arabic," in *16th International Congress of Phonetic Sciences (ICPhS)*, Saarbrücken, Germany, 2007.

[27] M. Amayreh and Y. Natour, Introduction to Communication Disorders, Amman, Jordan: Dar Ilfikr, 2012.

[28] A. Alsina, T. Mohanan and K. Mohanan, "How to get rid of the COMP," in *The Lexical Functional Grammar (LFG05)*, Bergen, Norway, 2005.

[29] J. J. McCarthy, "Consonant harmony via correspondence: Evidence from Chumash," *Papers in Optimality Theory III (University of Massachusetts Occasional Papers in Linguistics)*, p. 31, January 2007.

[30] A. M. Alkhaleel, The Phonetical Terminology According to Ancient Arab Linguists in the Light of Modern Linguistics, Amman, Jordan: Alwataneya Press, 1993, p. 136.

[31] R. Pavlik, "A Typology of Assimilations," *Journal of Theoretical Linguistics*, vol. 6, no. 1, pp. 2-26, 2009.

[32] Dictionary.com, "Dictionary.com LLC," 2016. [Online]. Available:http://dictionary.reference.com/browse/assimilation. [Accessed 2016].

[33] M. K. Hashem, "Assimilation in Arabic Language: Phonologically and Morphologically," *UOBabylon Journal of Humanities*, vol. 18, no. 3, 2010.

[34] D. Jurafsky and J. H. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, NJ, USA: Prentice Hall, 2009.

[35] J. Heinz, "Learning Long-Distance Phonotactics," *Linguistic Inquiry*, vol. 41, no. 4, pp. 623-661, 2010.

[36] C. Samra, P. H. Talukdar and J. Talukdar, "A rule based algorithm for automatic syllabification of a word of Bodo language," *International Journal of Computing, Communications and Networking*, vol. 1, no. 2, pp. 53-56, 2012.

[37] J. Eisner, S. Bird, J. Coleman, J. Goldsmith and L. Karttunen, "Computational Phonology," 2014. [Online]. Available: http://www.ling.fju.edu.tw/phono/computational.htm. [Accessed 5 6 2016].

[38] D. G. Bobrow and J. B. Fraser, "A phonological rule tester," *Communications of the ACM*, vol. 11, p. 766–72, 1968.

[39] S. Bird, A Constraint-Based Approach, Cambridge, England: Cambridge University Press, 1995.

[40] J. S. Coleman, Phonological Representations: their names, forms and powers, Cambridge, England: Cambridge University Press, 1997.

[41] T. Hanneforth, "Finite-state Machines: Theory and Applications," Potsdam, 2010.

[42] M. Mohri, "Finite-State Transducers in Language and Speech Processing," *Computational Linguistics*, vol. 23 , no. 2, pp. 269-311, 1997.

[43] S. Bird, "Computational Phonology," Cornell University Library, NY, USA, 2002.

[44] A. T. Imam, "Relative-fuzzy: A novel approach for handling complex ambiguity for software engineering of data mining models," De

Montfort University, Leicester, UK, 2010.

[45] M. Gasser, "Learning distributed representations for syllables," in *The Fourteenth Annual Conference of the Cognitive Science Society*, Bloomington, USA, 1992.

[46] A. T. Imam, J. Aryfe and I. Hussein, "Automatic Diagnosis of Distortion Type of Arabic /r/ Phoneme Using Feed Forward Neural," *Journal of Computer Engineering and Intelligent Systems,* vol. 5, no. 15, pp. 1-8, 2014.

[47] A. Prince and P. Smolensky, Optimality Theory Constraint Interaction in Generative Grammar, MA, USA: Blackwell Publishing Ltd, 2004.

[48] J. Pater, "Weighted Constraints in Generative Linguistics," *Cognitive Science,* vol. 33, no. 6, pp. 999-1035, 2009.

[49] N. J. Nilsson, "Introduction to Machine Learning (An Early Draft Of A Proposed Textbook)," Stanford, 2005.

[50] L. R. Rabiner and R. W. Schafer, An Introduction to Digital Speech Processing, vol. 1, Delft, The Netherlands: Now Publishers Inc, 2007.

[51] T. MathWorks, "Neural Network Toolbox Documentation," 2016. [Online]. Available: https://www.mathworks.com/help/. [Accessed 10 2016].

[52] Kohavi and Provost, "The Case Against Accuracy Estimation for Comparing Introduction Algorithm," in *ICML '98 Proceedings of the Fifteenth International Conference on Machine Learning*, 1998.