Concatenative Speech Recognition using Morphemes

Afshan Jafri
College of Computer and Information Sciences
King Saud University
Riyadh, SA

***Abstract*—This paper adopts a novel sub-lexical approach to construct viable continuous speech recognition systems with scalable vocabulary that use the components of words to form the elements of pronunciation dictionaries and recognition lattices. The proposed Concatenative ASR family utilizes combination rules between morphemes (prefixes, stems, and suffixes), along with their theoretical grammatical categories. The constrained structure reduces invalid words by using grammar rules governing agglutination of affixes with stems, while having a large vocabulary space and hence fewer out-of-vocabulary words. In pursuing this approach, the project develops automatic speech recognition (ASR) parameterized models, designs parameter values, constructs and implements ASR systems, and analyzes the characteristics of these systems. The project designs parameter values in the context of Arabic to yield a subset hierarchy of vocabularies of the ASR systems facilitating meaningful analysis. It investigates the characteristics of the ASR systems with respect to vocabulary, recognition lattice, dictionary, and word error rate (WER). In the experiments, the standard Word ASR model has the best characteristics for vocabulary of up to five thousand words and the Concatenative ASR family is most appropriate for vocabulary of up to half a million words. The paper shows that the approach used encompasses fundamentally different processes of word formation and thus is applicable to languages that exhibit concatenative word-formation processes.**

***Keywords*—Morphemes; sub-lexemes; speech recognition; Arabic; concatenative morphology**

I. Introduction

The standard automatic speech recognition (ASR) system uses Hidden Markov Models (HMMs) trained on phonetic units, along with a word pronunciation dictionary and a single level recognition lattice composed of words [1]. Application of the standard Word ASR model to vocabulary beyond a hundred thousand words poses complexities, including the construction of the pronunciation dictionary, estimation of the language model, efficient computation of the recognized utterance, and poor recognition performance due to out-of-vocabulary words (OOVs) [2]. For these reasons, the standard Word ASR model is not well suited to languages that are particularly rich in inflectional morphology and that consequently have large vocabularies.

Concatenative word formation of inflectional morphology, by far the most prevalent type in the world’s languages, involves the linear affixation of discrete morphemes, including prefixes, stems, and suffixes.

The concatenative morphology in Arabic is illustrated through two examples provided below. Henceforth the approach drops short vowels as they are not represented in modern Arabic orthography. Table I lists the Arabic characters and their roman transliterations.

The word "فكاتبت", transliterated as "fkqtbt" means ‘so she corresponded’, and demonstrates that a sentence is represented by a single highly inflected word. This word is composed of the stem "kqtb", the prefix ‘f’, and the suffix ‘t’:

prefix+ stem+suffix (1)

f + kqtb + t → fkqtbt ‘so she corresponded’

Another example is the noun "مدرسة", which is transliterated as "mdrsO" and means "school". This word is composed of the stem "mdrs", and the suffix "O". Its derivation is shown below, where ϕ is null:

prefix + stem + suffix (2)

ϕ + mdrs + O → mdrsO ‘school’

By integrating speech recognition constructs with the morphological structure of a given language, the paper aims to develop models that have scalable vocabulary, valid words, moderate computational requirements, and good recognition performance. The objective is to explore the feasibility of sub-lexical models in speech recognition, rather than to optimize the performance of the proposed model families. Consequently, the paper does not deviate into stochastic models, focusing instead on deterministic models.

Vocabulary scalability is attained by constructing a variety of multilevel recognition lattices that utilize the components (sub-lexemes) of words, along with the component categories at different levels of abstraction. The vocabulary is the space of words spanned by the lattice, and the nodes correspond to word components and their categories.

The vocabulary is constrained to valid words in two ways. First, models are defined that constrain the vocabulary of the ASR system and implicitly the word lengths without actually listing words. Second, combination rules are imposed on word components or their categories to eliminate invalid words.

The computational requirements of the ASR system depend on the number of nodes and edges, as well as the structure of the recognition lattice; the size of the pronunciation dictionary; and the search method. Consequently, models with fewer nodes, edges, and items in the dictionary are desirable. Use of word components rather than words to represent nodes and dictionary items reduces the size of both the lattice and dictionary components of the models, thus reducing the computational requirements of the system.

TABLE I. ARABIC CHARACTERS

Ar	Rm	Ar	Rm	Ar	Rm	Ar	Rm
ـً	A	آ	F	ذ	c	غ	x
ـٌ	U	ب	b	ر	r	ف	f
ـٍ	I	ة	O	ز	z	ق	K
ء	Q	ت	t	س	s	ك	k
أ	E	ث	B	ش	Z	م	m
ؤ	M	ج	j	ص	S	ن	n
إ	L	ح	H	ض	D	ه	h
ئ	N	خ	X	ط	T	و	w
ا	aa	ل	l	ظ	C	ي	y
ى	P	د	d	ع	R	ـّـ	G
No standard transliterations between Arabic (Ar) and Roman (Rm)							

Recognition performance as measured by word error rate (WER) is determined by the HMMs and vocabulary of the test set, as well as by recognition lattice vocabulary, lattice structure, and search method. This multitude of factors makes prediction of recognition performance difficult, and hence required careful design of the experiments to produce empirical results that would enable us to measure and compare the recognition performance of different versions of the ASR systems, and to compare these results to those of standard Word ASR system counterparts.

The project's methodology for attaining the above is to: (1) construct parameterized models to build sub-lexical ASR models of increasing complexity and abstraction to attain larger vocabularies; (2) design parameter values in a way that parsimoniously yields a subset hierarchy of a wide spectrum of vocabularies; (3) construct implementable ASR systems using the derived parameter values (4) set up experiments through selection of speech training and test sets, and conduct ASR system training and recognition; (5) investigate the characteristics of the ASR systems with respect to vocabulary, recognition lattice, and word error rate (WER), and observe their robustness with respect to out-of-vocabulary words (OOVs).

The primary objective of this paper is to develop ASR models that are scalable and produce only valid words. Arabic has been chosen as the context for developing this new ASR paradigm. More specifically, Modern Standard Arabic (MSA) is utilized because it is widely used and has well established and standardized grammar and phonetics.

The paper is organized as follows: Section II contains literature review; Section III introduces the parameterized ASR models; Section IV constructs ASR systems for concatenative model; Section V explains how the system is constructed; Section VI discusses the experimental setup; Section VII evaluates the system; Section VIII discusses the results; and Section IX has the conclusion.

II. LITERATURE REVIEW

To overcome the limitations of the Word ASR model, a number of approaches have been suggested that have in common their use of morphemes (prefixes, stems, and suffixes) rather than words as the basic unit of analysis. Indeed, several studies have investigated the use of sub-lexical language constructs in speech recognition [3,4] and models incorporating this idea have been used in many languages, including German and Finnish [5,6], Korean [7,8,9], Dutch [10], Arabic [11,12, 13,14,15,16], Turkish [5,17], Slovenian [18] and English [5]. Other works utilizing such an approach for multiple languages have been published [19,20].

Existing approaches use empirical morphemes and direct relationships between prefixes, stems, and suffixes. They suffer from generation of invalid words because the recognition lattice does not adequately constrain formation of words from morphemes. The invalid words lead to lower recognition performance. The problem is alleviated to some extent by replacing the morphemes of most frequently occurring words by surface forms (complete words) themselves.

Recent work has been conducted for MSA automatic speech recognition utilizing weighted finite state transducer structure in the Kaldi ASR system [21]. Finite state transducer has also been utilized for MSA morphological analysis and diacritization [22].

III. PARAMETERIZED ASR MODELS

The concatenative grammar-based parameterized models' objective is to have increasing levels of complexity and abstraction to attain larger vocabularies. This is achieved by the models by utilizing categories of word components rather than word components alone. The categories reflect two basic sub-lexical classes (stems and affixes) and the objects they can combine with.

The four models are termed: Direct Morpheme, Affix Category, Stem Category, and Full Category in addition to the baseline model called Independent Morpheme (described in the Appendix), which corresponds to currently proposed models in the literature.

With the exception of Independent Morpheme, all of the system's ASR models have a vocabulary of only valid words because they use three-dimensional combination matrices that constrain the relations between morphemes or their categories. The baseline Independent Morpheme model does admit invalid words in the vocabulary because it lacks these constraints.

Each of these grammar-based ASR models has a distinct set of parameters, with the common parameters being Prefix, Stem, and Suffix –more specifically, the indexed listings of prefixes, stems, and suffixes. For the same set of parameter values of Prefix, Stem, and Suffix, the various ASR models have the same terminal nodes comprising prefixes, stems, and suffixes, and the same dictionary, whose items are the union of prefixes, stems and suffixes.

However, the models have distinct recognition grammars. The reason for the distinct recognition grammars is that the models use component categories and different two-dimensional binary association matrices defining associations between components and their categories, as well as three-dimensional binary combinations defining licit combinations between morphemes or between their categories. The morphemes, categories, associations, and combinations are based on theoretical morphological grammar.

A. Direct Morpheme ASR Model

The Direct Morpheme ASR parameterized model involves the most constrained structure, incorporating direct combination constraints among prefixes, stems, and suffixes. The parameters are Prefix, Stem, and Suffix, and the binary three-dimensional combination matrix PrefixXStemXSuffix. The recognition grammar is given below:

'#' {Word '&' }$^+$ '#'

Word → WordStem$_1$ | WordStem$_2$ | …;

WordStem$_1$ → stemPrefix$_{11}$ stem$_1$ stemSuffix$_{11}$ |

stemPrefix$_{12}$ stem$_1$ stemSuffix$_{12}$ | …;

WordStem$_2$ → stemPrefix$_{21}$ stem$_2$ stemSuffix$_{21}$ |

stemPrefix$_{22}$ stem$_2$ stemSuffix$_{22}$ | …;

The second line expands a word into stem-grouped words, which share a common stem. The words are not explicitly listed. Each stem-grouped word is a choice of prefix-stem-suffix combinations for the particular stem, as allowed by the combination matrix PrefixXStemXSuffix. An implementable example is shown below:

('#' {Word '&' }$^+$ '#')

Word → WordStem_ktb | WordStem_drs | … ;

WordStem_ktb → ' ' 'ktb' ' ' | 'f' 'ktb' 't' | …;

WordStem_drs → 'w' 'drs' 'h' | 'l' 'drs' 'hmq' | … ;

B. Affix Category ASR Model

The Affix Category ASR parameterized model is both an abstraction of the Direct Morpheme model and potentially more efficient than that model because it classifies affixes (prefixes and suffixes) according to their grammatical categories. The parameters of this model are: Prefix, Stem, Suffix; PrefixCateg, SuffixCateg; the binary association matrices Prefix_PrefixCateg and Suffix_SuffixCateg; and the binary combination matrix PrefixCategXStemXSuffixCateg. The recognition grammar for the Affix Category parameterized model is:

'#' {Word '&' }$^+$ '#'

Word → WordStem$_1$ | WordStem$_2$ | …;

WordStem$_1$ → PrefixCateg$_{11}$ stem$_1$ SuffixCateg$_{11}$ | ...
PrefixCateg$_{12}$ stem$_1$ SuffixCateg$_{12}$ | …;

WordStem$_2$ → PrefixCateg$_{21}$ stem$_2$ SuffixCateg$_{21}$ |

PrefixCateg$_{22}$ stem$_2$ SuffixCateg$_{22}$ | …;

PrefixCateg$_{11}$ → prefix$_{11}$ | prefix$_{12}$ | …;

PrefixCateg$_{21}$ → prefix$_{21}$ | prefix$_{22}$ | …;

SuffixCateg$_{11}$ → suffix$_{11}$ | suffix$_{12}$ | …;

SuffixCateg$_{21}$ → suffix$_{21}$ | suffix$_{22}$ | …;

The second line expands a word into alternatives among words grouped according to stems. Each stem grouped word is a choice between PrefixCateg-stem-SuffixCateg combinations for the stem, as allowed by PrefixCategXStemXSuffixCateg. Each PrefixCateg and SuffixCateg is expanded into prefixes and suffixes according to the association matrices.

C. Stem Category ASR Model

The Stem Category ASR parameterized model is also an abstraction of the Direct Morpheme model by its classification of stems into their grammatical categories. In classifying stems rather than affixes, this model is more effective than the Affix Category model because the number of stems is much larger than the number of affixes. The parameters are Prefix, Stem, Suffix; StemCateg representing the indexed listing of stem categories; binary association matrix Stem_StemCateg; binary combination matrix PrefixXStemCategXSuffix. The recognition grammar for the Stem Category model is:

'#' {Word '&' }$^+$ '#'

Word → WordStem$_1$ | WordStem$_2$ | …;

WordStem$_1$ → prefix$_{11}$ StemCateg$_1$ suffix$_{11}$ |

prefix$_{12}$ StemCateg$_1$ suffix$_{12}$ | …;

WordStem$_2$ → prefix$_{21}$ StemCateg$_2$ suffix$_{21}$ |

prefix$_{22}$ StemCateg$_2$ suffix$_{22}$ | …;

StemCateg$_1$ → stem$_{11}$ | stem$_{12}$ | …;

StemCateg$_2$ → stem$_{21}$ | stem$_{22}$ | …;

The group for each WordStem is a choice of prefix-StemCateg-suffix combinations for the specific item in StemCateg, as allowed by PrefixXStemCategXSuffix. A specific member in StemCateg is expanded into stems according to the association matrix.

D. Full Category ASR Model

The Full Category ASR parameterized model abstracts all morphemes--prefixes, stems and suffixes--into their grammatical categories, thereby producing the most abstract Concatenative ASR model. The parameters are Prefix, Stem, Suffix; PrefixCateg, StemCateg, SuffixCateg; binary association matrices Prefix_PrefixCateg, Stem_StemCateg, Suffix_SuffixCateg; and binary combination matrix PrefixCategXStemCategXSuffixCateg. The recognition grammar is given below:

'#' {Word '&' }$^+$ '#'

Word → WordStemCateg$_1$ | WordStemCateg$_2$ | …;

WordStemCateg$_1$ →

PrefixCateg$_{11}$ StemCateg$_1$ SuffixCateg$_{11}$ |

PrefixCateg$_{12}$ StemCateg$_1$ SuffixCateg$_{12}$ | ...;

WordStemCateg$_2$ →

PrefixCateg$_{21}$ StemCateg$_2$ SuffixCateg$_{21}$ |
PrefixCateg$_{22}$ StemCateg$_2$ SuffixCateg$_{22}$ | …;

PrefixCateg$_{11}$ → prefix$_{111}$ | prefix$_{112}$ | …;

PrefixCateg$_{12}$ → prefix$_{121}$ | prefix$_{122}$ | …;

StemCateg$_{1}$ → stem$_{11}$ | stem$_{12}$ | …;

StemCateg$_{2}$ → stem$_{21}$ | stem$_{22}$ | …;

SuffixCateg$_{11}$ → suffix$_{111}$ | suffix$_{112}$ | …;

SuffixCateg$_{12}$ → suffix$_{121}$ | suffix$_{122}$ | …;

Each collection of words centered on a specific StemCateg is a choice between PrefixCateg-StemCateg-SuffixCateg combinations for the given StemCateg as allowed by PrefixCategXStemCategXSuffixCateg.

The categories PrefixCateg, StemCateg, and SuffixCateg are expanded into prefixes, stems, and suffixes according to the association matrices Prefix_PrefixCateg, Stem_StemCateg, Suffix_SuffixCateg respectively. An illustrative example is as follows, with FW1Wa denoting a stem category, Pref1Wa a prefix category, and Suff10 a suffix category.

Word_FW1Wa →

Prefix_Pref1Wa Stem_FW1Wa Suffix_Suff10 |

Prefix_Pref10 Stem_FW1Wa Suffix_Suff10 ;

Stem_FW1Wa → ’EbnqQ’ | ’Ef’ | ’Em’ | ’En’ | ’Ew’ |

’Ey’ | ’Eyn’ | ’LtZ’ | ’LBnqn’ | ’Lcq’ | ’Ls’ | ...;

Prefix_Pref1Wa → ’f’ | ’w’;

Suffix_Suff10 → ’ ’;

This section presented four Concatenative grammar-based ASR parameterized models to develop a hierarchy of vocabularies from the same set of parameter values and to provide models suitable for a variety of circumstances.

The Direct Morpheme model is suitable for cases where |Prefix|/|PrefixCateg| ~ 1, |Suffix|/|SuffixCateg| ~ 1, and |Stem|/|StemCateg| ~ 1; the Affix Category model is appropriate for situations where |Prefix|/|PrefixCateg| >> 1, |Suffix|/|SuffixCateg| >> 1, and |Stem|/|StemCateg| ~ 1; the Stem Category model is suitable for cases where |Prefix|/|PrefixCateg| ~ 1, |Suffix|/|SuffixCateg| ~ 1, and |Stem|/|StemCateg| >> 1; and the Full Category model is appropriate for situations where |Prefix|/|PrefixCateg| >> 1, |Suffix|/|SuffixCateg| >> 1, |Stem|/|StemCateg| >> 1.

IV. Parameter Design

This section illustrates how the parameter values and combination matrices are derived and ASR systems constructed for concatenative models. Parameter values are designed to parsimoniously cover a wide spectrum of vocabulary for construction of the implementable ASR systems from the models developed in Section III.

The system vocabulary is derived indirectly by the specification of morphemes, and their combinations and association matrices. This is in contrast to Word ASR, in which systems may be constructed for arbitrary vocabulary sizes.

The careful parameter design yields a subset hierarchy of vocabularies for the ASR systems, thereby facilitating comparative analysis of the various models. Both a language dataset and a speech corpus are used to derive the parameter values for the ASR systems, as the approach combines the speech and language aspects into the development of an ASR system.

The Buckwalter language dataset was chosen because it is the most complete morphological dataset and the Saavb corpus as both a speech and text corpus because it has accurate transcriptions in Modern Standard Arabic validated by IBM [23]. The recognition grammar is generated from the text corpora, while the training and test sets are generated from the speech corpus with the difference between the recognition lattice span and the recognition set determining the out-of-vocabulary (OOV) words.

The Buckwalter dataset contains three lexicon files and three compatibility tables with a vocabulary of more than five million consisting of only valid words. The three lexicon files tabulate the prefixes, stems, and suffixes with their grammatical categories. Categories of stems and affixes reflect both language classification and the objects that they can combine with. The three compatibility files have two-column tables that provide the relations between the following: prefix categories & suffix categories, prefix categories & stem categories, and suffix categories & stem categories.

The parameter values for the ASR models are computed in three stages, which are briefly described below, with details omitted due to space considerations. In Stage I, we compute the various listings, association and combination matrices from the Buckwalter lexicon files, and compatibility tables. To accomplish this, we first compute the indexed listings of unique prefixes, stems, and suffixes from the tokens in the three lexicons, and compute the categories of the prefixes, stems, and suffixes from both the lexicon files and the compatibility tables. Table II lists the sizes of the Buckwalter parameter values, such as BuckwalterStem, BuckwalterStemCateg. Then, the computed indexed listings are used, along with the lexicon files and compatibility tables to produce the two-dimensional binary association matrices (such as Suffix_SuffixCateg) and two-dimensional binary compatibility matrices (such as PrefixXSuffix). These two-dimensional compatibility matrices are used to derive the three-dimensional binary combination matrices (such as PrefixXStemXSuffix).

TABLE II. Buckwalter Morpheme Parameter Sizes*

Parameter	Size
BuckwalterPrefix	131
BuckwalterSuffix	209
BuckwalterStem	43870
BuckwalterPrefixCateg	88
BuckwalterSuffixCateg	173
BuckwalterStemCateg	218
* From original Buckwalter dataset	

In Stage II, the system morphologizes the Saavb corpus words according to the generated Buckwalter listings and matrices to produce the SaavbMorphologicalTable consisting of the following columns: word, prefix, stem, suffix, prefixCateg, stemCateg, and suffixCateg. Because a word may have multiple decompositions, each word in the table may have more than one row corresponding to it. Saavb words that are outside the vocabulary of Buckwalter (mainly mispronunciations) are decomposed as prefix = ϕ, stem = word, suffix = ϕ, and stemCateg = 'NonSubword'. This results in updated values of |BuckwalterStem| = 44212 and |BuckwalterStemCateg| = 219. Henceforth, these extended parameter values are referred to as Buckwalter parameter values.

In Stage III, the subsets of the appended listings are extracted and matrices to define the parameter values. The system computes two groups of subsets of the Buckwalter listings and matrices: Saavb Group and Buck Group. The Saavb Group is created by traversing through the SaavbMorphologicalTable to compute the indexed listing SaavbPrefix, SaavbStem, SaavbSuffix, SaavbPrefixCateg, SaavbStemCateg, SaavbSuffixCateg, as well as the two-dimensional binary association matrices and three-dimensional binary combination matrices. These parameter sizes are summarized in Table III. For the Buck Group, a subset of the Buckwalter listings and matrices is created that is larger than the Saavb Group by extracting subsets of BuckwalterPrefix, BuckwalterStem, and BuckwalterSuffix whose categories are the same as SaavbPrefixCateg, SaavbStemCateg, and SaavbSuffixCateg respectively. The resulting listings are BuckPrefix, BuckStem, BuckSuffix, BuckPrefixCateg, BuckStemCateg, and BuckSuffixCateg, with sizes summarized in Table IV.

TABLE III. SAAVB MORPHEME PARAMETER SIZES*

Parameter	Size
SaavbPrefix	37
SaavbSuffix	34
SaavbStem	1586
SaavbPrefixCateg	36
SaavbSuffixCateg	102
SaavbStemCateg	110
SaavbNonSubword	342

*From Saavb corpus. A member of SaavbPrefix may be associated with more than one member of
SaavbPrefixCateg. Same applies for Suffix and Stem

TABLE IV. BUCK MORPHEME PARAMETER SIZES*

Parameter	Size
BuckPrefix = BuckwalterPrefix	131
BuckSuffix = BuckwalterSuffix	209
BuckStem = BuckwalterStem + NonSubword	44212
BuckPrefixCateg = SaavbPrefixCateg	36
BuckSuffixCateg = SaavbSuffixCateg	102
BuckStemCateg = SaavbStemCateg	110

* From Saavb morpheme categories and modified Buckwalter morphemes

V. CONSTRUCTION OF ASR SYSTEM

The parameters generated in the previous section are used with the ASR parameterized grammar-based models of Section III to construct seven ASR systems with a wide range of vocabularies. The Saavb Group parameter values of Table III are used with the ASR models of Section III to construct the following ASR systems: (1) Saavb Independent Morpheme (IM), (2) Saavb Direct Morpheme (DM), (3) Saavb Affix Category (AC), (4) Saavb Stem Category (SC), (5) Saavb Full Category (FC), (6) LargeBuck Full Category (LBFC), and (7) SmallBuck Full Category (SBFC).

The LargeBuck Full Category ASR system is created by using the Buck Group parameters summarized in Table IV with the Full Category model. The SmallBuck Full Category ASR system is built from Saavb Group stems and Buck Group affixes, with the parameters consisting of indexed listings BuckPrefix, SaavbStem, BuckSuffix, SaavbPrefixCateg, SaavbSuffixCateg, SaavbStemCateg summarized in Tables III and IV; the association matrices:

- BuckPrefix_SaavbPrefixCateg,
- SaavbStem_SaavbStemCateg,
- BuckSuffix_SaavbSuffixCateg; and the combination matrix
- SaavbPrefixCategXSaavbStemCategXSaavbSuffixCateg.

The vocabulary sizes of the concatenative ASR systems are listed in Table V. All vocabularies have only valid words except for the Independent Morpheme system. The following represents the subset relations between vocabularies: DM ⊂ AC, DM ⊂ SC, AC ⊂ FC, SC ⊂ FC and FC ⊂ SBFC ⊂ LBFC. The vocabulary of DM is equal to the SAAVB vocabulary by construction.

TABLE V. ASR SYSTEM CHARACTERISTICS

System	Vocabulary	Dictionary	Nodes (Edges)	WER %
Saavb Concatenative ASR Systems and Word counterparts				
IM	1,995.188	1,623	1,666 (3,320)	67
W_IM	1,995.188	1,995.188	1,995.192 (3,990,379)	*
DM	1,719	1,623	5,920 (7,890)	55.7
W_DM	1,719	1,719	1,723 (3,441)	55.8
AC	5,069	1,623	36,778 (55,586)	57.5
W_AC	5,069	5,069	5,073 (10,141)	57.8
SC	30,603	1,623	42,011 (82,208)	63.3
W_SC	30,603	30,603	30,607 (61,209)	64.6
FC	74,543	1,623	68,547 (133,491)	63.6
W_FC	74,543	74,543	74,547 (149,089)	64.6
Buck Concatenative ASR Systems and Word counterparts				
SBFC	226,861	1,875	73,477 (143,292)	63.8
W_SBFC	226,861	226,861	226,865 (453,725)	*
LBFC	5,323,415	44,429	1,135,723 (2,267,729)	*
W_LBFC	5,323,415	5,323,415	5,323,419 (10,646,833)	*

- DM=Direct Morpheme, AC=Affix Category, SC=Stem Category, FC=Saavb Full Category, SBFC=Small Buck Full Category, LBFC=Large Buck Full Category.
- W_ indicates corresponding word ASR
* indicates that recognition experiments were not conducted because of the large lattice size

The dictionary of all the ASR systems consists of pronunciations of the union of Prefix, Stem, and Suffix. Hence, the dictionaries of all Saavb concatenative ASR systems are the same, as indicated in Table V, which also shows the dictionary sizes for the SBFC and LBFC ASR systems. The recognition lattice sizes of the ASR systems are likewise summarized in Table V. The LBFC system, with a lattice size encompassing more than one million nodes and two million edges, is not implementable.

The concatenative ASR model is much more scalable than the standard Word ASR model for languages with inflectional morphology.

VI. EXPERIMENTAL SETUP

This section presents implementation issues of ASR systems. Subsection A presents the conventional word ASR with which comparisons of the proposed ASR systems are made. Subsection B presents training and test sets used in the experiments. Subsection C summarizes the speech training and recognition steps taken.

A. Conventional ASR Model

The standard word ASR model structure is used as a reference to evaluate the ASR models in terms of vocabulary size, computational requirements such as the number of nodes, edges, dictionary size, and recognition performance as measured by the word error rate (WER). The word ASR is the most structured model as the grammar specifies exactly the vocabulary of the recognition system, and hence provides complete control of the character sequences that are allowed.

The EBNF syntax for the word ASR recognition grammar with words being the terminal nodes is as follows: '#' {Word '&' }+ '#'; Word -> 'word1' | 'word2' | 'word3' |.

Although an end of word marker is not needed, '&' is used to be consistent with the grammar of the ASR model structures. An example of the second line is Word = 'fy' | 'mn' | 'RlP' | 'En' | 'LlP' | 'qlty' | 'mNO' |.

The standard Word ASR systems that are build are counterparts to the Concatenative ASR systems by computing the vocabulary of the developed ASR through the span of its recognition lattice, determining the dictionary based on the vocabulary, and constructing a word-loop recognition lattice with the nodes representing the words in the vocabulary. As the counterpart Word ASR systems are generated from the vocabulary of the Concatenative ASR systems, a similar subset hierarchical relationship holds true. Table V lists the vocabulary, dictionary, and lattice size for W_IM, W_DM, W_AC, W_SC, W_FC, W_SBFC, and W_LBFC, where 'W' denotes the word counterpart ASR system.

B. Training and Recognition Sets

The SAAVB speech corpus consists of prompted utterances spoken over cellular telephones in a quiet environment and received by land telephones sampled at 8 kHz. This corpus is appropriate for comparison between the different ASR systems. The data available for the paper consist of a total of approximately 25,000 utterances comprising 50 utterances with an average duration of 5.7 seconds per utterance spoken by each of the 484 subjects, with a vocabulary of 1719 (unique) words.

The utterances are divided into three mutually exclusive and collectively exhaustive sets, A, B, and C. Each balanced set consists of utterances for different speakers. Three partitions are utilized: Training set consisting of A and B with recognition set being C; training set composed of A and C with recognition set being B; training sets B and C and recognition set A.

C. ASR Training and Recognition

The HTK toolkit is used in accordance with standard practices [24]. The HTK command HParse converts the generated EBNF of Sections 2 and 3 into recognition lattices. For each of the utterances, feature vectors are based on MFCC of length thirty-nine. Orthographic transcription is mapped into phonetic sequences using a pronunciation algorithm.

Training of HMMs is conducted on the three partitions. HMMs are left to right non-skip with twelve mixtures and they model the phonetic units associated with the Modern Standard Arabic transcriptions. The K-fold method is used with three folds to implement statistically valid training and recognition tasks [25]. Recognition is conducted using the Viterbi algorithm and the empirical results are obtained by averaging the recognition performance values and time durations for the three folds.

As this research's objective is proposal and analysis of sub-lexical speech recognition, rather than optimization of the proposed models, no optimization is conducted by using context dependent phones, large number mixtures, optimized size and structure of HMM, adaptive techniques, or use of stochastic lattices. Optimized choices may reduce word error rate by approximately 30%.

The ASR systems for Concatenative model have the same phonetic units and HMMs as the Word ASR systems and differ only in the lattice structure. Hence improvements in models of phonetic units would translate towards improvement in performance in the same manner for both Word ASR systems and systems proposed in this paper.

VII. Performance and Analysis

This section analyzes the characteristics of the ASR systems and the results are presented in Table V. It also compares the various sub-lexical ASR systems with their Word ASR counterparts, and derives conclusions on the suitability of the ASR models for the different cases examined in this paper.

Table V lists the vocabulary size, dictionary size, recognition lattice size, and word error rate (WER) for each of the concatenative and word ASR systems. Values for WER for LBFC, W_SBFC and W_LBFC are not available, as empirical experiments could not be conducted due to the large lattice size.

Fig. 1 and 2 plot lattice size and WER versus vocabulary on the log scale for the Concatenative ASR systems, in which the abscissa represents vocabulary of only valid words. The squares represent the DM, AC, SC, FC, and SBFC systems and the circles represent their Word counterparts.

A. Vocabulary

Table V shows that using the same prefixes, stems, suffixes, and dictionary of the Saavb Group, the Concatenative ASR family has vocabularies that range from 1,719 to 74,543, with vocabulary size increasing in relation to the level of abstraction. This finding demonstrates the power of utilizing various levels of abstraction.

Examination of W_SBFC, SBFC and LBFC reveals the empirical limitations of the Word ASR system and the Concatenative ASR system imposed by the lattice size.

All of the Concatenative ASR systems, with the exception of IM, have only valid words. The vocabulary of FC is the maximal vocabulary for an ASR system containing the prefixes, stems, and suffixes of Saavb, and is a subset of IM vocabulary. Thus, the vocabulary size of 74,543 of FC is the number of valid words in IM, suggesting that only 3.7% of the two million words in IM are valid.

B. Lattice Size

Table V reveals the 1:1 ratio of the number of nodes to vocabulary size for Word ASR systems. The Concatenative ASR systems become increasingly more efficient for larger vocabularies, having a smaller lattice than the Word ASR systems for vocabularies of more than 50,000 words. In particular, the Full Category systems (FC, SBFC, LBFC) yield very compact lattices because of combination relations between categories of morphemes rather than morphemes themselves.

C. Dictionary

As illustrated in Table V, the dictionary size of a Word ASR system equals the vocabulary size, and thus poses problems for large vocabulary. In contrast, the dictionary size for Concatenative ASR systems is relatively insensitive to vocabulary size. The dictionary size of Concatenative ASR systems relatively insensitive to vocabulary size in contrast to linear dependency of Word ASR systems.

D. Computation Time

Computation time increases with size of the lattice and dictionary. The Word ASR system exhibits an increasing relationship between the number of nodes and the dictionary size with respect to vocabulary size. Consequently, computation time in a Word ASR system is expected to increase with vocabulary size at a higher rate than in a Concatenative ASR system. In contrast, as the Concatenative ASR systems keep dictionary size constant, their computation time is expected to increase at a slower rate than in the Word ASR system. Empirical computation time versus vocabulary for the Concatenative and Word ASR systems confirms the observations above. While the Word ASR system is more efficient for small vocabulary size, the Concatenative ASR systems are superior for vocabulary sizes greater than 10,000 words.

E. Word Error Rate

In order to avoid miscalculation of the word error rate (WER) due to inflation of the correct rate arising from '&', the WER for the Concatenative ASR systems is calculated by concatenating the prefix, stem / character sequence and suffix through the end-of-word '&' marker, into words.

In general, for any given model, WER is expected to increase as a function of vocabulary size. Because the test set is the same, there are no OOVs, and the vocabulary has a subset structure, this trend can be attributed to larger search space. Accordingly, the order of ASR systems with respect to WER is expected to be the following: FC<SBFC<LBFC for the Concatenative ASR systems, and W_DM<W_AC, W_DM<W_SC, W_AC<W_FC, W_SC<W_FC, W_FC<W_SBFC<W_LBFC for the Word ASR systems.

Comparison of the Concatenative ASR systems to their Word counterparts indicates that a Word ASR system is inferior to a Concatenative ASR system for vocabulary of more than 5,000 words. Even though the Concatenative ASR systems have the same vocabularies as their Word counterparts, the WER can be different because the recognition performance depends not only on vocabulary size but also on the lattice. The lattice structure of the Concatenative ASR is very different from the lattice structure of the Word ASR, even though the recognition lattices have the same vocabulary. The other factors that determine performance, such as HMMs, test set, and lattice search method are the same in both cases.

Comparing WER of IM with FC, and DR with IR, which have deviations of around 4%, provides some indication of the importance of combination constraints to ASR systems, and the effect of inflating vocabulary with invalid words on recognition performance.

F. ASR Systems with OOV Words

This section studies the effect of out-of-vocabulary (OOV) words on the performance of Concatenative and their Word ASR counterparts. In the empirical experiments, the test set is constrained by the speech corpus, and hence the OOV issue is best handled by modifying the vocabulary of the ASR system to exclude some of the words in the test set. Furthermore, in order to provide uniform comparison across all ASR systems, OOV words are fixed for all systems and are not varied according to the ASR system vocabulary.

The vocabulary of Concatenative cannot be specified directly, and hence a practical approach is to specify OOV words as those for which stemCateg have particular value. Words for which stemCateg = 'NonSubword' are a good choice for OOVs, as the stems in this category are not additionally classified under other categories.

The test set is of fixed vocabulary and the systems have a subset hierarchy of vocabularies. Consequently, the deterioration in performance is expected to increase with the increase in vocabulary size of the ASR system. However, this is not an issue in our case because the objective is to compare the deterioration in performance of sub-lexical ASR models proposed in the paper with respect to their Word counterparts.

Comparison of performance of ASR systems with OOV to ASR systems without OOV indicates that the deterioration in performance of the Concatenative ASR systems is comparable to that of Word ASR systems at 3% for Direct Morpheme, reaching an insignificant level for Full Category with higher vocabulary. The models developed and analyzed in this paper are observed to be as robust to OOV as their Word counterparts.

VIII. Results and Discussion

This paper has developed promising Concatenative grammar-based ASR models for languages with distinctly different word formation processes with the objective of vocabulary scalability and good recognition performance, in which words are formed through affixation of prefixes, stem, and suffixes.

Theoretical grammar constructs of a language are used to develop a rich hierarchical structure of ASR models affording scalability. The concept of combination matrices to limit vocabulary to only valid words has been rigorously developed and applied. Empirical experiments show the viability of using Concatenative grammar-based ASR models to attain good recognition performance. Future work can develop stochastic Concatenative ASR models by addressing the issues presented in the paper.

In the experiments, the standard Word ASR model has the best characteristics for vocabulary of less than 5000 words and the Concatenative ASR family is most appropriate for vocabulary up to half a million words. Theoretical grammar-based combination constraints are an important factor in ASRs, and although ASRs without combination constraints have smaller lattices, their vocabularies have a significant number of invalid words and a higher WER.

IX. Conclusion and Future Work

A future research plan is to develop a stochastic concatenative ASR models to improve performance by incorporating statistics of word sequences in the recognition lattice. In contrast to uniform Word ASR lattice which may be extended to stochastic Word ASR by simply supplementing the single level lattice with additional edges between word nodes to reflect bigram statistics, stochastic concatenative ASR model is fundamentally different from the theoretical grammar-based ASR model presented in this paper.

The lattice structures of the stochastic concatenative models need to be developed distinctly for the variety of paradigms developed. Particular attention has to be paid to ensure that stochastic models have vocabularies of only valid words just as the proposed concatenative ASR models which use combination matrices. Morphological processing of valid word lists yields vocabulary with invalid words.

Another challenge in the development of stochastic models is that words have multiple morphological decompositions, and hence the unigram statistics of a component would be based not only on the word unigram, but also on the conditional statistics of the decompositions of a given word. This issue carries on with higher level statistics.

Acknowledgment

This project was funded by the National Plan for Science, Technology and Innovation (MAARIFAH), King Abdulaziz City for Science and Technology, Kingdom of Saudi Arabia, Award Number (10-INF1324-02).

References

[1] Huang, Xuedong, et al. Spoken language processing: A guide to theory, algorithm, and system development. Prentice hall PTR, 2001.

[2] Hirsimäki, Teemu, et al. "Morphologically motivated language models in speech recognition." Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning. 2005.

[3] Seneff, Stephanie. "The use of subword linguistic modeling for multiple tasks in speech recognition." Speech communication 42.3-4 (2004): 373-390.

[4] Sak, Haşim, Murat Saraşlar, and Tunga Güngör. "Integrating morphology into automatic speech recognition." 2009 IEEE Workshop on Automatic Speech Recognition & Understanding. IEEE, 2009.

[5] Creutz, Mathias, et al. "Morph-based speech recognition and modeling of out-of-vocabulary words across languages." *ACM Transactions on Speech and Language Processing (TSLP)* 5.1 (2007): 1-29.

[6] Kneissler, Jan, and Dietrich Klakow. "Speech recognition for huge vocabularies by using optimized sub-word units." Seventh European Conference on Speech Communication and Technology. 2001.

[7] Kwon, Oh-Wook, and Jun Park. "Korean large vocabulary continuous speech recognition with morpheme-based recognition units." Speech Communication 39.3-4 (2003): 287-300.

[8] Ircing, Pavel, Josef V. Psutka, and Josef Psutka. "Using morphological information for robust language modeling in Czech ASR system." IEEE transactions on audio, speech, and language processing 17.4 (2009): 840-847.

[9] Ri, Hyok-Chol. "A usage of the syllable unit based on morphological statistics in Korean large vocabulary continuous speech recognition system." International Journal of Speech Technology 22.4 (2019): 971-977.

[10] Ordelman R., Hessen A., and Jong F., "Compound Decomposition in Dutch Large Vocabulary Speech Recognition," Proceedings of EUROSPEECH, pp. 225 –228, 2003.

[11] Kirchhoff K., Vergyri D., Bilmes J., Duh K., and Stolcke A., "Morphology-Based Language Modeling for Conversational Arabic Speech Recognition," Computer Speech and Language, vol. 20, no. 4, pp. 589-608, 2006.

[12] Choueiter G., Povey D., Chen S.F., and Zweig G., "Morpheme-based language modeling for Arabic LVCSR," Proceedings of ICASSP, 2006.

[13] Lamel L., Messaoudi A., and Gauvain J., "Automatic speech-to-text transcription in Arabic", ACM Transactions on Computational Logic, 2009.

[14] Xiang B., Nguyen K., Nguyen L., Schwartz R., and Makhoul J., "Morphological decomposition for Arabic broadcast news transcription", ICASSP, 2006.

[15] Diehl F., Gales M., Tomalin M., and Woodland C., "Morphological Decomposition in Arabic ASR Systems" Computer Speech and Language, 2012.

[16] Mousa A., Schl¨uter R., and Ney H., "Investigations on the use of morpheme level features in language models for Arabic LVCSR", ICASSP, 2012.

[17] Liu, Chang, et al. "Evaluating Modeling Units and Sub-word Features in Language Models for Turkish ASR." 2018 11th International Symposium on Chinese Spoken Language Processing (ISCSLP). IEEE, 2018.

[18] Rotovnik T., Maučec M., and Kačič Z., "Large Vocabulary Continuous Speech Recognition of an Inflected Language Using Stems and Endings," Speech Communication, vol. 49, no. 6, pp. 437-452, 2007.

[19] Ablimit, Mijit, Tatsuya Kawahara, and Askar Hamdulla. "Morpheme Segmentation and Concatenation Approaches for Uyghur LVCSR." International Journal of Hybrid Information Technology 8.8 (2015): 327-342. From 5.

[20] Donaj, Gregor, and Zdravko Kačič. "Speech Recognition in Inflective Languages." Language Modeling for Automatic Speech Recognition of Inflective Languages. Springer, Cham, 2017. 5-29.

[21] Menacer, Mohamed, et al. "An enhanced automatic speech recognition system for Arabic." *The third Arabic Natural Language Processing Workshop-EACL 2017*. 2017.

[22] Alkhairy, Maha, Afshan Jafri, and David A. Smith. "Finite State Machine Pattern-Root Arabic Morphological Generator, Analyzer and Diacritizer." *Proceedings of The 12th Language Resources and Evaluation Conference*. 2020.

[23] Buckwalter T., "Buckwalter Arabic Morphological Analyzer," Version 2.0; Linguistic Data Consortium, December 2004.[22] AlGhamdi M., AlHargan F., AlKanhal M., Alkhairy A., Eldesouki M., and Alenazi A., "Saudi Accented Arabic Voice Bank," Journal of King Saud University: Computer Sciences and Information, vol. 20, pp. 43-58, 2008.

[24] Young S. et al., The HTK Book (for HTK Version 3.4), Cambridge University, 2006.

[25] Blum A., Kalai A., and Langford J., "Beating the Hold-Out: Bounds for K-fold and Progressive Cross-Validation," Proceedings of COLT/99, pp. 203-208, 1999.

Appendix

The Independent Morpheme ASR parameterized model has no constraints imposed on the prefix-stem-suffix combinations either directly or indirectly, and hence allows invalid words in the vocabulary. The parameters of this model are the indexed morpheme listings Prefix, Stem, and Suffix. These models correspond to currently proposed models in the literature. The recognition grammar is illustrated below:

'#' {Word '&' }$^+$ '#'

Word -> Prefix Stem Suffix;

Stem -> 'mktb' | 'mdrsO' | 'ktb' | 'mktbO'…

Prefix -> prefix1 | prefix2 | …;

Suffix -> suffix1 | suffix2 | …;

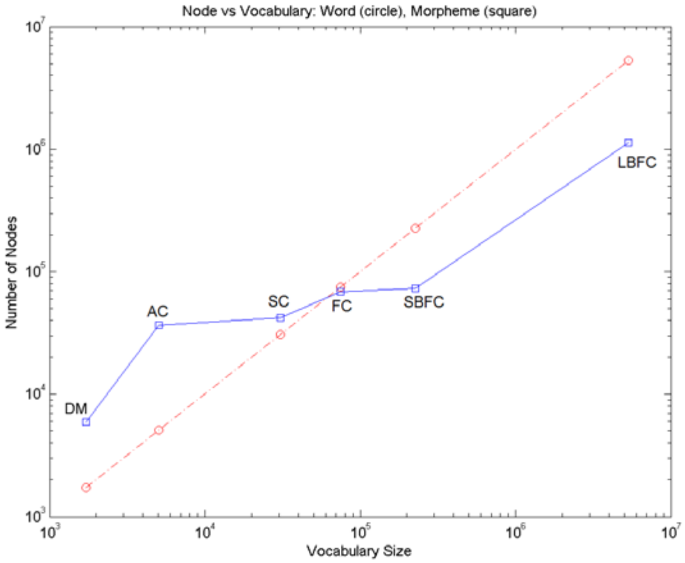

Fig. 1. Node vs Vocabulary. Comparison of the Relation between Number of Nodes and Size of Vocabulary in Word ASR Systems and Concatenative ASR Systems. Circles and Dashed Lines Represent Word ASR Systems; Squares and Solid Lines Represent Concatenative ASR Systems. [Left to Right: DM=Direct Morpheme, AC=Affix Category, SC=Stem Category, FC=Saavb Full Category, SBFC=Small Buck Full Category, LBFC=Large Buck Full Category (LBFC)]. The Figure Shows that Concatenative ASR Systems are more Efficient with Increasing Vocabulary Size, Surpassing Word ASR Systems for Vocabulary of more than 50,000 Words.

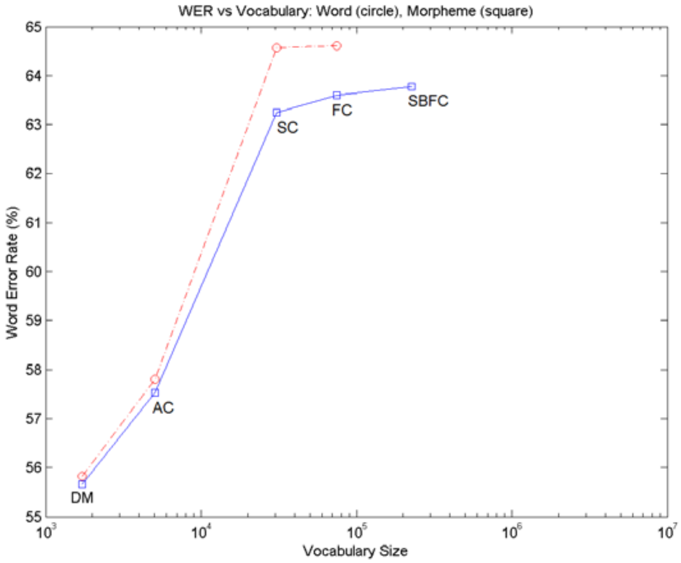

Fig. 2. WER vs. Vocabulary. Comparison of the Relation between Word Error Rate (WER) and Size of Vocabulary in Word ASR Systems and Concatenative ASR Systems. Circles and Dashed Lines Represent Word ASR Systems; Squares and Solid Lines Represent Concatenative ASR Systems. [Left to Right: DM=Direct Morpheme, AC=Affix Category, SC=Stem Category, FC=Saavb Full Category, SBFC=Small Buck Full Category]. The Figure Shows that WER of Concatenative ASR Systems are Lower than Word ASR Systems for Vocabulary Size Larger than 5,000 Words.