A Finite-State Approach To Arabic Broken Noun Morphology

Naser k. A. Alajmi, Prof. Dr. Safa'ai bin deris
Dr. Salah Alnajem
Faculty of Computer Science & Information Systems
Universiti Tcknologi Malaysia
niserka@yahoo.com

ABSTRACT

In this paper, a finite-state computational approach to Arabic broken plural noun morphology is introduced. The paper considers the derivational aspect of the approach, and how generalizations about dependencies in the broken plural noun derivational system of Arabic are captured and handled computationally in this finite-state approach. The approach will be implemented using Xerox finite-state tools.

Key words

Finite state, Arabic morphology, Plural broken plural.

Introduction

The basis systems of most natural language processing are formed by using Morphological analysis techniques. Morphology is the branch of linguistics that deals with the internal structure of words. It studies word formation, including affixation behavior, roots, and pattern properties (Al-Khuli, 1991; Hull & Grefenstette, 1996; Krovetz, 1993). Morphology can be classified as either inflectional or derivational (Aref, 1997; Hull & Grefenstette, 1996; Krovetz, 1993; Spencer, 1991). Inflectional morphology is applied to a given stem with predictable formation. It does not affect the word's grammatical category, such as noun, verb, etc. Case, gender, number, tense, person, mood, and voice are some examples of characteristics that might be affected by inflection. Derivational morphology, on the other hand, concatenates to a given word a set of morphemes that may affect the syntactic category of the word. The distinction between these two classes is not an easy one to make, and it differs from one language to another (Al-Sughairyer, 2004). For instance, Arabic morphology represents a special type of morphological system. It is a non-concatenative (non-agglutinative) system, that is, one which depends on manipulating root letters (radical letters) in a non-concatenative manner using processes like infixation and gemination. Word structure is therefore not built by concatenating (linking together) morphemes as is the case in concatenative morphological systems such as in English. As a result, Arabic morphology poses descriptive, theoretical, and computational challenges quite distinct from those posed by concatenative morphological systems. While, Arabic morphology has many aspects of regularities and sub-regularities. Alnajem (2004b) says there are dependencies in the derivational system of Arabic morphology which have not been formalized and implemented effectively. And also there are generalizations and syncretisms in the inflectional system which have not been considered nor implemented in a perfect manner. This means that special approaches are required to deal with this special nature of Arabic morphology. These approaches should capture the dependencies, generalizations and syncretism existing in the derivational and inflectional system of Arabic. In this paper, a finite-state computational approach to Arabic broken noun morphology is introduced. The paper considers the derivational aspect of the approach, and how dependencies in the broken noun derivational system of Arabic are captured and handled computationally in this finite-state approach. The importance of this work is to make the Arabic Morphology handles generating and analysis processes in a compact non-redundant manner, when it is implemented in many useful applications, such as information retrieval, text categorization, dictionary automation, text compression, data encryption, vowelization and text dictrization and spelling aids, automatic translation, and computer-aided instruction.

Arabic Morphology Balance

Verbs and nouns in Arabic are structured from roots which consist of three or four letters. From theses roots, verbal stems are derived using a number of canonical forms known by the Arab grammarians as the Morphological Balance Forms (MB forms).

The forms themselves use abstract letters (*f, 9*, and *l*) to represent root (radical) letters as follows:

- The first radical letter of the root is mapped to '*f*'
- The second radical letter of that root is mapped to '*9*'
- The third radical letter is mapped to '*l*'
- The fourth radical letter (in quadriliteral roots) is mapped to an additional '*l*' known as '*l2*'. The MB forms have vowels to represent the stem vocalization in addition to derivational stem-affixes such as the derivational stem–infixes, so that they enable mapping technique to map root letters to MB forms to produce verbal or nominal stems. The stems function to construct verbs or nouns through prefixing suffixing inflectional prefixes and suffixes to those stems. Examples of these are the verbal MB form *fa9al* ($C_1aC_2aC_3$) and nominal MB form *mif9aal* ($miC_1C_2aaC_3$). The two MB forms produce verb stem (*fatah, to open*) and noun stem (*miftaah,key*) form the root (*fth*) after mapping $f \rightarrow f(C_1)$,$9 \rightarrow t(C_2)$ and $l \rightarrow h$ (C_3). These forms are known as Measures or traditional Arabic mizan (balance) by the Western Linguists or Arab.

CV Approach

MacCarthy (1981) proposed a linguistic model for Arabic morphology under the framework of autosemental phonology. In this model stem is represented by three types of morphemes: root morphemes which consist constants, vocalism morphemes consist of vowels and pattern morphemes are CV-skeletal which consist of both vowels and constants; some stems includes affix morphemes. Each morpheme sits on its own autonomous ties in the autosegmental model. An example of this model is the analysis of the verb /*katab*/ (to write) has three morphemes: root morpheme {*ktb*}, vocalism morpheme {*a*} and the pattern morpheme {CVCVC}.

Computational Approach To Arabic Morphology

Due to nature of Arabic morphology, Arabic morphology is a different type of morphological system. This special nature of Arabic morphology required special model in order to handle it computationally. Kay (Kay 1987) introduced a finite-state model for the generation and analysis of Arabic morphology. This finite-state model was inspired by the theory of Autosegmental Phonology, and by the Templatic Analysis of Arabic morphology which was introduced by McCarthy (McCarthy 1981, 1982). In his proposal, Kay uses finite-state Transducers which work with four tapes. One tape is for the root tier, another for the pattern tier, third for vocalism tier and fourth for the surface string which is generated or analyzed. These four tapes are read and combined by Kay's FSTs at the same time instead of reading two tapes, as normally happens. Thus, Kay introduces, in fact, a multilinear (multitiered) computational analysis of Arabic morphology through the use of multiple tape FSTs.

Beesley (Beesley 1991) introduced a large computational system for Arabic morphology analysis based on the Two-Level Morphology approach. This model uses two lexicons: one for the roots and another for the patterns precompiled with their vocalisms. The root and pattern interdigitation is done using an algorithm called Detouring. Detouring technique consists of programs that allow the system to follow, at the same time, a lexical path through two lexicons, which are the root and pattern lexicons. This process is done at run time. Two level rules are used to deal with morphological and orthographic variations. These rules map abstract lexical strings, obtained through Detouring, to surface strings. In 1995, this system was rebuilt using Xerox Finite-State Tools. Root, pattern, and vocalisms interdigitation is done through intersection. Intersection is a mathematical operation supported by the Xerox Finite-State Tools. So, the stem daras is a result of intersecting the root {*drs*}, the vocalism {*aa*} and the pattern {CVCVC}. Figure one illustrates how the stem daras is constructed (Beesley 1996, 1998):

Abstract lexical level:
Drs+FormI+Perfect+Active
Abstract intermediate level:
Drs+CVCVC+aa
Abstract intersected level:
Daras
Figure 1

The transducers which correspond to the abstract levels of figure one are composed together to produce a single transducer called the Lexicon transducer. After this composition, the intermediate levels disappear and we are left with an upper-level lexical

string (like: Drs+FormI+Perfect+Active) and a lower-level lexical stem which is still abstract. Finite-state two-level rules are used to map the abstract lexical stem of the lexicon transducer to surface string. These rules are used to handle deletion, assimilation, and other variations. These rules are intersected together to form a single finite-state transducer. This transducer is composed to the bottom of the lexicon transducer. On the top of the lexicon transducer, another transducer is composed witch maps unwanted feature tags to Epsilons. This transducer also applies long-distance morphotactic restrictions. This composition produces a transducer known as the Lexical Transducer which maps a lexical string containing a root and feature tags (the upper string) to a surface string (the lower string) as illustrated in figure Two. The intermediate levels disappear. All the process mentioned above are done at compile time, not at run time as is the case it Beesley's 1989 system.

Upper Level: Drs+CVCVC+aa

Lower Level: daras

Figure 2

Inspired by Kay (Kay 1987), Kiraz (Kiraz 1994, 2001) introduced a multi-tape two-level computational model of Arabic morphology. This model is built on a developed version of the two-level morphology technique introduced by Pulman and Hepple (Pulman and Hepple 1993) and others. In this developed version, the two-level rules map between strings of lexical and surface symbols. This is different from what we have in the conventional two-level rules where we deal with one symbol at a time. This developed version allows encoding surface and lexical contexts in the two level-rules

In addition to using the previous developed version of the two-level morphology technique, Kiraz adopts Kay's (Kay 1987) idea of using four tapes to deal with Arabic morphology. The approach is similar to Kay's in that Kiraz uses three tapes to represent the lexical level and one tape to represent the surface level. Beside the previous conventions, Kiraz introduces extensions related to the expressions in the lexical side of the two-level rules. Kiraz applied this computational model using three linguistic theories: the Templates Analysis (McCarthy 1981, 1982), the Moraic Analysis (McCarthy and Prince 1990), and the Affixational Analysis (McCarthy 1992). Beside theses approaches, other finite-state approaches to Arabic morphology were introduced which include Kornai (Kornai 1991), Bird and Ellison (Bird and Ellison 1992, 1994), and Narayanan and Hashem (Hashem 1993).

The majority of models suggested to model Arabic morphology used the computational power of finite-state Automata, so that finite-state Morphology is considered as the main computational means used for modeling Arabic morphology. On the other hand, other non-finite-state approaches to Arabic morphology have been suggested. Cahill (Cahill 1990, 1991), Gibbon (Gibbon 1990), Reinhard and Gibbon (Reinhard and Gibbon 1991), and Alnajem (Alnajem 1998) suggested three models of Arabic morphology using the Default Inheritance approach.

A Finite-State Approach To Arabic Broken Noun Derivation

This section considers the derivational part of a finite-state approach to Arabic broken noun morphology, in which I attempts to capture dependencies in broken noun derivation; and generalizations, sub-generalizations, and syncretisms governing broken noun inflection. The inflectional part of the approach was considered in (Alnajem; 2002a, 2002d). The orthographic part of the approach was considered in (Alnajem; 2002b, 2002c). The approach has been actually implemented using Xerox Finite-State Tools. To handle dependencies, stem structure has been split into two parts. Mathematical operations on finite-state transducers have been used to computationally handle dependencies between measures. These operations are union, concatenation, and composition. In addition to these operations, subtraction is used to handle root and pattern interdigitation and measure selection. This approach is a bi-directional approach which can be used for generation as well as analysis. The approach is part of a larger computational approach to the derivation and inflection of Arabic verbal and nouns morphology.

Alnajem (2004b) mentions focusing on capturing the dependencies, generalizations, sub-generalizations, and syncretisms in Arabic morphology is absent in the computational systems and approaches introduced in the literature. Capturing such generalizations, dependencies, and syncretisms yields a linguistically motivated computational formalization for Arabic morphology. It also saves this formalization from the redundancy

caused by ignoring such dependencies, generalizations, sub-generalizations, and syncretisms; which are overt in Arabic morphology.

Current Issues and Futures Directions

Perhaps, the major problems and current issues of Arabic morphology are following. First, the very poor researches and tools developed in this field, such example, Beesley has discovered that detouring operations in real time were inherently inefficient, since the resulting system was rather slow, analyzing about two words per second on a small IBM mainframe (Beesley, 1996). In addition, Ali (1988) mentioned that, even though two-level finite-state systems are capable of analysis and generation, they are more suitable for languages with concatenative morphology than for Semitic languages. This is true in that traditional two-level systems were not able to deal with Semitic languages without some modifications. Second, maybe, it is the complexity of the standard system of Arabic orthography because of the great irregularities between the lexical and surface strings caused by phenomena including weak roots, hamza orthography, and the zero realization of short vowels and gemination mark (Beesley, 1991). Moreover, Kiraz (1995) believes that nobody has dealt computationally with the challenging problem of the Arabic broken plural. Broken plurals are not handled by most of current Arabic stemmers (Xu & Weischedel, 2002). Third, Focusing on capturing the dependencies, generalizations, sub-generalizations, and syncretisms in Arabic morphology is absent in the computational systems and approaches introduced in the literature, which yields a linguistically motivated computational formalization for Arabic morphology. It also saves this formalization from the redundancy caused by ignoring such dependencies, generalizations, sub-generalizations, and syncretisms; which are overt in Arabic morphology.

Conclusion

This paper introduced a finite-state computational approach to the derivation of Arabic Broken Nouns. This finite-state computational approach captured generalizations about dependencies governing the derivation of Arabic Broken Nouns. Capturing such generalizations saves our computational approach from redundancy which is caused by ignoring such generalizations. To handle dependencies, stem structure has been split into two parts. Mathematical operations on finite-state transducers have been used to computationally handle dependencies between measures. These operations are union, concatenation, and composition. In addition to these operations, subtraction is used to handle root and pattern interdigitation and measure selection. This approach is a bi-directional approach which can be used for generation as well as analysis. The approach is part of a larger computational approach to the derivation and inflection of Arabic verbal morphology.

References

Alnajem, S. (1998). Computational approaches to Arabic morphology. Ph.D. dissertation, University of Essex.

Alnajem, S. (2002a). A finite-state approach to Arabic verbal inflection. Ms., University of Kuwait.

Alnajem, S. (2002b). A computational approach to the variations in Arabic verbal orthography. Ms., University of Kuwait.

Alnajem, S. (2002c). A computational approach to Arabic orthographic relaxation. Ms, University of Kuwait.

Alnajem, S. (2002d). Manhaj haasoubi lit taʕaamul maʕa ?isnaad al ?afʕaal ?ilad damaa?ir. Arabic Ms., University of Kuwait.

Al-Khuli, M. (1991). A dictionary of theoretical linguistics: English-Arabic with an Arabic-English glossary. Published by Library of Lebanon. Al-Sughaiyer, &. Al-Kharashi. Journal of the American Society for Information Science and Technology Volume 55, Issue 3, 2004. Pages 189-213

Aref, M.M. (1997). Object-oriented approach for morphological analysis.

Beesley, K. (1991). Computer analysis of Arabic morphology: A two-level approach with detours. In Comrie, B. & Eid, M. (eds.), Perspectives on Arabic linguistics III: Papers from the third annual symposium on Arabic linguistics. Amsterdam: John Benjamin's Publishing Company. 155-172.

Beesley, K.R. (1996). Arabic finite-state morphological analysis and generation. In Proceedings of the 16th International Conference on ComputationalLinguistics (COLING-96), Volume 1 (pp. 89–94), Universityof Copenhagen, Denmark

Beesley, K. (1998). Arabic morphology using only finite-state operations. In Rosner, M. (ed.), Computational approaches to Semitic languages: Proceedings of the workshop. Montreal: University of Montreal. 50-57.

Bird, S & Ellison, T. (1992). One-level phonology: Autosegmental representations and rules as finite-state automata. Technical report, Research Paper EUCCS/RP-51, University of Edinburgh.

Bird, S & Ellison, T. (1994). One-level phonology. Computational linguistics 20(1). 55-90.

Cahill , L. & Evans, R. (1990). An application of DATR: The TIC lexicon. In Proceedings of the ninth European conference on artificial intelligence. 120-125.

Cahill, L. (1991). Syllable-based morphology for natural language processing. Ph.D. dissertation, University of Sussex.

Hull, D.A., & Grefenstette, G. (1996). A detailed analysis of English stemming algorithms. Rank Xerox Research Centre.

Gibbon, D. (1990). Prosodic association by template inheritance. In Daelemans, W. & Gazdar, G. (eds.), Proceedings of the international workshop on inheritance and natural language processing. Tilburg: Institute for Language Technology. 65-81.

Kay, M. (1987). Nonconcatenative finite-state morphology. In Proceedings of the third conference of the European chapter of the association for computational linguistics. Copenhagen. 2-10.

Kiraz, G. (1994). Multi-tape two-level morphology: A case study in Semitic non-linear morphology. In Proceedings of COLING 94. 180-186.

Kiraz, G. (2001). Computational nonlinear morphology: With emphasis on Semitic languages. Cambridge: Cambridge University Press.

Kornai, A. (1991). Formal phonology. Ph.D. dissertation, University of Stanford.

Krovetz, R. (1993, July). Viewing morphology as an inference process. In Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 191–202).New York: ACM.

McCarthy, J. & Prince, A. (1990). Foot and word in prosodic morphology: The Arabic broken plural. Natural language and linguistic theory 8. 209-283.

McCarthy, J. (1981). A prosodic theory of nonconcatenative morphology. Linguistic inquiry 12. 373-418.

McCarthy, J. (1982). Formal problems in Semitic phonology and morphology. Ph.D. dissertation, MIT.

McCarthy, J. (1992). Template form in prosodic morphology. In Stvan, L. et al. (eds.), Papers from the third annual meeting of the formal linguistics society of Midamerica. Bloomington: Indiana University Linguistics Club. 187-218

Narayanan, A. & Hashem, L. (1993). On abstract finite-state morphology. In Proceedings of the sixth conference of the European chapter of the association for computational linguistics. 297-304.

Pulman, S. & Hepple, M.. (1993). A feature-based formalism for two-level phonology: A description and implementation. Computer Speech and Language 7. 333-358.

Reinhard, S. & Gibbon, D. (1991). Prosodic inheritance and morphological generalizations. In Proceedings of the fifth conference of the European chapter of the association for computational linguistics. Berlin. 131-136.

Roche, E. & Schabes, Y. (1997). Introduction. In Roche, E. & Schabes, Y. (eds.), Finite-state language processing. Cambridge: MIT Press. 1-66.

Spencer, A. (1991). Morphological theory. Oxford: Basil Blackwell.