Abdelhadi Soudi[1], Violetta Cavalli-Sforza[2]
[1]Center for Computational Linguistics, Ecole Nationale de l'Industrie Minérale, Rabat, Morocco ; asoudi@gmail.com
[2]Language Technologies Instiute, Carnegie Melon University, Pittsburgh, United States ; violetta@cs.cmu.edu

Arabic generation in an Interlingua-based Machine Translation system[1]

Abstract

In this paper, we show how Arabic sentences are generated from the Interlingua representations (IRs) used in the KANT knowledge-based machine translation (MT) system, an interlingua-based software architecture for translation from English to several languages (Nyberg, E.H. and Mitamura, T. (1992)). The system which generates Arabic sentences from Interlingua Representations consists of 4 subsystems: the mapping system, the sentence generation system, the sentence/morphology generation interface and the morphological generation system. We describe these subsystems and address some issues in generating Arabic from IRs. In what follows, we provide a brief description of the subsystems above. A detailed description will be provided in the final version of the paper:

1. Arabic morphology generation: Benefiting from the findings of morphology and computational morphology, we propose an approach to Arabic morphology which is based on the lexeme concept, in contrast to morpheme-based approaches which have standardly been used. We use the theory of lexeme-based morphology (LBM) (Aronoff (1994), and Beard (1995) to represent the linguistic resources and EMORPHE (Violetta and Soudi (2003), an enhanced version of MORPHE (Leavitt (1994)), as a computational tool to perform them. (E)MORPHE is a tool that compiles morphological transformation rules into a word generation program. The basic premise of LBM is that the stem is the phonological domain of realization rules. It is argued that such a morphological theory captures generalizations in the Arabic morphological system. The system's enhancements triggered by the linguistic analysis significantly reduce the number of rules required for the generation of the large variants of Arabic words. Such a reduction keeps the system small and also increases its understandability and maintainability. We provide generation examples of Arabic verbs (strong and weak) and nouns (sound and broken). The exclusion of affixes from the lexicon allows us to test our approach with a non-fragmented lexicon (i.e., without sub-lexicons: a sub-lexicon for vocalism, a sub-lexicon for roots and another sub-lexicon for patterns).

2. Arabic sentence generation: To generate Arabic sentences, we use Genkit (Generation) Kit (Tomita M. and Nyberg E.H. (1988)), a system that compiles a grammar written in a formalism called Pseudo-Unification Grammar into a sentence generation program. The generator follows a top-down, depth-first strategy for applying rules during generation. We describe the tool, provide examples of unification-based grammar rules for generating sentences and address a couple of issues related to the generation of different syntactic structures in Arabic.

3. Mapping system: The mapping system produces FSs for Arabic from IRs, using a set of mapping rules and a mapping lexicon. An FS is a list of feature-value pairs that reflects the syntactic structure of the target language. A mapping rule is a set of slots and values that specify operations involved in building an FS. Target language lexicon entries are FSs. They are retrieved during mapping and added to the sentence FS under construction. We show that the generation of properly inflected Arabic verbs and nouns could be a concern of both the mapper and the generator. For example, the generation of correct agreement between nouns and their modifiers or other parts of the sentence may be performed either during mapping or during generation [2]. In this context, Different agreement cases are considered.

The Arabic sentence generation system has been tested on 35 different structures and has produced good results. In this paper, we will also show some of the limitations of the software technology used in the generation of Arabic sentences.

[2] The morphology/generation interface consists of a lisp program that defines some functions that are used to call the morphological generator from the sentence generator.

[1] This work (in progress) is partially supported by the National Center for Technical and Scientific Research via PROTARS III research grant n° D41/09