نظام خبير كأداة للترجمة الآلية
حسنى المحاسب ومصطفى عارف
قسم علم الحاسب الآلي والعلومات
جامعة الملك فهد للتول والمعادن
الظهران 31261 - المملكة العربية السعودية
KHABEER (خبير) AS A MACHINE TRANSLATION TOOL

Husni A. Al-Muhtaseb¹ and Mostafa M. Aref²
Information and Computer Science Department
King Fahd University of Petroleum and Minerals
Dhahran 31261, Saudi Arabia
E-Mail: ¹husni@ccse.kfupm.edu.sa ²aref@ccse.kfupm.edu.sa

ABSTRACT: Khabeer is an Arabic expert system shell supports object oriented programming. This paper presents the use of Khabeer as a machine translation tool. Several phases of machine translation are demonstrated. These phases include lexical and morphological analysis, syntax analysis, knowledge representation and sentence generation. Due to the fact that Khabeer was developed to help in building Arabic-related applications, the paper emphasizes the use of Khabeer in dealing with Arabic sentences. This includes morphological analysis and syntax analysis of Arabic sentences and Arabic sentence generation.

1. Introduction

Khabeer (خبير) is an Arabic CLIPS-based expert system tool [1-5] developed using the conventional language C. Khabeer uses rules as its primary knowledge representation approach and supports a rich pattern-matching language for specifying rule conditions. It has also object oriented features and a rich query language. All commands and syntax of Khabeer are written in Arabic. This paper presents Khabeer as a machine translation tool. In machine translation a script written in a source language is translated automatically to a target language. The process of translation undergoes through different number of steps depending on the paradigm or the approach used. There are several approaches used in machine translation, some of which are transfer-based approach, Interlingual-based approach, translation by example approach, etc.. Several phases are common in most of these approaches. A typical machine translation system can have a lexical and morphological analyzer, syntax analyzer, a knowledge base system and a sentence generator. In the following sections we demonstrate the use of Khabeer in implementing these phases. Section 2 presents using Khabeer in lexical and morphological analysis. Section 3 presents the use of Khabeer in a syntax analyzer. Knowledge base implementation issues are presented in section 4. Section 5 is dedicated to sentence generation. The conclusion is presented in section 6.

2. Lexical And Morphological Analysis

Arabic lexical and morphological analysis can be described as processing Arabic sentences at the word level. The first step is to break a sentence into tokens. Then, each token is analyzed into its components: prefix, infix, suffix and word stem. The word stems are the basic forms of words that have been stored in the knowledge base. Non-word tokens are separated from the words. Word stems are checked for existence in the knowledge base and their categories are determined. The affixes (prefix, infix, and suffix) are used to determine the categories of tokens of a given sentence. Figure 1 shows an input sentence broken into tokens. Several morphological rules are applied to these tokens to determine their stems and categories.
Figure 1. An Arabic sentence broken into tokens

Figure 2 shows two examples of Arabic morphological rules. In the first rule, a token is broken into two parts. The first part is compared with “ال” and the second part is checked for existing Arabic stem. Same code can be generalized to check for any prefix. In the second rule, a token, except its last letter, is checked for existing Arabic stem. The Arabic syntax of Khabeer makes it easy to express these two rules and others by naming rules, variables and functions.

Figure 2. Examples of Arabic Morphological Rules.

3. Syntax Analysis

The purpose of the syntactical analysis is to transform the surface structure of a sentence into a deep structure [6]. This is done through transformation rules that reflect the Arabic grammar rules. Khabeer as a production system provides the format of these transformation rules. These rules describe different components of Arabic grammar such as: nominal sentences, verbal sentences, prepositional phrases, adjectives, adverbs, etc. Khabeer easily allows the implementation of these Arabic transformation rules. Organized sets of transformation rules for Arabic are well categorized in [7-9]. Khabeer rules can be used to describe different components of Arabic grammar where these components can be expressed in a natural way. Two examples of the grammar rules are shown in Figure 3. The first example demonstrates a rule to exchange the positions of the subject of a sentence and a tool used by the subject. The second example figures out the existence of one type of Arabic phrases, the prepositional phrase (جار ومحرر).
Figure 3. Examples of Arabic Grammar Rules.

4. Knowledge Representation

Knowledge base is an essential part of any machine translation system. The knowledge base should not only contain the word stems of the language, but it should also contain the classification of these stems, their attributes and procedures (demons) that may be used in the morphological and syntactical analysis. Khabeer, as an object-oriented tool, provides several essential features to support such needs. Some of these object-oriented features are inheritance, encapsulation, abstraction, polymorphism and dynamic binding.

4.1 Word Classification and Inheritance

Knowledge is sometimes classified into two categories: language-dependent category and concept-dependent category. Language-dependent knowledge represents information related to the specific language/languages such as whether a given stem is a noun or a verb and some other language characteristics. Such information may be kept in a lexicon, monolingual dictionary, bilingual dictionary or multilingual dictionary, depending on the specific application, languages in use and translation paradigm. In the other hand, concept-dependent knowledge is mainly the representation of concepts of the domain of a machine translation system. Concepts in the world are the same irrespective of the language. Some concepts may slightly vary in representation and semantics due differences in cultures.
In Khaeer, both categories of knowledge can be represented by objects (classes, subclasses and instances of these classes) with multiple inheritance features. Khaeer allows developers to set different facets to describe various features of a slot in a defined class. Some of these facets are: default value, cardinality, storage, access, inheritance propagation and source facets.

![Diagram of Arabic stems classification](image.png)

**Figure 4.** The hierarchical classification of the Arabic stems.

**Figure 5.** Examples of Arabic words classification.
Figure 4 shows the suggested hierarchical classification of the Arabic stems. The corresponding Khabeer implementation is shown in Figure 5. In this Figure, other slots are added to the defined classes to describe Arabic features of these stems. Describing these features in an English-based tool would be difficult and artificial.

4.2 Affixes and dynamic binding

Part of the morphological analysis is the capability of removing prefixes and suffixes from the input tokens to form word stems. These affixes should be handled with the same manner irrespective of their different values. The knowledge representation should include these affixes with their demons. The demon is a small procedure (a message handler) attached to the class of these affixes. The demon works in the same manner with different values of the affixes. In Figure 6, a class of suffixes is defined. Sample of suffixes that include the pronouns are listed. A demon is attached to the suffixes class. The task of this demon is to check whether a word has one of these suffixes or not. The demon works with all instances of the class suffixes.

4.3 Verbs and their forms

Concepts may be presented by classes of verbs. Each class should contain the root of the verb, the molds of the verb, the type of the subject, object, cause, instrument, time period, place, etc. Figure 7 defines part of a hypothetical knowledge. The top class of any verb is defined to include several common slots. The second level is defining subclasses to categorize the verbs semantically. Four examples of concept categories are coded: mental verbs (e.g. فكر, نكر), spoken verbs (e.g. قال, تكلم), action verbs (e.g. حمل, جاء) and feeling verbs (e.g. حسن).
4.4 Pronouns their polymorphism

In addition to the verbs, nouns and articles, the knowledge base contains pronouns. These pronouns may be used in sentence generation to transform a sentence from the deep structure into the surface structure. To handle these transformations, some pronouns information is needed. In Figure 8, a class of pronouns is defined. Then samples of possessive pronouns are defined with their information.

5. Sentence Generation

In sentence generation, at least four steps are needed [10]. The first step is deep content...
determination which determines the information needed to be communicated. The second step is sentence planning which is concerned with defining a skeleton or an abstract for the sentence and the text which will be used. The third step is surface realization where the order of words and syntactic structure is generalized from the output of the previous step. The fourth step is morphology and post-processing where actual inflected words (actual surface structure) are produced. By these four steps sentences are generated from the deep structures (internal representation) into the surface structures. The generation follows grammar rules similar to the grammar rules in the syntactical analysis. Sentence generation also utilizes the information in the knowledge base and its demons to form the proper target sentences. Figure 9 shows an instance of a verb that reflects the deep structure of a sentence and the corresponding Arabic sentence.

Figure 9. Class instance and the corresponding Arabic Sentences.

6. Conclusion

The material presented in this paper is a demonstration of using Khabeer expert system shell as a tool in machine translation systems. Several simple examples were introduced to show the power of Khabeer as an implementation tool for different phases of a machine translation system. Although these examples were tested under Khabeer, they are from representing a complete translation system.

Many string functions are supported by Khabeer to simplify lexical and morphological analysis and generation. The nature of Khabeer as a production system allows writing syntactical transformation rules directly. Object oriented features supported by Khabeer including inheritance, encapsulation, abstraction, polymorphism and dynamic binding helps a lot in designing and implementing a general knowledge base. Khabeer, running under Microsoft Windows environment, will be soon a freeware product for interested researchers.

ACKNOWLEDGMENT

The Authors wish to acknowledge KFUPM for utilizing the various facilities in preparation and presentation of this paper.

REFERENCES


