Lexicon Design Using a Paradigmatic Approach

Cristian Dumitrescu
Research Institute for Informatics
Alex. Averescu Blvd. 8-10,
71316 Bucharest 1, Romania, Fax: 653095

Abstract

The paper describes models for representation and methods to handle lexicographic structures supplied by the *MORPHO-2* system. It was built to manage monolingual lexicons and to incorporate lexical processing.

1 Introduction

Most advanced systems for natural language processing use powerful lexicons and morpho-lexical processing environments.

The system described below enables monolingual lexicon handling and incorporates morpho-lexical processes (i.e. word-form analysis and synthesis) at lexicon level. At present, it works as a component of the *IURES* environment for building natural language applications (Tufis and Cristea, 1985).

Since in our approach the morphological processes obey a paradigmatic morphology (Tufis, 1989), word-forms analysis and synthesis take into account only grammatical endings (which includes both desinences and suffixes) and the lexicons handled by *MORPHO-2* system are root- or lemma-oriented. By lexicon we don't mean only a collection of roots and associated features. At lexicon level we also encounter the control structures needed for morphological processing: morpho-lexical acquisition menus, root modification rules, word-forms synthesis rules, etc.

The services provided by the system may be classified according to the following goals: morphological model design, lexical stock building and morpho-lexical processing(Dumitrescu, 1991).

2 Morphological Model Design

In order to build the morphological model, an integrated environment which allows editing, viewing and compiling the morphological model description, is available to the linguist.

Defining the morphological model takes place in several steps, during which the linguist has to specify the following:

a) the categories, subcategories, features and their values, in a hierarchical manner
b) the paradigmatic descriptions
c) the default feature specifications associated to each paradigmatic description
d) the lemma - entry correspondence, for each paradigmatic description
e) the inflectional paradigms and root detection rules.

The hierarchical description of features is achieved by correlating several feature specifications. A feature specification is given in the form of a (feature: value$^+$) pair. We call a paradigmatic description a hierarchical description build of several simple (feature: value) pairs.

Figure 1 partially presents, in the form of an incomplete tree, the hierarchical description of features from the morphological model for the Romanian language. By tree traversal, all paradigmatic descriptions of the model may by generated.

Each non-terminal node contains a single feature specification. The leaf nodes may contain one or more feature specifications. According to the successor selection criteria, which is applied when visiting a non-terminal node, we can distinguish

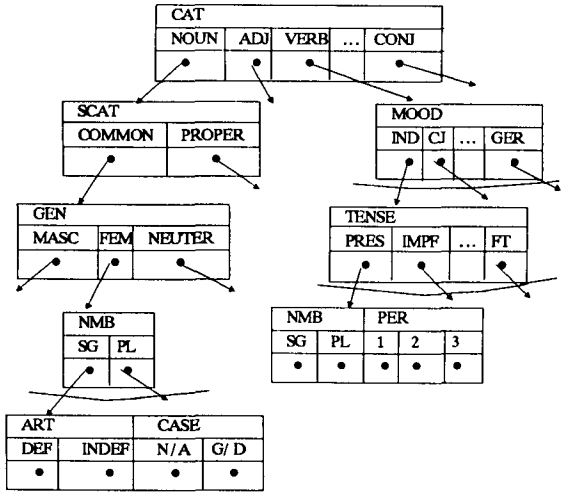

Figure 1 Hierarchical description of features

CHOOSE nodes(when only one successor is selected) or FOREACH nodes (when the individual selection of each successor is required). In the figure, a FOREACH node is outlined by a curve drawn over the emerging edges. By traversing the tree across the longest path which starts from the root node, thru CHOOSE nodes only, the selector of a paradigmatic description is obtained (e.g. CAT = NOUN & SCAT = COMMON & GEN = FEM, CAT = VERB).

The description attached to a leaf node is represented by means of a morpho-lexical acquisition scenario. A scenario entry (further on referred to as a slot) corresponds to a point of the paradigmatic description space.

Selectors of those descriptions allowing default feature specifications are attached with (feature: value$^+$) pairs which are default inheritances of the corresponding slots. In our example the following association is possible: (CAT = VB) -> (PER 1 2 3).

The area of the morphological model where the lemma - entry (from paradigmatic description) correspondences are described, consists in a specification of the points from the paradigmatic description spaces, which characterize the lemma field from the lexicon entry. This way, the lexical level required by the lexical transfer is ensured.

The last step in the morphological model description is to inform the system about how to build inflexional paradigms and root detection rules. For each paradigmatic description the linguist may specify more paradigmatic ending families from which the system then builds the inflectional paradigms. For the Romanian language, there have been identified 136 inflectional paradigms (Tufis, 1989).

Based on the inflectional paradigms, the system will determine the rules for root detection and word-form generation.

Such a rule has the following form:

< inflexion > : = (< inflectional-paradigm > < slot-number >)

with the following meanings:

a) if a word ends in < inflexion > then

- the root is what remains from the word after dropping the < inflexion >
- the root belongs to the < inflectional-paradigm >
- the contextual information corresponding to the current word is given by < slot-number >

b) if a root belongs to the < inflectional-paradigm > and it is used in the context given by < slot -number > then
- the word is obtained by concatenating the given root with the < inflexion >.

The lexicographer's interface is strictly dependent on the specifications from the linguist's interface since a large part of the former is built automatically from the specifications of the latter.

3 Lexical Stock Building

MORPHO-2 lets the lexicographer define new entries in the lexicon by means of a user-friendly window oriented interface.

A lexicon entry has the following formal structure:

< entry > :: = (< lemma >
(< paradigmatic-description-selector >
< inflexional-paradigm >
(< root > < morphologic-description >)*
(< syntactic-description > < semantic-description > *) *) *)

The fields < lemma >, < paradigmatic-description-selector > and < inflexional-paradigm > have the obvious meaning.

The field < root > may contain one or more roots. Inserting roots in the lexicon takes place in such a way that these should inherit the morphological descriptions belonging to the slots where they occur.

By < syntactic-description > we refer to restrictions on co-occurrence with other words (or phrases). In order to specify such restrictions for the Romanian language we have performed a subcategorization of verbs based on their valency, object categories which they govern(e.g., a direct object may be an accusative noun without preposition, a reflexive pronoun or a non-finite form of a verb) and semantic features.The latter allow a noncontextual subcategorization (for example of nouns) and a contexual one using selectional restrictions (in the case of verbs).

Typically, verbs have a valency between 1 and 3 (though impersonal verbs may have valency 0). The intransitive verbs are classified according to semantic criteria (verbs of motion, state) or by their syntactic usage (like predicative auxiliaries, unipersonal verbs with dative). We should notice that the same verb may be transitive or intransitive, according to its meanings; for example *a ajunge* (to get to) with the meaning *a prinde* (to catch) is transitive and with the meaning *a fi suficient* (to be enough) is intransitive.

Trivalent verbs include verbs taking:
- two direct objects which have different meanings and are not coordinated (the first one is doubled by an accusative, personal pronoun).
  Pe Ion l-am ascultat *lectia*.
  I examined **Ion** about *the lesson*.
- a direct object and an object clause
  L-am rugat *sa-mi imprumute pixul*.
  I asked **him** *to lend me the pen*.
- a direct object and an indirect object
  L-am intrebat *despre carte*.
  I asked **him** *about the book*.

For each syntactic description, the lexicographer may provide one or more semantic descriptions. The < semantic-description > field contains the name of a case-frame structure placed in a generic-specific hierarchy. The actual semantic descriptions are stored in a separate data area, than the rest of the lexicon, and they are managed independently of *MORPHO-2*.

A lexicon editor offers the lexicographer commands for deleting, modifying a lexicon entry and their listing according to different requests with respect to entry fields (Dumitrescu,1991).

4 Morpho-Lexical Processing

The target natural language processing system is the beneficiary of the morpho-lexical processes executed by *MORPHO-2*. Word-forms analysis and synthesis are mediated by a process interface.

In the case of lexical analysis, if the interface is given a sequence of words, it will return a sequence of morpho-lexical atoms. The structure of these atoms is presented below:

(< root > (< lemma >
(< paradigmatic-description-selector >
< morphologic-description >
(< syntactic-description > < semantic-description > *) *) *)

A morphological description contains both contextual and context-free information. The former is obtained from ending analysis and the latter from the lexicon entry corresponding to the root. The information for the other fields from the atom structure is also taken from the lexicon entry corresponding to the root.

With respect to the result of morphological congruence and root retrieval within the lexicon, we may classify the morpho-lexical atoms as unambiguous, ambiguous and undetermined.

The unambiguous morpho-lexical atoms associate the analyzed word with a single lemma. In the case of a root which corresponds to one lemma and has more possible morphological descriptions, for the same paradigmatic description selector, the system will attempt to compact them.

The ambiguous morpho-lexical atoms come from words to which several lemmae may be attached. The association of a root with several lemmae is possible either due to ambiguity of category (e.g. noun vs. verb) or to apparent homography, generated by the absence of prosodic markers in the Romanian language (módele, modéle, acéle, ácele, modúl, módul, etc.). The possible interpretations are ordered in such way that those which come from shorter roots (that means longer ending) have priority.

The undetermined morpho-lexical atoms correspond to words which have no entry in the lexicon. The atoms generated in this situation have the following structure:

(UNKNOWN < unknown-word >
(< possible-root > < morphologic-description > *))

The unknown word is associated with all legal segmentations and for each of them the morphological information deduced from the identified endings is provided.

Lexical synthesis is the reverse of lexical analysis. The process interface ensures conversion of a morpho-lexical atom sequence into a word sequence. The morpho-lexical synthesis requires the description of morpho-lexical atoms according to the pattern:

(< entry-identifier > < morphologic-description >
< syntactic-description >)

where < entry-identifier > may be a lemma, a root or a semantic description.

We have to point out that previous to morpho-lexical analysis and synthesis, the target processor may configure the structure of morpho-lexical atoms according to the designed application, by means of a communication protocol.

5 Implementation

The *MORPHO* project, started in 1986, has achieved as a first result, a prototype version now available on a PDP-11 compatible computer. The second version of the system, the one presented in this paper, is implemented in C on a IBM-PC compatible.

The network representation of data and techniques used for implementation, like lexicon indexing using prefixed virtual B^+ trees (based on which, for 20000 pseudo-random generated words, of variable length, retrieval requires 2 external accesses only), have led to an average response time of lexical processes, quite independent of the lexicons size (for more details on performance analysis see (Tufis and Dumitrescu,1990)).

References

Dumitrescu, C. MORPHO user manual. I.C.I., Bucharest, 1991

Tufis, D., Cristea D. IURES: A human engineering approach to natural language question answering. Artificial Intelligence: Methodology, Systems, Applications, North Holland, Elsevier, 1985

Tufis, D. It would be much easier if WENT were GOED. Proceedings of the 4th Conference of ECACL, Manchester, 1989

Tufis, D., Dumitrescu, C.: MORPHO - A dictionary management system. Proceedings of the 13th International Seminar on DBMS, Mamaia, 1990