Generation of Turkish Noun and Verbal Groups with Systemic-Functional Grammar[1]

Ilyas Cicekli
Dept. of Comp. Eng. and Info. Sc.
Bilkent University,
06533 Bilkent, Ankara, Turkey
`ilyas@cs.bilkent.edu.tr`

Turgay Korkmaz[2]
Dept. of Comp. and Info. Sc.
Syracuse University,
Syracuse, NY 13244, USA
`tkorkmaz@mailbox.syr.edu`

Abstract

This paper mainly presents noun and verbal group parts of a Turkish sentence generator to produce the actual text from its semantic description. In order to concentrate on text generation rather than text planning, we assume that the lexicalized semantic description of the text is produced in some way, currently given by hand. In the generation, we need a linguistic theory to describe linguistic resources, and also a software tool to perform them in a computational environment. We use a *functional linguistic theory* called Systemic–Functional Grammar (SFG) to represent linguistic resources, and FUF text generation system as a software tool to perform them. In this paper, we present the systemic–functional representation and the realization of noun and verbal groups in Turkish sentences.

1 Introduction

Natural language generation is a kind of process that encodes the mental picture of reality into a sequence of words called grammatical units such as clause, verbal group, noun group etc. The units of a grammar can be ordered in terms of a *rank scale*, from the largest to the smallest unit (*structural classification*) [6]: a *sentence* consists of one or more clauses; a *clause* consists of one or more phrases (groups); a *phrase* consists of one or more words; a *word* is the smallest unit.

A *simple sentence* consists of only one main process and several components that complement or modify the main process. Each component may be realized by complex syntactic structures but it does not change the simple structure of the sentence. In other words, the number of words in a sentence does not determine whether the sentence is simple or not. The main property of the simple sentence is that each component in the sentence has a function that is determined by the main process such as actor, goal, time, manner, etc.

A simple sentence consists of the following three semantic functions that draw the mental picture of reality (*semantic representation*) [6]: *process* is a general term to represent an event or a state; *participants* are persons or things involved in a process; *circumstances* give further information—time, place, manner etc. about a process. Generally, *participants* are realized by noun groups that are considered in more detail in Section 3, and *circumstances* are realized by noun groups, adverb groups or post-positional groups in Turkish. A *process* is realized by a verbal group that is considered in more detail in Section 2.

We analyze simple Turkish sentences from the systemic-functional perspective to determine their structural and functional descriptions [12]. By using these descriptions, we have constructed the system network of simple sentences and we have implemented a sentence generator in Functional Unification Formalism (FUF) [3] to perform the linguistic resources. In our analysis, we determine the main process, participants and circumstantials of a sentence, and how they are realized in Turkish. In this paper, we concentrate on the analysis of Turkish noun and verbal groups.

The remainder of this paper is organized as follows. In Section 2 and Section 3, we consider the grammatical analysis of noun and verbal groups, and their realizations in Turkish. Section 4 presents the implementation of a Turkish sentence generator, and demonstrates the generation of noun and verbal groups. Finally, Section 5 presents the conclusion and future work.

2 Verbal Groups

Verbal groups are used to realize *processes* of nominal and verbal sentences. A verbal group is constructed on a lexical element called **base** that can be a *verb* or a *nominal group*. The **base** is the single lexical element that is given for the formation of a verbal group. Other lexical elements (such as `değil` (neg. noun), `mi` (question), `ol` (be)), relevant suffixes and components of the verbal group are determined and organized by the systemic-functional grammar designed for Turkish to express appropriate meanings. This section presents the possible structures of verbal groups and their internal organization in Turkish [1, 9].

There are more than one grammatical structure of verbal groups to express many distinct meanings. Fortunately, they may be generalized according to the type of **base** (*nominal group, verb*) and the **mood** (*finite, non-finite*). The selected features from these two systems (*type-of-base* and *mood*) determine the appropriate structure for the verbal group. The selected features from other systems organize the internal structure of the verbal group. As a result, the following four general structures can occur:

[1] This work was supported by NATO Science for stability Project Grant TU-LANGUAGE.

[2] This work was done when the second author at Bilkent University

1. **base** is a *verb* and **mood** is *finite*:
   This case is selected to realize the *process* of a verbal sentence, or question. There exist two distinct components of the verbal group for interrogative sentences (questions): **base** and **interrogative tag**. The *mode, person,* and *number* are added to **base** or **interrogative tag** depending on the selected values of these functions.

(1) a. Gerçek-ler-i *gör-ebil-meli-sin.*
fact+PL+ACC see+POT+NEC+2SG
'*You ought to be able to see* the facts.'

b. Masa-yı *boya-t-acak* *mı-ydı-n*?
table+ACC paint+CAUS+FUT Ques+PAST+2SG
'*Were you going to have* the table *painted*?'

2. **base** is a *verb* and **mood** is *non-finite*:
   The structure of finite verbal group of a verbal sentence can be used in this case by replacing the *finite* with a *non-finite* element. A non-finite verbal group realizes the *process* of a clause that may be used as a noun (infinitive), adjective (participle) or adverb (adverbial).

(2) a. *beğen-il-mek* güzel-dir.
appreciate+PASS+INF* nice+COP
'It is nice *to be appreciated.*'

b. Masa-yı *boya-yacak* adam gel-me-di.
table+ACC paint+PART* man come+NEG+PAST
'The man *who would paint* the table did not come.'

3. **base** is a *nominal group* and **mood** is *finite*:
   This case is selected to realize the *relational processes* that express the way of "being". Here, the **base** is a nominal group that may be an attribute or an identifier in a nominal sentence or question. The type of "being" may be intensive, circumstantial, or possessive. According to its type, the **base** may take some suffixes such as locative and possessive before the formation of the verbal group. In the generation of a verbal group, we assume that the **base** is a lexical element, and the required suffixes or the distinct elements are determined by the systemic grammar to express the appropriate meanings. This case involves two types of grammatical structures, and one of them is selected to realize a relational process by depending on the value of the *time*. If the value of *time* is present, past, narrative or conditional then the first case is selected, otherwise the second one is selected. In the first case, a *substantive* verb is attached to **base**. In addition, a distinct element called **neg-noun** is located after **base** to express the negative meaning, and relevant suffixes are attached to this negative noun. In the second case, an auxiliary verb `ol` (be) appears as a separate element after the **base**.

(3) a. Ali *mühendis-dir.*
Ali engineer+COP
'Ali *is an engineer.*'

b. Ali *mühendis değil-dir.*
Ali engineer not+COP
'Ali *is not an engineer.*'

c. Ali *mühendis ol-ma-yacak-tı.*
Ali engineer be+NEG+FUT+PAST
'Ali *was not going to be an engineer.*'

4. **base** is a *nominal group* and **mood** is *non-finite*:
   The second case of previous construct is used by changing the value of the **mood** of auxiliary verb (`ol`) to *non-finite.*

(4) a. mühendis ol-mak
engineer be+INF*
'*to be an engineer*'

b. mühendis ol-an
engineer be+PART*
'*(someone) who is an engineer*'

c. mühendis ol-arak
engineer be+ADV*
'*as an engineer.*'

3 Noun Groups (NP)

A noun group (noun phrase-NP) is a grammatical unit that contains at least one noun called *head*, and it may contain modifiers that modify that head noun. The modifiers express various types of information about the head noun, but do not change its semantic features. Indeed, the semantic features of the head noun also belong to the entire NP. Thus, the NP can be interpreted as an expansion of the head noun.

The head noun can be a common noun, a proper noun, or a pronoun. According to this choice, the head noun is modified by different grammatical functions that may be interpreted as the constituents of the NP. The general grammatical functions that expand the head noun can be:

- *determiner* which indicates whether the subset of the head noun is specific or not.
- *numerative* which expresses the numerical features of the head noun.
- *describer* which indicates the subjective and objective properties of the head noun.
- *classifier* which indicates a particular subclass of the head noun.
- *qualifier* which indicates the characteristics of the head noun in terms of some process in which the head noun is directly or indirectly involved as a participant.
- *post-modifier* which indicates a particular subset of the head noun.

These grammatical functions can be divided into more specific sub-functions. The order of these functions in a Turkish noun group is determined by partial orders among them in the implementation.

3.1 Determiner

Determiners can be decomposed into the following sub-functions: *possessor*, *demonstrative*, and *ns-deictic*. These sub-functions are also known as deictic elements [6] which indicate whether some specific subset of the head noun is intended or not. *Possessor* and *demonstrative* are specific deictics, and *ns-specific* represents non-specific deictic elements. The default order of these functions in Turkish noun groups is as follows: $possessor \quad demonstrative \quad ns - deictic$.

- The *possessor* function indicates the owner of the head noun, and it is realized by a noun phrase that is marked with genitive case. In addition, the agreement of the possessor and the possessive case mark of the head noun must be same, and this is enforced by the designed Turkish grammar. Since the information expressed by the possessor can also be given by the possessive case mark of the head noun, the explicit possessor may be dropped from the noun group if it is not emphasized. *Possessor* specifies the head noun in two mutually exclusive types: *determinative* or *interrogative*.
- The *demonstrative* function indicates a particular subset of the head noun according to the distance between the speaker and the specified thing, and it is realized by a demonstrative adjective. If distance is *near, far,* or *too-far*, the demonstrative will be `bu` (this), `şu` (that), or `o` (that), respectively. In contrast to English, there is no plural form of demonstrative adjectives in Turkish. *Demonstrative* elements also specify the head noun in two mutually exclusive types: *determinative* or *interrogative*.
- The *ns-deictic* function indicates the sense of *all*, *none* or *some* of the unspecified subset of the head noun. They are realized by non-specific determiners in Turkish such as `her` (every), `bütün` (all), `biraz` (a little), `bazı` (some), `birçok` (many), and the indefinite article `bir` (a/an). Some *ns-deictic* elements can only specify the head nouns that are morphologically marked plural, and some can only specify the singular ones. The countability of the head noun also plays a role in the selection of the *ns-deictic* determiner.

3.2 Numerative

The *numerative* sub-functions *ordinal* and *quantitative* represent numerative elements which indicate some numerical features of the subset of the head noun. The *numerative* elements tend to follow *determiners* in Turkish noun groups, and the *ordinal* sub-function tends to precede *quantitative* sub-function.

- The *ordinal* function indicates the place of the head noun in order, and it is realized by ordinative adjectives. It expresses either an exact place or an inexact place. Some of the Turkish ordinative adjectives which indicate an exact place are: `ilk` (first), `birinci` (first), `ikinci` (second), etc., and `sonuncu` (last). Although `ilk` and `birinci` seem to have the same semantic meaning, they have a slightly different behavior. In Turkish, ordinals `birinci` and `sonuncu` can take the adverbial intensifier `en` (most), the other ordinals cannot take the adverbial intensifier including `ilk`. Some of the ordinative adjectives which indicate an inexact place are: `önceki` (previous) and `sonraki` (next).
- A *quantitative* element expresses the numerical features of the thing denoted by the head noun. A *quantitative* element can be *exact*, *indefinite* or *fuzzy*. An *exact quantitative* element can be: *a cardinal*, *a distributive* element, *a fraction*, or *a range*. An *indefinite quantitative* element is an adjective that expresses an indefinite quantity such as `az` (little) and `çok` (lot). A *fuzzy quantitative* element consists of a phrase indicating a limit (lower, upper, or both) or an approximate quantity, and an *exact quantitative* element. If a *quantitative* element is *exact* or *fuzzy*, it can be followed by a *measure* noun which can be: `adet, tane, parça` (all of them mean 'piece'); a unit of length, area, weight, etc.; a container. If the *measure* noun is a container, it can be followed by `dolusu` (full of) to indicate that the container is full. If a *quantitative* element is *indefinite*, it can be followed by the measure noun `miktarda` (amount). Although a noun group is semantically is plural when a *quantitative* element indicates a quantity more than one, the head noun in that noun group morphologically is not marked as plural.

3.3 Describer

The *attitude* sub-function of the *describer* function expresses the subjective property of the head noun. This sub-function tends to precede *numeratives* in the default order of Turkish noun groups. On the other hand, the *quality* sub-functions (*age, size, and color*) which express the objective properties of the head noun tend to follow *numeratives*. Describers are realized by adjectives, and they may accept the following degrees of comparison: similarity, comparative, superlative.

3.4 Classifier

The *classifier* element represents a particular subclass of the head noun, and it is realized by a noun group. It immediately precedes the head noun and the head noun takes third person singular possessive case mark.

3.5 Qualifier

The *qualifier* function is realized by prepositional phrases or clauses, and they are located after the head noun in English. However, the order of qualifiers in Turkish NPs is determined according to the function of the qualifier. If it specifies the head noun, it is located at the beginning of the NP. If it describes the head noun, it is located between the determiner and the describer. The qualifiers in Turkish NPs are realized by noun phrases, adverb groups, or non-finite clauses. The sub-functions of the *qualifier* function are as follows:

- The *qual-set* element expresses a superset of the head noun, and it is realized by an NP which is semantically marked plural and in ablative case.
- The *qual-descriptor* element expresses some properties of the head noun in a process, and it is realized by a participle clause.
- The *qual-location* element expresses the spatial or temporal location of the head noun. It may be realized by an NP or an adverbial group that takes the `+ki` relative suffix. In addition, if it is realized by an NP, it must be in locative case.
- The *qual-comparison* element expresses the properties of the head noun in comparison with another thing. It is realized by an NP followed by a postposition (a postpositional group). Some of these postpositions are `kadar` (as much/many as), `ait` (belongs to), and `gibi` (like). The case of the NP depends on the postposition used.
- The *qual-madeof* element indicates what the head noun is made of, and it is realized by a noun phrase in ablative case.
- The *qual-madeon* element indicates what the head noun is made on, and it is realized by a noun phrase in locative case.
- The *qual-possession* element expresses what belongs or what does not belong to the head noun. To explain the positive or negative possession, it is realized by an NP which is in munitive or privative case, respectively.

3.6 Post-Modifier

The *post-modifier* function expresses a particular subset of the head noun. Most of the functions discussed above can also be used as a *post-modifier* element. The *post-modifier* element immediately follows the head noun, and it takes the third personal possessive case mark. The head noun is realized in genitive case.

4 Implementation

A system network is a set of systems such that each system is described as "a set of linguistic choices in a specific linguistic context" by Firth [12]. In addition, the system network displays the graphical organization of the grammar. In the generation with SFG, the system network is traversed from left to right by selecting a feature from each system, and executing the realization rules attached to this feature [10, 12]. If the selected feature has a function that is realized by one of the grammatical units in the rank scale, the systemic network is re-entered, and recursively traversed for the generation of that unit. After traversing the entire system network, generations of the grammatical units are completed. In this way, the whole text that consists of these grammatical units is generated. The required systems, the realization rules, and the appropriate context of each system in the linguistic description for noun and verbal groups are determined and organized by using the analysis described in the previous sections.

In order to develop a text generator with the systemic-functional grammar, we need to implement the linguistic descriptions (system networks and realization rules) in a computational environment. For this purpose, we use the FUF text generation system[4] including its functional unification grammar (FUG) and typed feature formalisms. In this section, we present a brief overview of the generation in FUF, and then, we particularly consider the generation of noun and verbal groups.

The FUF text generation system consists of two main modules: a *unifier* and a *linearizer* [3]. The unifier takes, as input, a *lexicalized semantic description* of the text to be generated, and an *extended form of FUG*, and then produces as output a *rich syntactic description* of the text or some new inputs[3] (the semantic and syntactic descriptions) for the grammatical units that realize the specific components of the text [4]. After the unification process, the linearizer takes the generated syntactic description as input, and then produces the morphological description of the text. The morphology unit produces the worded text by using this morphological description [11]. We assume that an application program that is not included in our implementation produces the *lexicalized semantic description* of the text.

In FUG framework, a data structure called functional description (FD) is handled. A FD is a list of pairs. Each pair has an attribute name and value. Since we use the FUG formalism in our implementation, we need to translate the system network into this formalism. A system in the system network can be translated into disjunction of FDs, where each FD corresponds to an alternative in that system [8]. Realization rules and relations between systems are also translated into attribute-value pairs. This process is described by Kasper as an algorithm that translates SFG into FUG [8]. In addition, FUF provides a typed feature formalism to implement the mutual exclusion, and hierarchical relations in SFG [2].

By using these formalisms, we have designed and implemented a single sentence generator with SFG. For this purpose, we have designed a Turkish grammar for simple sentences in the FUG formalism, and we have made the required changes in the linearizer of the FUF text generation system in order to handle Turkish morphology. Although we implemented a full sentence generator, we only present noun and verbal group part of this generator in this paper.

[3] These new inputs are produced and recursively performed by the unifier.

The lexicalized semantic representation of a noun or verbal group must contain the required functions for that group. If a function does not appear in the input set but it is required, a default value for that function is used. The following groups are generated by the system implemented in FUF:[4]

Example 1: şu ilginç beş eski kitap
that interesting five old book
'those old five interesting books'

Lexicalized Semantic Input:

```
((cat common) (lex "kitap") (specific yes)
 (determiner (demonstrative ((type determinative) (distance far))))
 (numerative ((quantitative ((exact yes) (quan-type cardinal) (value ((lex "beS")))))))
 (describer ((attitude ((lex "ilginC"))) (quality ((age ((lex "eski"))))))))
```

Output:

```
[[CAT=ADJ][ROOT=Su]]  [[CAT=ADJ][ROOT=ilginC]]  [[CAT=ADJ][ROOT=beS]]
[[CAT=ADJ][ROOT=eski]]  [[CAT=NOUN][ROOT=kitap][AGR=3SG][POSS=NONE][CASE=NOM]]
```

Example 2: gör-ebil-meli-sin
see+POT+NEC+2SG
'You ought to be able to see'

Lexicalized Semantic Input:

```
((cat verbal-group) (lex "gOr") (type-of-base verb) (polarity positive)
 (desc-verb potential) (desc-polarity positive) (mood finite) (time necessitative)
 (mode none) (person second) (number singular) (interrogative none))
```

Output:

```
[[CAT=VERB] [ROOT=gOr] [SENSE=POS] [COMP=YABIL] [TAM1=NECES] [AGR=2SG]]
```

5 Conclusion and Future Work

Our main purpose is to design and implement a Turkish sentence generation system by using the systemic-functional approach. To realize this system, we need to develop a large Turkish grammar based on systemic-functional theory, and to implement it in the computational environment. The grammar can be divided into small parts as shown in the rank scale. Then, each part may be developed independently. The most important part of the grammar is the noun and verbal groups that realize the several semantic functions of sentences. So, at the beginning, we have considered the most common grammatical structures of Turkish and their implementation in FUF. The other parts of the grammar such as complex sentences, and the overall generation system including an application program that maps interlingua representations of sentences onto their lexicalized semantic representations are currently under development.

References

[1] T. Banguoğlu, *Türkçenin Grameri*, Türk Tarih Kurumu Basım Evi, Ankara, 1986.

[2] M. Elhadad, Types in functional unification grammars, in: *Proc. of the ACL'90*, 1990, pp. 642-655.

[3] M. Elhadad, *Using Argumentation to Control Lexical Choice: A Functional Unification Based Approach*, Ph.D. Dissertation, Columbia University, New York, 1990.

[4] M. Elhadad, *FUF: The Universal Unifier User Manual 5.2*, Department of Computer Science, Ben Gurion University of the Negev, 1993.

[5] E.E. Erguvanlı, *The Function of Word Order in Turkish Grammar*, Ph.D. Dissertation, University of California, Los Angeles, 1979.

[6] M. A. K. Halliday, *An Introduction to Functional Grammar*, Edward Arnold, London, 1985.

[7] B. Hoffman, *The Computational Analysis of the Syntax and Interpretation of "Free" Word Order in Turkish*, Ph.D. Dissertation, University of Pennsylvania, Philadelphia, 1995.

[8] R.T. Kasper, Systemic Grammar and Functional Unification Grammar, in: *Systemic Functional Approaches to Discourse*, Ablex, 1988.

[9] N. Koç, *Yeni Dilbilgisi*, İnkilap Kitapevi, İstanbul, 1990.

[10] C. M. Matthiessen, and J. A. Bateman, *Text Generation and Systemic-Functional Linguistic*, Communication in Artificial Intelligence Series, Pinter Publishers, 1991.

[11] K. Oflazer, Two-level Description of Turkish Morphology, in: *Proc. of the Sixth Conference of the European Chapter of the Association for Computational Linguistic*, April 1993.

[12] T. Patten, *Systemic Text Generation as Problem Solving*, Studies in Natural Language Processing, Cambridge University Press, 1988.

[4] Extra Turkish letters are represented as follows: C is ç, I is ı, G is ğ, O is ö, S is ş, U is ü.