Generation of Turkish Verbal Groups with Systemic-Functional Grammar*

Turgay Korkmaz, and Ilyas Cicekli
Department of Computer Engineering and Information Science
Faculty of Engineering, Bilkent University, 06533 Bilkent, Ankara Turkey
E-mail: {turgay,ilyas}@cs.bilkent.edu.tr

Abstract. This paper mainly presents the verbal group part of a Turkish sentence generator that is currently under development for producing the actual text from its semantic description. To concentrate on the text generation rather than text planning, we assume that the semantic description of the text is produced in some way, currently given by hand. In the generation, we need a linguistic theory to describe the 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 the linguistic resources, and FUF text generation system as a software tool to perform them. In this paper, we present the systemic–functional representation and realization of Turkish verbal groups.

Keywords: Natural Language Generation, Computational Linguistic, Systemic-Functional Grammar, Functional Unification Grammar.

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. [10, 11]. The units of a grammar can be ordered in terms of a **rank scale**, from the *largest* to the *smallest* unit (*structural classification*) [5]: 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 the following three semantic functions that draw the mental picture of reality (*semantic representation*) [5]: **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. Each semantic function is realized by one of the grammatical units given above. Generally *participants* and *circumstances* are realized by noun groups and adverb groups, respectively. A *process* is realized

*This work was supported by NATO Science for stability Project Grant TU-LANGUAGE.

by a verbal group that will be considered in the following sections. In addition, the verbal group also realizes the *interpersonal functions* such as *mood, voice, tense, modality* etc.

The generation of a sentence depends on the generation of the smaller units in the rank scale given above. In this study, we begin the sentence generation with the generation of verbal groups because they realize the main components (*process* and some *interpersonal functions*) of the sentence, and the sentence is a configuration of participants and circumstances around those main components. Therefore, we analyze Turkish verbal groups from the systemic-functional perspective to determine their structural and functional descriptions.[1] By using these descriptions, we construct the system network of the verbal group, and then we implement a system in FUF to perform the linguistic resources.

The remainder of this paper is organized as follows. In Section 2, we consider the grammatical analysis of the verbal groups. Section 3 gives a brief overview of the systemic-functional grammar approach to text generation, and particularly presents the system network of the verbal group. Next, in Section 4, the implementation of a Turkish sentence generator is introduced, and then the generation of the verbal groups is demonstrated. Finally, Section 5 presents conclusion and future work.

2 Grammatical Analysis of Verbal Groups

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. The other lexical elements such as `değil`, `mi`, `ol` and the relevant suffixes, the components of the verbal group, are determined and organized by the systemic-functional grammar to express appropriate meanings. So, this section presents the possible structures of the verbal groups and their internal organization in Turkish [1, 7].

There are more than one grammatical structures of the 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 the other systems in Figure 2 (given in Section 3) organize the internal structure of the verbal group. As a result, the following general structures can occur:[2]

• if **base** is a *verb* and **mood** is *finite*
This case is selected to realize the *process* of a verbal sentence, or question. The type of the process can be material or mental. The structure of verbal groups for this case is shown in Table 1.[3] 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

[1]These two descriptions are complementary in the SFG [13]: the functional description says *"What it does,"* and the structural description says *"How it does it."*

[2]The structures are considered in the tabular forms. The center row of the table describes the required functional elements of the verbal group in a grammatical order. The top rows of the table give examples, and bottom rows present their grammatical values, respectively.

[3]M-P-N stands for Mode, Person, and Number; DP stands for Descriptive Polarity

base or **interrogative tag** depending on the selected values of these functions.

Table 1: The Structure of Finite Verbal Group from Verb

sev yaz	-dır		-ebil		-melisin -acak	mı	-y-dı
Base	Voice Frame	Polarity	Desc-Verb	DP	Finite	Interr-Tag	M-P-N
Verb Verb	... ...	Pos Pos	Potential none	Pos Pos	... ...	none yes-no	...

	-dır	
Subj-Obj-rel	Transition	Voice
none none	none Trans 1	Active Active

Voice Frame

-meli -acak	-dı	-sin	
Time	Mode	Person	Number
Necess Future	none Past	Second Third	Sing Sing

Finite

(1) a. Arkadaşlarını *sevebilmelisin.*
friend+3PL+2PP+ACC love+POT+NEC+2SG
'You *ought to be able to love* your friends.'

b. Ali mektubu *yazdıracak mıydı*?
Ali letter+3SG+ACC write+CAUS+FUT Ques+PAST+3SG
'*Was* Ali *going to have* the letter *written*?'

• if **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). As a result, the structure for this case is given in Table 2.

Table 2: The Structure of Non-Finite Verbal Group from verb

Sev Oku Koş	-il				-mek -yacak -arak
Base	Voice Frame	Polarity	Desc-Verb	DP	Non-Finite
Verb Verb Verb	... ...	Pos Pos Pos	none none none		infinitive participle adverbial

(2) a. Birisi tarafından *sevilmek* güzeldir.
someone by love+PASS+CONV=NOUN nice+COP+AOR+3SG
'*To be loved* by someone is nice.'

b. Mektupu *okuyacak* adam gelmedi.
letter+3SG+ACC read+CONV=ADJ man come+NEG+PAST+3SG
'The man *who will read* the letter did not come.'

c. Ali okula *koşarak* gitti.
Ali school+DAT run+CONV=ADV go+PAST+3SG
'Ali went to school *by running*.'

• if **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 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. One of them is selected to realize a relational process by depending on the value of the *Time*. In the first structure shown in Table 3, a *substantive (predicative)* verb like an auxiliary verb is attached to **base** to demonstrate the "to be" meaning of the *process*. In addition, a distinct element called **neg-noun** is located after **base** to express the negative meaning. In the second structure shown in Table 4, an auxiliary verb "`olmak`" appears as a separate element after the **base**. If the value of *Time* is *Aorist, Past, Narr,* or *Cond* then the first structure is selected, otherwise the second one is selected.

Table 3: The Structure of Finite Verbal Group from Nominal group (1)

Öğretmen	-dir
Öğretmen	
Öğretmen	
Base	Finite
Noun	substantive
Noun	
Noun	

değil	-dir
Neg-Noun(polarity)	Finite
Pos	
Neg	subst...
Pos	

mi	-dir
Interr-Tag	Finite
none	
none	
yes-no	subst...

(3) a. O bir *öğretmendir*.
He a teacher+COP+AOR+3SG
'He *is a teacher*.'

b. O bir *öğretmen değildir*.
He a teacher not+COP+AOR+3SG
'He *is not a teacher*.'

c. O bir *öğretmen midir*?
He a teacher Ques+COP+AOR+3SG
'*Is* he a *teacher*?'

Table 4: The Structures of Finite Verbal Group Nominal group (2)

Öğretmen
Base
Noun

olmayacaktı
Aux::verbal-group, mood::finite
...

(4) Ali *öğretmen olmayacaktı.*
Ali teacher be+NEG+FUT+PAST+3SG
'Ali *was not going to be a teacher.*'

• if **base** is a *nominal group* and **mood** is *non-finite*
In this case, the same structure in Table 4 is used by changing the value of the **mood** of auxiliary verb to *non-finite.*

(5) yazar olmak/ yazar olan/ yazar olarak.
writer be+CONV=NOUN/ writer be+CONV=ADJ/ writer be+CONV=ADV
'*to be a writer/ (someone) who is a writer/ as a writer.*'

3 System Network of Verbal Group

A system network is a set of systems such that each of these systems is described as "a set of linguistic choices in a specific linguistic context" by Firth [13]. In addition, the system network displays the graphical organization of the grammar. In the generation with SFG, the system network (shown in Figure 1) is traversed from left to right by selecting a feature from each system, and executing the realization rules attached to this feature [9, 13]. If

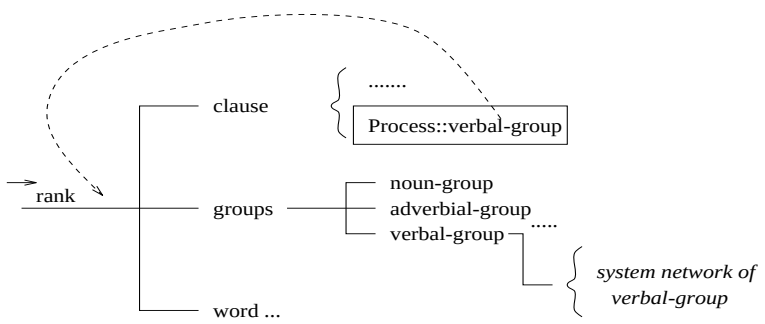

Figure 1: A System Network for the Sentence Generation

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 sentence that consists of these grammatical units is generated. In Figure 1, if we select the *clause* feature from the *rank* system, SFG introduces the *process* as a function of the clause, and then realizes it as a verbal group by re-entering the network. The selection of a feature from each system, and the representation

of realization rules depend on the implementation formalism. These issues will be considered in Section 4.

The required systems, the realization rules, and the appropriate context of each system in the linguistic description of the verbal group are determined and organized by using the analysis in the previous section. As a result, the system network given in Figure 2 is constructed. In the network, only the systems and their appropriate contexts are displayed to express

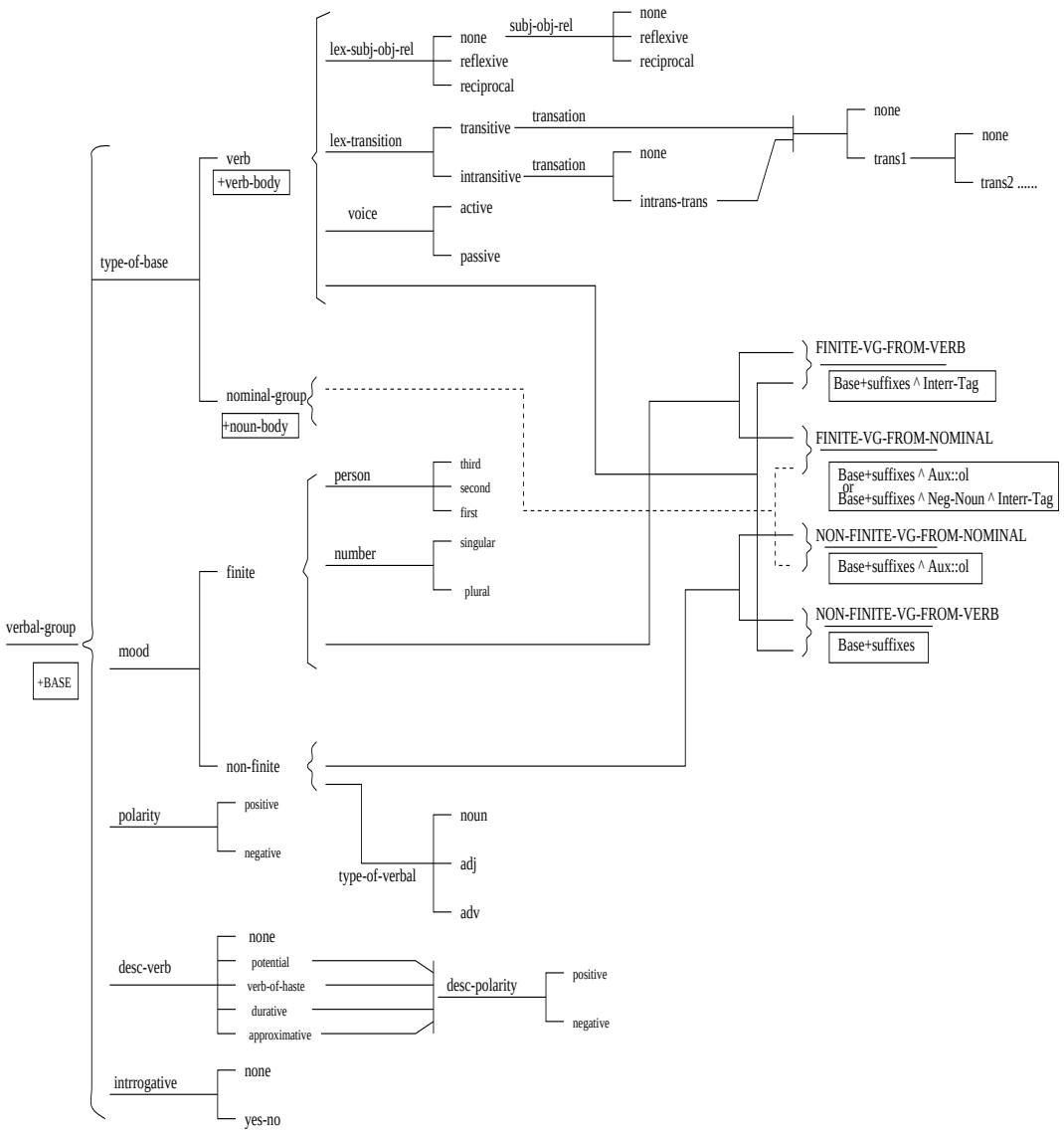

Figure 2: A system Network for Verbal Groups in Turkish

the basic linguistic description of the verbal groups. Because of this simplification, more specific rules and relations are not displayed in the network. However, they are considered and handled in the implementation.

To generate a verbal group, the system network is traversed by using the algorithm given

above. For example, to produce the verbal group "`sevebilmelisin`", the following systems are entered and the appropriate features are selected: Enter type-of-base, select verb; enter mood, select finite; enter polarity, select positive; enter desc-verb, select potential; enter interrogative, select none. According to these selected features, the other systems are entered and so on. At the end, the system `FINITE-VG-FROM-VERB` is entered to realize the verbal group by using the given structure in Table 1.

4 Implementation

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, and its functional unification (FUG) and typed feature formalisms. In this section, we will present a brief overview of the generation in FUF, and then, we will particularly consider the generation of 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[4] (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 [12]. By the way, we assume that an application program that is not included in our implementation produces the *semantic description* of the text. Consequently, the final text generation system can be organized as shown in Figure 3.

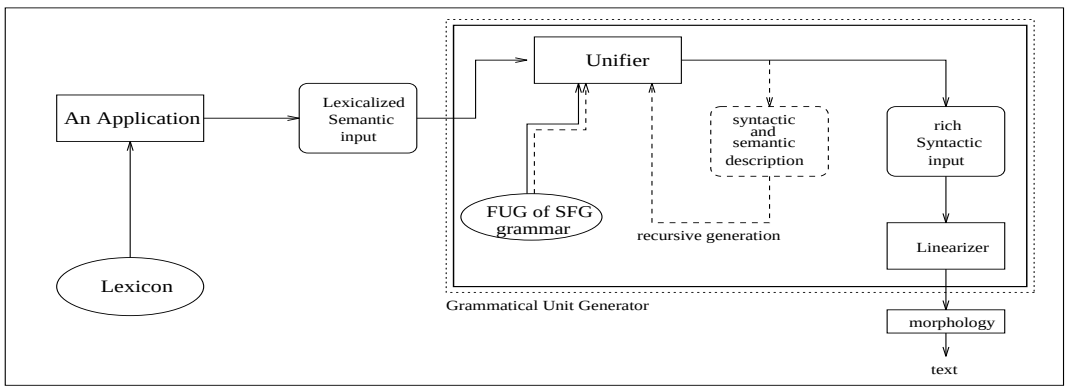

Figure 3: The Architecture of the Text Generator

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. In the implementation, we use

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

the FUG formalism. So, we need to translate the system network into this formalism that is extended in FUF. A system of the system network can be translated into disjunction of FDs, where each FD corresponds to an alternative in that system [6, 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 [6]. In addition, FUF provides a typed feature formalism to implement the mutual exclusion, and hierarchical relations in SFG [2].

By using these formalisms, we are currently trying to design and implement a single sentence generator with SFG. In the generation of a sentence, the generator produces a linguistic description (LD) for the verbal group to realize process, voice, tense etc., and then recursively transforms this LD into a worded text. In this partial implementation, we use the LD of the verbal group. So, we consider what it is rather than how it is produced from the semantic representation of the sentence. The required functions and their possible values for the linguistic description of the verbal group are presented in Table 5.

Table 5: The Input Functions for the Formation of Verbal Group

Condition	Function	Alternative values
	cat	verbal-group
	lex	*a lexical verb* or *nominal*
	type-of-base	verb, nominal
if type-of-base is verb	lex-subj-obj-rel	none, reflexive, reciprocal
	lex-transition	transitive, intransitive
	subj-obj-rel	none, reciprocal, reflexive
	transition	none, intrans-trans trans1, trans2, trans3
	voice	active, passive
	polarity	positive, negative
	desc-verb	potential, haste, durative, approximative
	desc-polarity	positive, negative
	mood	finite, non-finite
if mood is finite	time	aorist, past, narr, progress, future, cond, optative, necessitative, imperative
	mode	none, past, narr, cond
	person	first, second, third
	number	singular, plural
	interrogative	none, yes-no
if mood is non-finite	type-of-verbal	noun, adj, adv
	non-finite	mek, me, iş (infinitive) Present, Past, Future (participle) ip, ere, ince, dikce ... (adverbial)

At least, the functions **cat**, **lex**, and **type-of-base** must be given with their values. The

other functions are optional. If a function does not appear in the input set but it is required, the first alternative is selected as a default value for that function. The following verbal groups are generated by the implemented system in FUF:[5]

1. Example: *sevebilmelisin*

```
(uni
 '((cat verbal-group)  (lex "sev")           (type-of-base verb)
   (polarity positive) (desc-verb potential) (desc-polarity positive)
   (mood finite)       (time necessitative)  (mode none)
   (person second)     (number singular)     (interrogative none)))
[[CAT=VERB][ROOT=sev][SENSE=POS][COMP=YABIL][TAM1=NECES][AGR=2SG]]
```

2. Example: *öğretmen olmayacaktı*

```
(uni
 '((cat verbal-group)  (lex "OGretmen")  (type-of-base noun)
   (polarity negative) (desc-verb none)  (desc-polarity positive)
   (mood finite)       (time future)     (mode past)
   (person third)      (number singular) (interrogative none)))
[[CAT=NOUN][ROOT=OGretmen][AGR=3SG][POSS=NONE][CASE=NOM]]
[[CAT=VERB][ROOT=ol][SENSE=NEG][TAM1=FUTURE][TAM2=PAST][AGR=3SG]]
```

3. Example: If Time and Mode are PAST and NARR respectively in the same verbal group, it will be ungrammatical.

```
>(uni '((cat verbal-group) (lex "kIr")       (type-of-base verb)
        (mood finite)      (time past)       (mode narr)
        (person third)     (number singular) (interrogative none)))
<fail>
```

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. One of the most important parts of the grammar is the verbal group that realizes the several semantic functions of a sentence. So, at the beginning, we have considered the most common grammatical structures of Turkish verbal groups and their implementation in FUF. The other parts of the grammar and the overall generation system are currently under development.

The ultimate generation system will take as input the semantic description of a sentence from an application program, and produce the worded text. The semantic description consists of

[5] 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.

three metafunctions: *ideational* such as *agent, actor, goal, process, location* for representing the constituents of the sentence and their roles, *interpersonal* such as *mood, modality* for establishing the relationship between the speaker and the listener, and *textual* such as *topic, focus, background* for presenting information as text in context. The systemic-functional grammar will provide us with useful mechanisms to organize and realize the linguistic resources.

References

[1] T. Banguoğlu. *Türkçenin Grameri.* Number 528 in Türk Dil Kurumu Yayınları. Türk Tarih Kurumu Basım Evi, Ankara, 1986.

[2] M. Elhadad. Types in functional unification grammars. In *Proceedings of ACL'90*, pages 642–655, 1990.

[3] M. Elhadad. *Using Argumentation to Control Lexical Choice: a Functional Unification Based Approach.* PhD thesis, Department of computer Science, Columbia University, 1990.

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

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

[6] R. Kasper. *Systemic Grammar and Functional Unification Grammar*, chapter 9, pages 176–199. In Systemic Functional Approaches to Discourse. Ablex, 1988.

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

[8] T. Kumano, T. Tokunga, K. Inui, and H. Tanaka. Genesys: An integrated environment for developing systemic functional grammars. In *Proceddings of the International Workshop on Sharable Natural Language Resources*, pages 78–85, Nara Institute of Science and Technology Ikoma, Nara, Japan, 10 - 11 August 1994.

[9] C. M. Matthiessen and J. A. Bateman. Nigel: A systemic grammar for text generation. Technical Report ISI/RR-83-105, Information Sciences Institute, University of Southern California, February 1983.

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

[11] D. D. McDonald. Natural language generation. *Encyclopedia of Artificial Intelligence*, pages 642–655, 1987.

[12] K. Oflazer. Two-level description of Turkish morphology. In *In Proceedings of the Sixth Conferance of the European Chapter of the Association for Computational Linguistic.* April 1993.

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