Generation of Simple Turkish Sentences with Systemic-Functional Grammar*

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

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

Abstract

This paper mainly presents a Turkish sentence generator for producing the actual text from its semantic description. To concentrate on the 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 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 simple 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*) (Halliday, 1985): 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* consists of a root word, and zero or more morphemes; a *morpheme* 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 *complex sentence* consists of more than one simple sentence that may be structurally or semantically connected to each other. Because the generation of simple sentences must be achieved before the generation of complex sentences, we concentrate on the generation of simple sentences in this paper.

We analyze simple Turkish sentences from the systemic-functional perspective to determine their structural and functional descriptions (Patten, 1988). 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) (Elhadad, 1990-2) to perform the linguistic resources. In our analysis, we determine the main process, participants and circumstantials of a simple sentence, and how they are realized in Turkish.

The remainder of this paper is organized as follows. In Section 2, we consider the grammatical analysis of the simple sentence. We present the functional analysis of simple sentences, which determines participants and their realizations in Turkish. We paid a special attention to the verbal group part of sentences which is used in the realization of the process. Section 3 gives a brief overview of the systemic-functional grammar approach to text generation, and particularly presents the system network of the simple sentence. Next, in Section 4, the implementation of a Turkish sentence generator is introduced, and then the generation of simple sentences is demonstrated. Finally, Section 5 presents conclusion and future work.

2 Grammatical Analysis of Simple Sentences

2.1 Predicate Types of Sentences

Turkish sentences can be divided into two groups depending on the type of their predicates: *verbal* and *nominal* sentences. If the predicate of the sentence is derived from a verb, it is called a verbal sentence.

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

Ilyas Cicekli and Turgay Korkmaz (1998) **Generation of Simple Turkish Sentences with Systemic-Functional Grammar.** In D.M.W. Powers (ed.) *NeMLaP3/CoNLL98: New Methods in Language Processing and Computational Natural Language Learning,* ACL, pp 165-173.

If the predicate is derived from a nominal[1] group, it is called a nominal sentence.

In verbal sentences, a verb is used as a base of a verbal group, other information such as time, mode, person is realized as suffixes to this base. The following is an example for a Turkish verbal sentence.

(1) Yarın okul-a gid-eceğ-iz.
Tomorrow school+DAT go+FUT+1PL
'We will go to school tomorrow.'

In positive nominal sentences, a nominal group becomes a finite verb with a substantive verb which is used as an auxiliary verb to demonstrate "to be" meaning of the predicate in four grammatical tenses: present, past, narrative and conditional.

(2) a. Ahmet başkan-dır.
Ahmet chairman+COP.
'Ahmet is the chairman.'

b. Kız çok güzel-di.
Girl very beautiful+PAST
'The girl was very beautiful.'

The negative sense of the nominal sentence is represented by a separate negative word `değil` (not to be) for the tenses mentioned above.

(3) Ahmet başkan değil-dir.
Ahmet chairman NegNoun+COP.
'Ahmet is not the chairman.'

The other tenses in the nominal sentences are expressed by the auxiliary verb `ol` (be). In this case, the auxiliary verb is realized the same way as the predicate of a verbal sentence.

(4) Yarın okul-da ol-ma-yacağ-ım.
Tomorrow school+DAT be+NEG+FUT+1SG
'I will not be at school tomorrow.'

2.2 Functional Analysis

From the functional perspective, all languages try to realize the common semantic functions with their own grammatical structures and lexical items. In this section, we consider the realization of each semantic function in Turkish.

A clause[2] (simple sentence) consists of three functional components: *process*, *participants*, and *circumstantials*. *Process* is the main constituent that represents an event or a state. *Participants* are persons or things involved in a process. *Circumstantials* are the optional constituents to describe the process from different perspective such as time, place, manner etc.

Participants and *circumstantials* are specified with new semantic functions to represent the special meanings, roles or relations in the clause. The specific participant functions depend on the type of process. The *transitivity* and *ergativity* analysises (Halliday, 1985) allow us to classify the processes in the language, and to describe the configuration of participants. The specific circumstantial functions do not strictly depend on the type of the process, and they are optionally used to give more information about the process.

2.2.1 Transivity Analysis

Transitivity analysis specifies the different types of processes recognized in the language, and determines the participants depending on these types. In this way, the logical relationships between the process and participants are provided. The types of processes and their special participants may be classified as follows.

1.*Material processes* express the notion that some entity "does" something which may be done "to" some other entity. Material processes contain *actor* as an obligatory participant that represents the one that does the deed, and *goal* as an optional participant that represents the one that the process is extended to. Material processes are realized by verbal sentences in Turkish.

2.*Mental processes* express feeling, thinking, and perceiving activities of humans. There are two participants in a mental process: *senser* who is the conscious being that senses, and *phenomenon* that is a thing or a fact which is sensed. Mental processes are also realized by verbal sentences in Turkish.

3.*Relational processes* express the way of "being". Relational processes can be classified according to the type of "being", and the explanation mode of "being". The type of "being" can be *intensive*, *circumstantial* and *possessive*. Each type can be explained in two modes: *attributive* and *identifying*. As a result, the six types of relational processes can occur.

The special participants for relational processes are determined depending on the mode of "being". In the attributive mode, there are two participants: *carrier* that is an entity to that an attribute is ascribed, and *attribute* is a determiner that is ascribed to *carrier*. The participant *identifier* is used to identify the participant *identified* in the identifying mode.

In Turkish, relational processes are realized by nominal sentences. In other words, the process of being is expressed by a substantive verb or a distinct auxiliary verb `ol` (be) depending on the time of the sentence. The *attribute* is conflated with the *process* of the sentence in the attributive mode, and the *identifier* is conflated with the *process* in the identifying mode. Thus, these two participants are used as nominal bases of verbal groups in the re-

[1] Nominal is a common name for nouns and adjectives

[2] Clause is used as a common name for sentences or sentence-like structures.

alization of the process. However, in the attributive mode of the possessive relation, a distinct word **sahip** (have) is used as the nominal base in the realization of the process, and the *attribute* is realized as a noun phrase in the dative case.

4.*Existential processes* express that something exists or happens. There is only one participant: *entity*. To express that *entity* exists or not, two distinct nouns **var** (exist) and **yok** (absent) are respectively used in Turkish. Thus, existential processes are also realized as nominal sentences, and these distinct nouns are used as nominal bases in the realization of the process.

2.2.2 Ergativity Analysis

If the process is "caused"-ergative, the analysis of the ergativity is required to find the functions *agent* (causer) and *medium* (affected) as participants of the process. Sometimes *medium* is conflated with *actor*, and sometimes with *goal*. In addition, the agent and the actor may be different participants to explain the fact that someone (agent) is causing someone else (actor) to perform the process. For instance, in (5), **Ali** (the agent) caused **Veli** (the actor) to paint the table.

(5) Ali masa-yı Veli-'ye boya-t-tı.
Ali table+ACC Veli+DAT paint+CAUS+PAST
'Ali had Veli paint the table.'

In Turkish, the causation hierarchy may be more complex. More than one agent–like participants may appear between the agent and the actor to explain that someone is causing another causing another and so on to perform the process. For example, an additional participant *agent-2* is illustrated in (6).

(6) Masa-yı Ali aracılığı ile Veli'ye boya-t-tır-dı-m.
Table+ACC Ali's help with Veli+DAT paint+CAUS+CAUS+PAST+1SG
'I told Ali to have Veli paint the table.'

We do not consider more complex causations, because they are not frequently used in practice.

2.2.3 Realization of Participants and Circumstantials

Participants are mapped onto syntactic functions such as *subject*, *direct-object*, *indirect-object*, etc. Participants are realized by noun groups and infinitive clauses in Turkish and their case markings depend on their syntactic roles in the sentence.

In contrast to participants, circumstantials are not mapped onto any syntactic functions, and they are directly realized by noun groups, post-positional groups or adverbs in the sentence structure. Circumstantial functions can be decomposed into seven classes and their possible realizations in Turkish are summarized in Table 1.[3]

Class	Sem. Func.	Realizations
Spatial	direction	AdvG, PP, NP+DAT/+LOC
	distance	NP+NOM
	origin	NP+ABL
	location	NP+LOC
	destination	NP+DAT
	path	PP, NP+DAT
Temporal	duration	PP, AdvG, NP+NOM
	frequency	AdvG, PP
	time	PP, AdvG, NP+LOC
Manner	instrument	PP, NP+INS
	quality	AdvG
	comparison	PP
Cause	reason	PP
	purpose	PP
	behalf	PP
Accom-paniment	comitative +	PP
	comitative −	NP+PRI
	additive +	PP
	additive −	PP
	Matter	PP
	Role	PP

Table 1: Realization of Circumstantials

2.2.4 Word-Order in the Sentence

The default word order of a Turkish sentence is 'Subject-Object-Verb'. Since Turkish is a free word order language, the syntactic functions in the sentence can be freely ordered to construct the sentence. Although the constituents can be freely ordered, each order provides the additional information to explain the different textual functions of each constituent. The textual functions can be identified as follows (Erguvanlı, 1979; Hoffman, 1995):

- the sentence-initial position as *topic*
- the immediately preverbal position as *focus*
- the postverbal position as *background information*

In the realization, each constituent may be conflated with one of these functions, and these functions are strictly ordered as shown in the following template[4]:

Topic ··· *Focus* *Process* *Background*

Naturally, the number of constituents in the sentence may be increased, and they can not be conflated with any textual function. For those kinds of constituents, we use a default word order in the implementation.

[3] Following notations are used: NP for noun phrase; PP for post-positional phrase; AdvG for adverb group; NOM for nominative; DAT for dative; ABL for ablative; LOC for locative; INS for instrumental; PRI for privative.

[4] The elements are represented in partial order. Three dots represent that different functions may be located.

In spite of the free word order characteristic of Turkish, there are some grammatical constraints on the word order. If *direct-object* is not focused in the sentence (7.a), it must be realized as a definite element. If *direct-object* is an indefinite element (7.b), it must be adjacent with the process. Otherwise, it will be ungrammatical (7.c).

(7) a. *Cam-ı* Ali kır-dı.
window+ACC Ali break+PAST
'Ali broke the window.'

b. Ali *cam* kır-dı
Ali window+NOM break+PAST
'Ali broke (a) window.'

c. * *Cam* Ali kır-dı.
window+NOM Ali break+PAST

2.3 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. The other lexical elements (such as `değil` (neg. noun), `mi` (question), `ol` (be)), the relevant suffixes and the 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 (Banguoğlu, 1986; Koç, 1990).

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 in Figure 1.b (given in Section 3) organize the internal structure of the verbal group. As a result, the following four general structures can occur:

1. **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 in this case is shown in Table 2 for the following examples in (8).[5] 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.

(8) a. Arkadaş-lar-ın-ı *sev-ebil-meli-sin.*
friend+3PL+2PP+ACC love+POT+NEC+2SG
'*You ought to be able to love* your friends.'

b.
Mektub-u *yaz-dır-acak* *mı-ydı-n*?
letter+ACC write+CAUS+FUT Ques+PAST+2SG
'*Were you going to have* the letter *written*?'

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). As a result, the structure of this case for the following examples is given in Table 3.

(9) a. Birisi tarafından *sev-il-mek* güzel-dir.
someone by love+PASS+INF* nice+COP
'It is nice *to be loved* by someone.'

b. Mektub-u *oku-yacak* adam gel-me-di.
letter+ACC read+PART* man come+NEG+PAST
'The man *who would read* the letter did not come.'

c. Ali okul-a *koş-arak* git-ti.
Ali school+DAT run+ADV* go+PAST
'Ali went to school *by running.*'

sev oku koş	-il				-mek -yacak -arak
Base	VF	POL	DV	DP	Non-Finite
verb	...	pos	none		infinitive
verb	...	pos	none		participle
verb	...	pos	none		adverbial

Table 3: Non-Finite Verbal Group from Verb

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" and the *existential processes*. 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

[5] 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. M-P-N stands for Mode, Person, and Number; VF stands for Voice Frame; POL stands for Polarity; DV stands for Descriptive Verb; DP stands for Descriptive Polarity.

sev			-ebil		-melisin
yaz	-dır				-acak
Base	VF	POL	DV	DP	Finite
verb	...	pos	potential	pos	...
verb	...	pos	none	pos	...

mı	-ydı-n
Interr-Tag	M-P-N
none	
yes-no	...

	-dır	
Subj-Obj-Rel	Transition	Voice
none	none	active
none	trans-1	active

Voice Frame

-meli		-sin	
-acak	-ydı	-n	
Time	Mode	Person	Number
necess	none	second	sing
future	past	third	sing

Finite

Table 2: Finite Verbal Group from Verb

öğretmen	-dir
öğretmen	
öğretmen	
Base	Finite
noun	substantive
noun	
noun	

değil	-dir
Neg-Noun(POL)	Finite
pos	
neg	substantive
pos	

mi	-dir
Interr-Tag	Finite
none	
none	
yes-no	substantive

Table 4: Finite Verbal Group from Nominal Group (1)

öğretmen	ol-ma-yacak-tı
Base	Aux::verbal-group, mood::finite
noun	...

Table 5: Finite Verbal Group Nominal Group (2)

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 4, a *substantive (predicative)* 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, and relevant suffixes are attached to this negative noun. In the second structure shown in Table 5, an auxiliary verb `ol` (be) appears as a separate element after the **base**. If the value of *time* is present, past, narrative or conditional then the first structure is selected, otherwise the second one is selected.

(10) a. O bir *öğretmen-dir.*
He a teacher+COP
'He *is a teacher.*'

b. O bir *öğretmen değil-dir.*
He a teacher NegNoun+COP
'He *is not a teacher.*'

c. O bir *öğretmen mi-dir?*
He a teacher Ques+COP
'*Is* he a *teacher?*'

(11) Ali *öğretmen ol-ma-yacak-tı.*
Ali teacher be+NEG+FUT+PAST
'Ali *was not going to be a teacher.*'

4. **base** is a *nominal group* and **mood** is *non-finite*: In this case, the same structure in Table 5 is used by changing the value of the **mood** of auxiliary verb to *non-finite.*

(12) a. öğretmen ol-mak
teacher be+INF*
'*to be a teacher*'

b. öğretmen ol-an
teacher be+PART*
'*(someone) who is a teacher*'

c. öğretmen ol-arak
teacher be+ADV*
'*as a teacher.*'

2.4 Noun Group (NP)

A noun group (noun phrase-NP) is a grammatical unit that contains at least one noun called *head*, and it may contain modifiers[6] 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.

[6] Modifier represents all kind of elements that determine, describe, modify, or classify 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 a subset of the head noun is specific or not, and 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; *qualifiers* which indicate the characteristics of the head noun in terms of some process in which the head noun is directly or indirectly involved as a participant. Qualifiers may be realized by a participle clause. 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. Although the details of the noun groups in Turkish are not given here, the noun groups are fully implemented in our system.

2.5 Post-Positional Group (PP)

Post-positional group (PP) has a simple structure that consists of an NP or infinitive, and a postposition particle in Turkish. Participles are closed class of words such as `göre` (according to), `doğru` (towards), `sonra` (after) etc. A particle cannot refer to any concept but it constructs a relationship between the NP and the other constituents. Each particle may enforce the NP in a particular case.

2.6 Adverb Group (AdvG)

Adverb group (AdvG) is used in the realization of several circumstantial functions given in Section 2.2.3. The main constituent of an adverb group is *head* which is an adverb that gives information about when, how, where, or in which circumstances something happens. In an adverb group, there may be additional modifiers to modify the head adverb.

3 System Network of Simple Sentence

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 (Patten, 1988). 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 (Matthiessen and Bateman, 1991; Patten, 1988). 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 sentence that consists of these grammatical units is generated. In Figure 1.a, if we select the *simple clause* feature from the *rank* system, we enter five more systems: *process*, *transivity*, *mood*, *voice* and *circumstantials*. After selecting proper features from these systems, 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 are considered in Section 4.

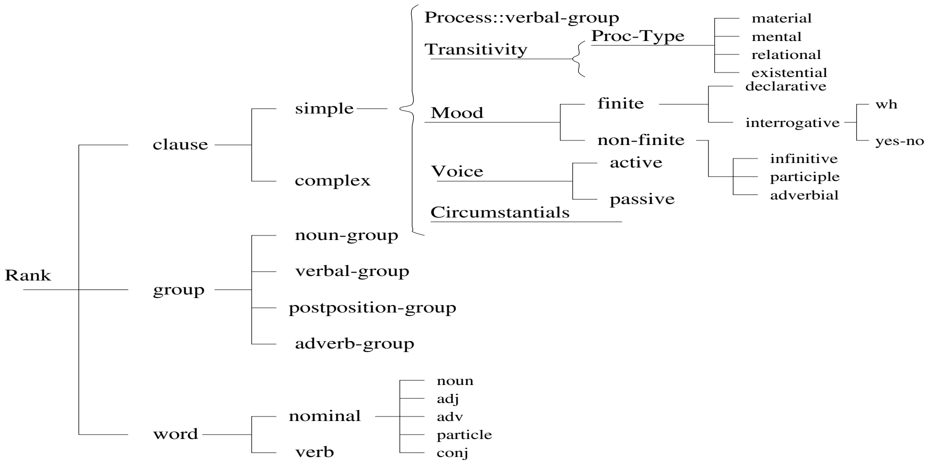
The required systems, the realization rules, and the appropriate context of each system in the linguistic description of the simple sentence are determined and organized by using the analysis described in the previous section. As a result, the system network given in Figure 1 is constructed. In the network, only systems and their appropriate contexts are displayed to express the basic linguistic description of simple sentences. 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 simple sentence, the system network is traversed by using the algorithm given above. For example, to produce the simple sentence `arkadaşlarını sevebilmelisin` given in (8.a), after the appropriate features are selected, we re-enter the system network to realize the *process* by a verbal group and to realize the *phenomenon* by a noun group. When the system network is re-entered to realize the *process* the following systems are entered and the appropriate features are selected in Figure 1.b: 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 2.

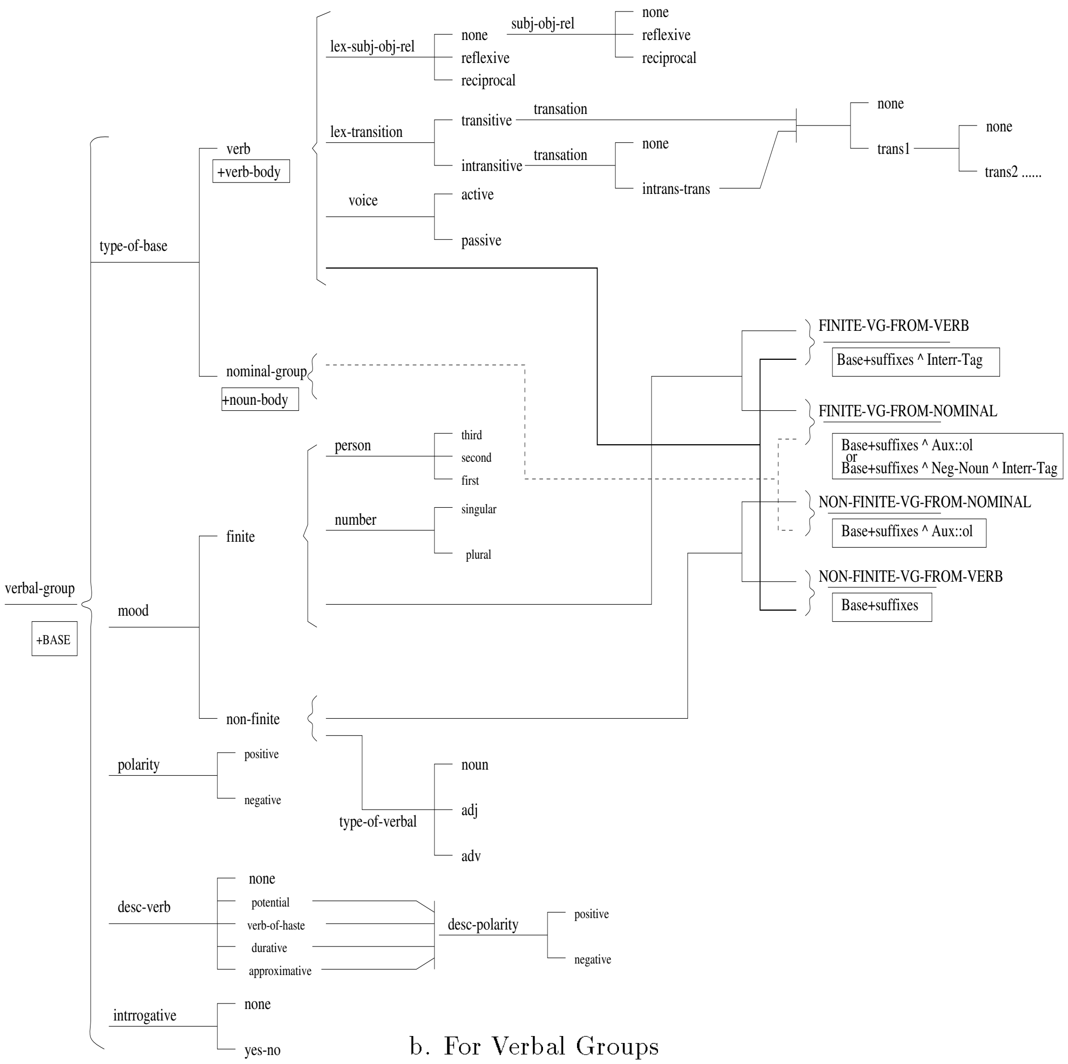
4 Implementation

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(Elhadad, 1993) 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 simple sentences.

The FUF text generation system consists of two main modules: a *unifier* and a *linearizer* (Elhadad, 1990-2). The unifier takes, as input, a *lexicalized*



a. For Rank System



b. For Verbal Groups

Figure 1: A System Network for the Sentence Generation in Turkish

semantic description of the text to be generated, and an *extended form of FUG*, and then produces

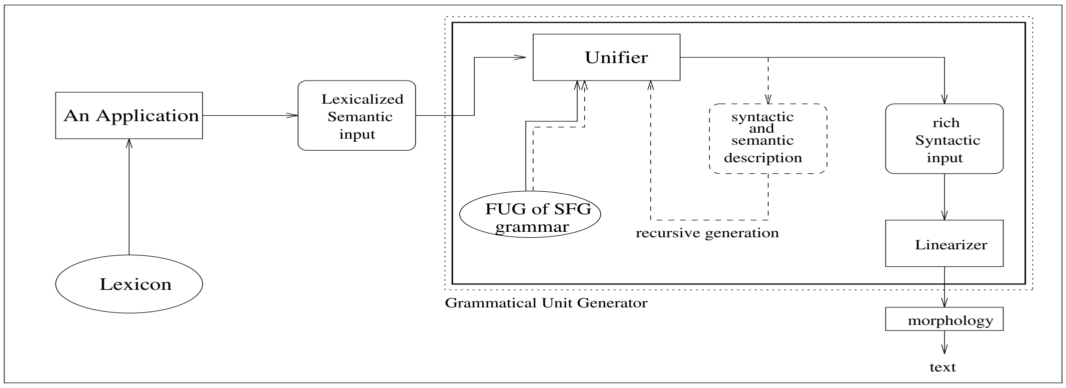


Figure 2: The Architecture of the Text Generator

as output a *rich syntactic description* of the text or some new inputs[7] (the semantic and syntactic descriptions) for the grammatical units that realize the specific components of the text (Elhadad, 1993). 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 (Oflazer, 1993). We assume that an application program that is not included in our implementation produces the *lexicalized semantic description* of the text. Consequently, the final text generation system can be organized as shown in Figure 2.

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 (Korkmaz, 1996; Kasper, 1988; Kumano et al., 1994). 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 (Kasper, 1988). In addition, FUF provides a typed feature formalism to implement the mutual exclusion, and hierarchical relations in SFG (Elhadad, 1990-1).

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.

The lexicalized semantic representation of a sentence must contain the required functions for that sentence. 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 simple sentence is generated by the system implemented in FUF:[8]

Example:

(13) dUn Ali Veli-'ye okul-da mektub-u dikkatlice yaz-dIra-ma-yabil-ir-di.
yesterday Ali Veli+DAT school+LOC letter+ACC carefully write+CAUS+NEGC+POT+AOR+PAST
'Ali might not have had Veli write the letter carefully at the school yesterday.'

In this example, the `time` function is the topic, and the `quality` function is the focus of the sentence. If the textual functions (`topic, focus, background`) were not given in this lexicalized semantic input, the default word order for participants and circumtantials would have been used, and the following sentence could have been generated.

Ali Veli-'ye okul-da dUn dikkatlice mektub-u yaz-dIra-ma-yabil-ir-di.

Lexicalized Semantic Input:

```
((cat simple-clause)
 (time aorist)
 (mode past)
 (mood declarative)
 (desc-verb potential)
 (desc-polarity negative)
 (voice  active)
 (process ((type material)
          (type-of-base verb)
          (agentive yes)
          (effective yes)
          (lex "yaz")))
 (participants (
   (actor ((cat proper) (lex "Veli")))
   (agent ((cat proper) (lex "Ali")))
   (medium ((cat common) (definite yes)
```

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

[8]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.

```
            (lex "mektup")))))
  (circum (
    (location ((cat common) (lex "okul")))
    (time ((cat adv) (lex "dUn")))
    (quality ((cat adv) (lex "dikkatlice")))))
  (topic {^ circum time})
  (focus {^ circum quality})
  (background none)))
```

Output:

```
[[CAT=ADVERB][ROOT=dUn]]
[[CAT=NOUN][ROOT=Ali][AGR=3SG]
 [POSS=NONE][CASE=NOM]]
[[CAT=NOUN][ROOT=Veli][AGR=3SG]
 [POSS=NONE][CASE=DAT]]
[[CAT=NOUN][ROOT=okul][AGR=3SG]
 [POSS=NONE][CASE=LOC]]
[[CAT=NOUN][ROOT=mektup][AGR=3SG]
 [POSS=NONE][CASE=ACC]]
[[CAT=ADVERB][ROOT=dikkatlice]]
[[CAT=VERB][ROOT=yaz][VOICE=CAUS][SENSE=POS]
 [SENSE=NEGC][TAM1=AORIST][TAM2=PAST][AGR=3SG]].
```

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 simple sentence that realizes the several semantic functions. 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.

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 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

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

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

M. Elhadad. *Using Argumentation to Control Lexical Choice: A Functional Unification Based Approach.* PhD thesis, Department of Computer Science, Columbia University, 1990.

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

E.E. Erguvanlı. *The Function of Word Order in Turkish Grammar.* PhD thesis, University of California, Los Angeles, 1979.

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

B. Hoffman. *The Computational Analysis of the Syntax and Interpretation of "Free" Word Order in Turkish.* PhD thesis, University of Pennsylvania, June 1995.

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

N. Koç. *Yeni Dilbilgisi.* İnkilap Kitapevi, İstanbul, 1990.

T. Korkmaz. *Turkish Text Generation with Systemic-Functional Grammar.* M.S. thesis, Bilkent University, June 1996.

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

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

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

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