PERSIS: A Natural-Language Analyzer for Persian

MOHAMMAD Ali SANAMRAD* and HARUYA MATSUMOTO**

A natural language analyzing system, PERSIS, is described. This system takes Persian texts as input and produces dependency networks which represent their meanings on a certain level of detail. Parsing is based on a model of grammar implemented by more than 850 syntactic production rules which employ 42 different structural descriptors. For the representation of concepts encoded in the language, PERSIS utilizes, depending on the structural descriptors, 17 prototypes with up to 10 syntactic and semantic attributes. These attributes are filled under control of the parser by iteratively applying feature-integration rules to simpler attribute lists with the primary attributes of words being obtained from the dictionary. At each parsing step, feasibility is verified through feature-checks made on the extracted attributes. Finally, when the parsing is finished, dependency networks for the input sentences and phrases are decoded from the final attributes by the recursive calling of 17 feature-interpreting routines associated with the attribute prototypes. Experimental results and examples throughout the paper illustrate capabilities of the system in handling syntax and semantics. PERSIS, the first analyzer developed for Persian, is written in LISP and machine translation is hoped to be among its many applications.

1. Introduction

For the past several decades, natural language processing has been one of the most active areas of research and at the same time one of the most difficult problems for the field of Artificial Intelligence [4], [9], [16]. Natural language processing has fascinating potential applications [1], [2], [7], [15] and has made significant contributions to other areas in the field of Computer Science.

Understanding natural language is a very complex decoding problem and the perfect understanding of a message in natural language may require full use of various levels and sources of knowledge such as syntactic, semantic, contextual, and even common sense. Nevertheless, the development of numerous theories and ideas in the areas of syntactic and semantic processing [5], [11], [12], [13], emerging from the fields of linguistics, philosophy, and psychology, have led to powerful language analysis techniques [8], [10], which in turn have resulted in the development of practical systems capable of understanding fragments of natural language [2], [17]. Such systems, though limited to working in restricted domains, have complete and privileged knowledge of their world and are capable of handling texts exhibiting a wide variety of linguistic phenomena.

Rapid progress in the field of natural language processing together with the practical applications brought about by this progress, are sufficient incentives for initiating such studies for languages not previously studied. In the present paper, we will describe a natural language analyzing system that takes Persian texts as input and produces dependency networks which represent their meanings on a certain level of detail. The system is named PERSIS (for Persian analysis) and will henceforth be referred to by this name. For the representation of concepts encoded in the language, PERSIS utilizes a set of syntactic and semantic attributes derived from syntactic structures and organized to ensure that the internal representation obtained is rich enough to represent the semantics of most Persian constructs.

Although it is far from a perfect system of understanding, and not as developed as other currently available analyzers [3], [6], PERSIS is the first analyzer developed for Persian. It is hoped that machine translation will be among its many possible applications since the dependency network extracted is very convenient and well suited to the mechanical translation of Persian into other languages.

The analysis algorithm is described in Section 2. Implementation of the analysis algorithm for Persian and the organization of the dictionary, grammar, and attributes are explained in Section 3. Examples in Section 4 illustrate the capabilities of PERSIS, and Section 5 consists of concluding remarks and further discussion.

PERSIS is written in LISP to run under the NEC ACOS-1000 time-sharing system; most of the linguistic knowledge (i.e., syntactic and semantic) is separated from the main routines and treated as data.

*Division of System Science, The Graduate School of Science and Technology, Kobe University, Nada, Kobe 657, Japan.
**Department of Instrumentation Engineering, The Faculty of Engineering, Kobe University, Nada, Kobe 657, Japan.

2. Analyzer Structure

Parsing is based on a model of grammar implemented in the form of production rules. The parser also has control over extraction of meaning and construction of attributes by feature integration rules. In this section, we give a formal description of the procedure of analysis.

2.1 Preliminaries

Assume that the "input string" is of the form

$$S=(s_1 s_2 \ldots) \tag{1}$$

with s_1, s_2, ... being the basic building units or "words".

The "dictionary" is of the form

$$D=(D_1 D_2 \ldots D_{ND}) \tag{2}$$

with

$$D_i=(s_i I_i) \quad \text{for} \quad i=1, 2, \ldots, ND. \tag{3}$$

For each word s_i, there is an information package I_i available:

$$I_i=(c_i F_i) \tag{4}$$

where c_i is a "structural descriptor" and determines grammatical category, and

$$F_i=(f_{i1} f_{i2} \ldots) \tag{5}$$

is a "list of attributes" associated with that word. The combination of c_i and F_i is then a "definition" of the word s_i. Of course, since there may be several definitions for a given word in a natural language, the actual entry for a particular word s_i in the dictionary may have the form

$$I_i=((c_i^1 F_i^1)(c_i^2 F_i^2) \ldots) \tag{6}$$

with superscripts being used to discriminate among the different definitions of that word. However, since only one definition of a word is normally considered at a time, for the sake of clarity, these superscripts will not be used in the rest of this discussion.

"Grammar" is represented as

$$G=(G_1 G_2 \ldots G_{NG}) \tag{7}$$

which is a set of "syntactic rules":

$$G_i=(G_i' G_i'') \tag{8}$$

$$G_i'=(g_{i1}' g_{i2}' \ldots g_{im}') \tag{9}$$

$$G_i''=(g_{i1}'' g_{i2}'' \ldots g_{ik}''), \tag{10}$$

with each syntactic rule G_i acting as a production rule which, if applied, will result in substitution of G_i' by G_i'' in the string under consideration. Note that g' and g'' are basically structural descriptors, and that m and k in (9) and (10) are particular to that i. It should also be noted that "syntactic patterns" (i.e., G_i') are organized such that the longest possible match is discovered first.

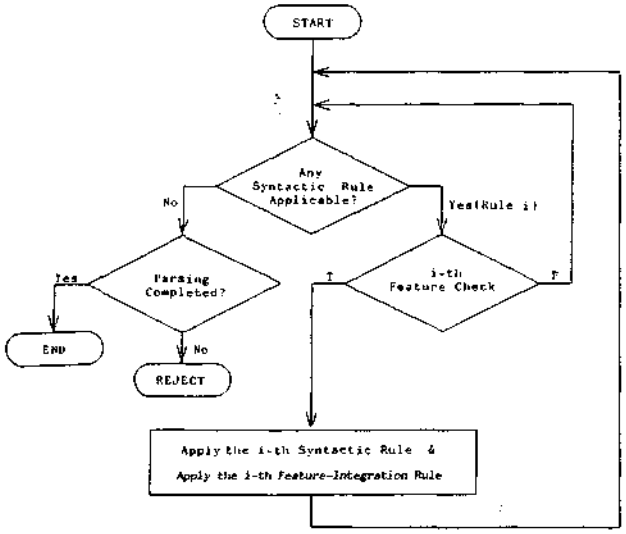

Fig. 1 Analysis flowchart.

For each syntactic rule G_i, there is a "feature check", FEATURE-CHECK$_i$, for verifying the feasibility of the parsing (e.g., checking the consistency of attributes of a noun group and its modifier). The parser tries to recognize basic structures and goes on by substituting these structures with their equivalents. It is then through feature checks that meaningless phrases are filtered out, or that a single structure is picked out from several different alternatives. It should be noted that this analyzer does not carry forward simultaneous parsings of the input string. It tries the first possible parsing and only if it is unsuccessful does it then try other paths and eventually other definitions of the words ("possible" in the sense that the feature checks are fulfilled).

For each syntactic rule G_i, there also exists a "feature integration rule" (or more precisely "routine"), INTEGRATE-FEATURES$_i$, for extracting new features and merging the previous ones. This means that each syntactic rule in the grammar is associated with a semantic rule. Syntactic rules allow longer and more complex syntactic structures to be built from simpler units. Conversely, associated semantic rules facilitate the compilation of complex attributes and meanings out of simpler ones. This is rather similar to Suppes' approach [14] to semantic processing. The analysis flowchart is shown in Fig. 1.

2.2 Analysis Algorithm

Now, let us follow the process of analysis to show how the above-mentioned rules are used in practice. For the input

$$S=(s_1 s_2 \ldots s_n), \tag{11}$$

a dictionary look up will produce a "string of structural descriptors"

$$C=(c_1 c_2 \ldots c_n), \tag{12}$$

and a "string of attribute lists"

$$F=(F_1F_2 \ldots F_n). \tag{13}$$

The parser then begins to match C with the syntactic patterns of the grammar using a depth-first searching algorithm and if the i-th syntactic rule is found to be applicable, that is, if a string

$$(c_jc_{j+1} \ldots c_{j+m-1})=(g'_{i1}g'_{i2} \ldots g'_{im})=G_i \tag{14}$$

is found in C, and i-th feature check is also fulfilled, that is

$$\text{FEATURE-CHECK}_i(F \text{ at point } j)=\text{True}, \tag{15}$$

then, the i-th syntactic rule is applied on the string of structural descriptors C and the i-th feature integration rule is applied on the string of attribute lists F, so that new strings for structural descriptors (C_{new}) and attribute lists (F_{new}) are built up respectively:

$$\begin{aligned} C_{\text{new}}&=(c_1c_2 \ldots c_{j-1}C^*c_{j+m} \ldots c_n) \\ &=(c_1c_2 \ldots c_{n'}) \end{aligned} \tag{16}$$

$$\begin{aligned} F_{\text{new}}&=(F_1F_2 \ldots F_{j-1}F^*F_{j+m} \ldots F_n) \\ &=(F_1F_2 \ldots F_{n'}) \end{aligned} \tag{17}$$

where

$$C^*=G''_i=(g''_{i1}g''_{i2} \ldots g''_{ik}), \tag{18}$$

$$F^*=\text{INTEGRATE-FEATURES}_i(F \text{ at point } j), \tag{19}$$

and

$$n'=n+k-m. \tag{20}$$

The process then continues by substituting C and F by C_{new} and F_{new} respectively, so that when the process is finished, C gives the parsing result (e.g., C may be merged to an N: noun group, or an SNT: sentence), while F contains the attributes associated with the whole input string. In other words, the final F is an internal representation of the concepts embedded in the input S. Examples in the following sections will clarify the analysis process.

3. Persian Analyzer

3.1 A Brief Introduction to Persian

Persian is a member of the Proto-Indo-European family of languages which is written using the standard Arabic alphabet plus four new letters added by the Iranians and is read from right to left. Throughout the paper, however, we have used a phonetic transcription based on that found in Persian Grammar (i.e., R1 below) with minor modifications for the sake of typing convenience.

The following are some facts about Persian syntax which should help the reader in understanding the examples to follow.

I) Word order in Persian is SOV (i.e., subject, object, verb).

II) There are no gender distinctions in Persian either in the form of affixes or in the pronominal system.

III) Verbs are inflected with respect to person, number, and tense. As a result of verbal agreement with the subject, the subject is often deleted when pronominal.

IV) In Persian, noun modifiers usually follow the noun.

V) Adverbal modification, on the other hand, always precedes the verb and is relatively unrestricted in the number and order of modifying expressions.

3.2 Persian Linguistic Data

To prepare the Persian linguistic data for the analyzer, two sources of information were used:

R1) Lambton, A. K. S., Persian Grammar, Cambridge University Press, 1953, Reprinted 1981.

R2) Khanlari, P., Junior Highschool Textbook (in Persian), Iran Ministry of Education, 1965.

After excluding phonological production rules and syntactic rules particular to Old Persian and Persian poetry, those rules which were deemed to be relevant to modern Persian were collected from both sources. Based on these rules and on the careful examination of thousands of sentences and phrases, more than 850 syntactic production rules were written, including nearly 150 simple and phrasal verb patterns. These were combined together with the necessary feature checks and feature integration rules. It should be noted that there are many cases in which different syntactic rules share the same feature integration rule or feature check. Moreover, there are many syntactic rules that are applied in all cases and, consequently, their corresponding feature checks are simply assigned the value True. Altogether, the number of necessary feature checks and feature integration rules is nearly 100 and 150 respectively.

It is usually very difficult (if not impossible) to quantitatively evaluate the linguistic knowledge of a natural language processor. Here, to give some general idea, Table I shows to what extent the above-mentioned collections of rules from the two sources R1 and R2 are included in PERSIS, either as syntactic rules or in the

Table 1 Use of rules from the information sources in forming syntactic rules, dictionary, and feature checks. Percents are for all of the rules that were considered to be relevant to modern Persian.

Rules	Included in Syntactic Rules	Included in Dictionary	Included in Feature Checks	Could not be Included
Rules of R1	41.8%	38.7%	6.4%	13.1%
Rules of R2	51.5%	31.7%	4.0%	12.8%

Table 2 Structural descriptors.

ADJ:	Adjectives	PPT:	Past Perfect
ADV:	Adverbs	PR:	Present
CADJ:	Comparative Degree of Adjectives	PRD:	Predicates
		PREP:	Prepositions
CPR:	Continuous Present	PROD:	Time Periods
CPT:	Continuous Past	PRPR:	Proportions
DAY:	Dates & Days	PRS:	Present Stem of Verbs
DOBJ:	Direct Objects	PSFX:	Pronominal Suffixes
FRC:	Fractions	PT:	Preterite
FTR:	Future	PTS:	Past Stem of Verbs
IMP:	Imperative Plural	SADJ:	Superlative Degree of Adjectives
IMS:	Imperative Singular		
INF:	Infinitives of Verbs	SPR	Subjunctive Present
INTJ:	Interjections	SPT:	Subjunctive Past
MNTH:	Months	SNT:	Sentences
N:	Nouns	TIME:	Time
NUM:	Cardinal Numbers	TXT:	Text
ORD:	Ordinal Numbers	UNIT:	Classifiers & Counters
PEND:	Personal Endings	VOC:	Vocatives
PP:	Past Participle	VRB:	Verbs
PPR:	Present Perfect	WEEK:	Days of the Week
PPRN:	Personal Pronouns	YEAR:	Years

form of feature checks and dictionary entries.

3.3 Dictionary

There are about 100 words, particles, case-markers, suffixes, prefixes, etc. which are used in syntactic patterns and act as "functional operators" in determining grammatical structures, and are included without further categorization in the dictionary. In addition to these entries, the dictionary currently contains more than 600 words, which are sufficient for carrying out the experiments described in this paper and may obviously be increased as necessary. Besides simple words, the dictionary contains contractions and different spellings of words, compound verbs, and common expressions and idiomatic proverbs that can not be understood as a composite of the meanings of their component parts. Also, since polite forms in Persian are produced by the substitution of words rather than a change in structure, these have also been included in the dictionary. Since dictionary entries include syntactic descriptors as well as associated attributes, the presentation of some typical examples of dictionary entries is postponed until section 3.5 after an explanation of the syntactic descriptors and attributes themselves.

3.4 Syntax

There are 42 structural descriptors employed in forming syntactic rules. These descriptors are listed in Table 2.

The following are some typical syntactic production rules:

N ha: ⟶ *N* (Plural)
N e/ye *N* ⟶ *N* (Genitive Case)
N e/ye ADJ ⟶ *N*
N i ke SNT ⟶ *N* (Relativized Sentence)
ADV ADV ⟶ ADV
VRB ⟶ PRD
ADV PRD ⟶ PRD
N PRD ⟶ SNT
PRD ⟶ SNT

The following examples illustrate how parsing is performed for a variety of Persian constructions. Note that functional operators are *not* assigned any descriptor. For the sake of clarity, those structures which are recognized at each step of the parsing process are underlined. However, especially in the later examples, the parsing process has been condensed to avoid repetition. The following abbreviations are used in the examples

CNJ: Conjunction
DOM: Direct-object marker
GEN: particle for determining genitive case
INDF: Indefinite marker
NEG: Negative
PFX: Prefix
PL: Plural
REL: Relative marker
SNG: Singular

The first example shows how relative clauses are treated.

keta:b i ke diruz kharid id
book INDF REL yesterday buy(PTS) 2nd-PL
(The book you bought yesterday)
N i ke ADV PTS PEND
N i ke ADV PT
N i ke ADV PRD
N i ke PRD
N i ke SNT
N

The next two examples illustrate the handling of adjectives:

gol ha: ye ghermez e ghashang
flower PL GEN red GEN beautiful
(Red beautiful flowers)
N ha: ye ADJ *e* ADJ
N ye ADJ *e* ADJ
N e ADJ
N

tond tar az ba:d
fast more than wind
(Faster than wind)
ADJ tar az *N*
CADJ az *N*
ADJ

The following example shows parsing of a simple sentence with an adverbial phrase:

hasan ra: dar keta:bkha:ne did am
Hassan DOM in library see(PTS) 1st-SNG
(I saw Hassan in the library)
N ra: PREP *N* PTS PEND
DOBJ ADV PT
DOBJ ADV PRD
DOBJ PRD

PRD
SNT

Time expressions are parsed as follows (NUM1 is the category of numbers from ten to nineteen).

sa:ate dah o pa:nzdah daghighe ye
clock ten CNJ fifteen minute GEN
sobh e shambe
morning GEN Saturday
(Saturday morning at 10:15)

sa:ate NUM1∘NUM1 daghighe ye TIME *e* WEEK
sa:ate NUM∘NUM daghighe ye TIME *e* WEEK
TIME ye TIME *e* DAY
TIME *e* DAY
TIME
ADV

The final two examples show the parsing of a conditional and an interrogative sentence:

agar ba: ghata:r na rav im dir
if by train NEG go(PRS) 1st-PL late
kha:h im resid
will 1st-PL arrive(PTS)
(If we don't go by train, we will be late)

agar PREP *N* na PRS PEND ADV kha:h PEND PTS
agar ADV PR ADV FTR
agar ADV PRD ADV PRD
agar PRD PRD
agar SNT SNT
SNT

key be esfa:ha:n rafte and
when to Isfahan go(PP) 3rd-PL
(When have they gone to Isfahan?)

ADV PREP *N* PP PEND
ADV ADV PPR
ADV PRD
PRD
SNT

3.5 Attributes, Semantics, and Dependency Networks

Meaning representation in PERSIS is based on the idea that meanings and concepts encoded in a language can be approached through the determination of a definite number of attributes. These attributes provide information (syntactic and semantic) about the words and their roles and relations in the phrase up to a certain depth. An important problem then is the determination of what attributes should be included in order to insure that the representation is minimally sufficient, complete, and capable of handling the meanings embedded in any possible sentence or phrase of the language. PERSIS makes use, depending on grammatical category, of up to ten attributes to represent the meaning encoded in different structures. There are 17 basic "prototypes" for attributes given in Appendix A. The prototype for attributes of *N*, for example, consists of the following items:

(a) Normal/Interrogative
(b) Person
(c) Quantity
(d) Abstract/Physical/Proper
(e) Counter Class
(f) Human/Animal/Place/Time/Condition, State/Inanimate
(g) Attributes of Apposition *N*
(h) Attributes of *N* in Genitive Case
(i) Attributes of ADJ
(j) Attributes of Relativized SNT+A Pointer to the Dictionary Entry

(a) determines whether the noun is interrogative or not. This attribute is normally used for pronouns which replace nouns. For example, *chekasa:ni* (who-plural) is interrogative (and plural and human as well). (b) and (c) determine person (1st, 2nd, 3rd) and number (singular, plural, No.) of the noun, while (d) and (f) categorize it and give some information about its nature. In Persian, different counters are used for different classes of nouns (e.g., *nafar* for human beings, *ba:b* for houses, *jeld* for books, . . .). This is specified in (e). Finally, (g), (h), (i), and (j) each contains a list of attributes of other nouns, adjectives, or sentences which are used for further description of the noun. The last item of the prototype is a pointer to the dictionary entry and is used for backtracking and generation of dependency networks.

Note that everything is coded alphanumerically and that the actual system does not utilize the long lists of symbols in the attributes.

The following are some examples of dictionary entries. Note that *X* means (still) not determined.

madrese
school
(madrese *N* (Normal *X* 1 Physical *X* Place *X X X X*))
sefid
white
(*sefid* ADJ (Normal Color *X*))
besor.at
quickly
(besor.at ADV (Normal Manner *X*))

The following are some examples of the attributes derived for simple phrases and sentences.

ruye miz
on table
(on the table)—Treated as ADJ
(Normal Position
(Normal 3rd 1 Physical *X* Inanimate *X X X X*+A Pointer to "*miz*")+A Pointer to "*ruye*")

sefid tar as barf
white more than snow
(whiter than snow)—ADJ
(Normal Color
(Normal 3rd *X* Physical *X* Inanimate *X X X X*+A Pointer to "*barf*")+A Pointer to "*sefid*")

ba: otomobil
by car
(by car)—ADV
(Normal Instrument-Conveyance
(Normal 3rd *X* Physical *X* Inanimate *X X X X*+A Pointer to "*otomobil*")+*X*)

bara:ye che
for what
(why?)—ADV
(Interrogative Motive-Purpose $X+X$)
key hasan ra: did id
when Hassan DOM see(PTS) 2nd-PL
(When did you see Hassan?)—SNT
(Interrogative Positive
(Active Transitive Doing-Action+A Pointer to "*didan*": to see)
(Normal 2nd Plural Physical $X\ X\ X\ X\ X\ X+X$)
X
(Normal 3rd 1 Proper X Human $X\ X\ X\ X+$A Pointer to "*hasan*": Hassan)
(Interrogative Time $X+X$)
Simple X Past)

Note that even if the structure of a sentence is not interrogative, the sentence will be considered so as soon as one of its attributes (i.e., subject, object, adverb, etc.) is recognized to be interrogative.

As should have already been noticed, the attributes associated with a particular descriptor may include lists of attributes of the same or different descriptors and this embedding can go to an arbitrary depth. The system is able to iteratively build more complex and higher-order lists of attributes from its current lists of attributes and by applying feature integration rules under control of the parser. Note that there is, of course, no need for all attributes in the lists to be filled during the process of analysis and, depending on the input, many of them may be left empty. The filled features are then used in the feature checks. For example, when a rule is going to assign a noun as the subject for a verb that it follows, first it is checked that the verb does not possess a passive infinitive (i.e., the attribute 'a' of the infinitive in the attribute 'c' of the verb is checked). Then, since the Persian verbs are inflected with respect to person and number, these attributes of the subject are checked not to contradict those of the verb (i.e., the attributes 'b' and 'c' of the noun should match the same attributes in the attribute 'd' of the verb).

The attributes contain enough information for generating a "dependency network" of the input (i.e., determining action, actor, object, instrument, location, time, etc. in a manner somewhat similar to other representations of meaning [3], [11]). However, these dependency networks are embedded structures designed so that they efficiently map the internal representation of the meaning in PERSIS. The main actions or topics are covered in the topmost structure. This structure contains other embedded structures which give further details, and this embedding may go to an arbitrary depth depending on the complexity of the input sentences and phrases. The dependency networks, several examples of which are given in the next section, can be useful for further processing and various applications. The module that decodes final attributes into dependency networks has a simple structure. It contains 17 "feature interpreting routines" for each of the 17 different attribute prototypes explained above. The process of decoding begins by calling the routine associated with the final structural descriptor derived by the parser, and continues by recursively calling necessary feature interpreting routines. The final output is then generated by a "revising module" which improves the readability. Examples given in the next section show how the meaning of the input sentences and phrases is extracted and put in the form of dependency networks suitable for possible further processing.

4. Experiments

As mentioned earlier, it is very difficult to evaluate how powerful a natural language analyzer is, and it is only through some experiments and examples that capabilities of the system can be illustrated. This section is devoted to such a task and contains several examples as well as some experimental results.

Before showing the result of the experiments, it is necessary to explain the case of ambiguous structures. Since the system uses a depth-first searching algorithm, ambiguity is not really encountered and other possible parsing paths for a sentence or phrase are not tried if an acceptable parsing is found. However, a parsing-tree is always carried such that back-tracking can occur when necessary. The parser also makes use of a "list of dead-end structures" to avoid fruitless searching. This list is gradually formed from those structures that were reached in the process of parsing and for which the parsing failed and a back-track occurred.

In an experiment, PERSIS was able to successfully parse about 88.5% of an input text composed of 200 sentences and phrases with more than 1600 words, randomly picked from R1. Those cases where the system failed are roughly classified in Table 3.

Several analysis examples are given below. Note that English glosses are obtained by the help of the dictionary entry pointers in the related attribute lists.

haza:r o nohsad o hashta:d
thousand CNJ nine-hundred CNJ eighty
o cha:ha:r
CNJ four

Table 3 Classification of those cases when PERSIS failed to correctly analyze the input sentences and phrases.

No. of Sentences and Phrases	Explanation
9	Expressions were neither noun groups nor sentences (e.g., address expressions, book titles, etc.)
9	Syntactic patterns were not enough
2	Wrong features extracted
3	Words which could be understood from the context, were omitted.
Total 23	

(One thousand nine hundred eighty four)
NUMBER: 1984
ali mive ha: ra: khord
Ali fruit PL DOM eat(PTS)
(Ali ate the fruits)
ACTION: eating
ACTOR: Ali
OBJECT: fruits
TIME: past
mive ha: bevasileye ali khorde shod
fruit PL by Ali eat(PP) become(PTS)
and
3rd-PL
(Fruits were eaten by Ali)
ACTION: eating
ACTOR: Ali
OBJECT: fruits
TIME: past
ma: ba: otomobil besor.at be kha:ne
we by car quickly to home
raft im
go(PTS) 1st-PL
(We went home by car quickly)
ACTION: going
ACTOR: we
DESTINATION: home
INSTRUMENT/CONVEYANCE: car
TIME: past
MANNER: quickly
vaghti az ghom mi raft am
when from Qom PFX go(PTS) 1st-SNG
ali ra: did am
Ali DOM see(PTS) 1st-SNG
(I saw Ali when I was going from Qom)
ACTION: seeing
ACTOR: I
OBJECT: Ali
TIME: past & ACTION: going
ACTOR: I
SOURCE: Qom
TIME: past—progressive
yek sa:l o nim
one year CNJ half
(One year and a half)
PERIOD: 1 year and 6 months
agar ghaza: bad-mazze ast na khor
if food not-tasty is NEG eat(PRS)
id
2nd-PL
(If the food is not tasty, do not eat)
PROHIBITION FROM ACTION: eating
ACTOR: you—plural
CONDITION: SUBJECT: food
HAS TASTE: bad
chera: heram e sabz ra: barda:sht
why pyramid GEN green DOM pick(PTS)
i
2nd-SNG
(Why did you pick up the green pyramid?)
ASKING REASON OF ACTION: picking up
ACTOR: you-singular
OBJECT: pyramid WITH COLOR: green
TIME: past

5. Discussions

A powerful natural-language analyzer, PERSIS, is presented which is capable of deriving meanings embedded in complicated Persian Constructions. Dependency networks for the input sentences and phrases are produced from a set of syntactic and semantic attributes which are filled under control of the parser by iteratively applying feature-integration rules to simpler attribute lists. While parsing, the analyzer does not perform any deep semantic processing; rather it tries to find a feasible parsing of the input by verifying the truth of feature checks at each parsing step. In PERSIS, the dictionary entries contain a great deal of information including the syntactic categories the words belong to as well as a list of primary associated attributes.

Experimental results and examples shown in the paper illustrate the capabilities of PERSIS in handling syntax and semantics. However, as is true of similar systems, positive examples do not guarantee (and we are not making a claim for) the complete coverage of all possible constructions to be found in Persian. Among the remaining problems are:

A) how the present system can be embedded in a contextual analyzer,
B) how to handle grammatically questionable input,
C) how to deal with inconsistent attributes which may conflict with the attributes presently extracted, such as is often the case with metaphor,

and finally,

D) how to implement the system for developing operational systems in application areas such as question-answering, data-base query, and machine translation.

We are now investigating strategies that will accomplish these tasks and hope that this study will be a first step towards powerful natural language processing systems for Persian in various application areas.

Last, it should also be mentioned that the system may be adopted to a new natural language by writing the necessary syntactic rules, designing attribute prototypes, and forming feature checks and feature integration routines based on these rules and attributes.

Acknowledgments

The authors would like to thank Mr. Stanley Dubinsky of the Department of Modern Languages and Linguistics at Cornell University for reviewing this paper and contributing valuable suggestions on the con-

tent and presentation.

References

1. CARBONELL, J. G. et al. Steps Toward Knowledge-Based Machine Translation, *IEEE Trans. Pat. Anal. Mach. Intel.*, PAMI-3, No. 4 (July 1981), 376–392.
2. CARBONELL, J. R. AI in CAI: An Artificial Intelligence Approach to Computer-Aided Instruction, *IEEE Trans. Man-Mach. Syst.*, MMS-11 (1970), 190–202.
3. GERSHMAN, A. V. A Framework for Conceptual Analyzers, in Strategies for Natural Language Processing, Lehnert, W. G. and Ringle, M. H. Eds., Erlbaum (1982), 177–197.
4. MCCORDUCK, P. Machines who think, W. H. Freeman and Company (1979), 239–271.
5. MINSKY, M. A Framework for Representing Knowledge, in The Psychology of Computer Vision, Winston, P. H. Ed., *McGraw-Hill* (1975), 211–277.
6. NAGAO, M. et al. Analysis of Japanese Sentences by Using Semantic and Contextual Information—Semantic Analysis (in Japanese), *Trans. IPSJ* 17, 1 (Jan. 1976), 10–18.
7. NAGAO, M. et al. An English-Japanese Machine Translation System of the Titles of Scientific and Technical Papers (in Japanese), *Trans. IPSJ* 23, 2 (March 1982), 202–210.
8. NISHIDA, F. Fundamentals of Language and Logic Processing (in Japanese), *Corona, Japan* (1981), 87–116.
9. ROSENSCHEIN, S. Natural-Language Processing: Crucible for Computational Theories of Cognition, *Proc.* IJCAI-83 (Aug. 1983), 1180–1186.
10. SAGER, N. Natural Language Information Processing, Addison-Wesley (1981).
11. SCHANK, R. C. Conceptual Information Processing, *North-Holland (1975)*.
12. SCHANK, R. C. and ABELSON, R. P. Scripts, Plans, Goals and Understanding, Erlbaum (1977).
13. SIMMONS, R. F. Semantic Networks, in Computer Models of Thought and Language, Schank, R. C. and Colby, K. M. Eds., W. H. Freeman and Company (1973), 63–113.
14. SUPPES, P. Variable-Free Semantics with Remarks on Procedural Extensions, in Language, Mind, and Brain, Simon, T. W. and Scholes, R. J. Eds., Erlbaum (1982), 21–34.
15. UCHIDA, H. and SUGIYAMA, K. A Machine Translation System from Japanese into English Based on Conceptual Structure, *Proc.* COLING-80 (Oct. 1980), 455-462.
16. WALTZ, D. L. The State of the Art in Natural Language Understanding, in Strategies for Natural Language Processing, Lehnert, W. G. and Ringle, M. H. Eds., Erlbaum (1982), 3–32.
17. WINOGRAD, T. Understanding Natural Language, Academic Press (1972).

Appendix A: Attribute Prototypes

I) *N*, DOBJ, VOC
- *F*: (a) Normal/Interrogative
  - (b) Person
  - (c) Quantity
  - (d) Abstract/Physical/Proper
  - (e) Counter Class
  - (f) Human/Animal/Place/Time/Condition, State/Inanimate
  - (g) Attributes of Apposition *N*
  - (h) Attributes of *N* in Genitive Case
  - (i) Attributes of ADJ
  - (j) Attributes of Relativized SNT
  - +A Pointer to the Dictionary Entry

II) ADJ, CADJ, SADJ
- *F*: (a) Normal/Interrogative
  - (b) Taste/Color/Shape/Size/Condition/Material/Relation/Order/Quality/Quantity/Place, Position
  - (c) Attributes of Compared *N*
  - +A Pointer to the Dictionary Entry

In the case where "Place, Position" occurs as the second feature, the third feature will be "Attributes of *N*" with which a position relation exists, and the dictionary entry pointer will indicate this "position".

III) ADV
- *F*: (a) Normal/Interrogative
  - (b) Manner/Time/Period of Time/Condition/Quantity/Instrument, Conveyance/Location/Motive, Purpose/Source/Destination/Material/Association
  - (c) Attributes of *N* Used in Forming the ADV
  - +A Pointer to the Dictionary Entry

When "Time" or "Period of Time" occurs as the second feature, the third feature will be "Attributes of TIME" or "Attributes of PROD" respectively.

IV) INF, PRS, PTS, PP
- *F*: (a) Active/Passive
  - (b) Transitive/Intransitive
  - (c) Having Quality/Taking Quality/Doing Action/Receiving Action
  - +A Pointer to the Dictionary Entry

V) VRB, PRD, CPR, CPT, IMP, IMS, PPR, PPT, PR, PT, SPR, SPT, SNT
- *F*: (a) Indicative/Interrogative/Imperative
  - (b) Positive/Negative
  - (c) Attributes of INF
  - (d) Attributes of Subject *N*
  - (e) Attributes of ADJ—needed when "Having Quality" or "Taking Quality" occurs in INF
  - (f) Attributes of Object *N*
  - (g) Attributes of ADV
  - (h) Simple/Compound: Reason/Condition/Time/Else
  - (i) Attributes of SNT—when "Compound"
  - (j) Tense

VI) PPRN, PEND, PSFX
- *F*: (a) Person
  - (b) Singular/Plural

VII) UNIT
- *F*: (a) Counter Class

VIII) INTJ
- *F*: (a) Vocative/Scolding/Admiration/Condemnation/Regret/Joy/Surprise/Pain/Caution

IX) NUM, ORD, FRC, PRPR
- *F*: (a) Value

X) TIME
- *F*: (a) Attributes of DAY
  - (b) Hour
  - (c) Minute
  - (d) Morning/Noon/Afternoon/Evening/Night/Midnight

(e) Attributes of SNT—Source of Time

XI) DAY
F: (a) Attributes of YEAR
(b) Attributes of MNTH
(c) Attribute of WEEK
(d) Day

XII) WEEK
F: (a) A Number from 1 to 7

XIII) MNTH
F: (a) Iranian/Foreign/Islamic
(b) A Number from 1 to 12

XIV) YEAR
F: (a) Iranian/Foreign/Islamic
(b) A Number

XV) PROD
F: (a) Number of Years
(b) Number of Months
(c) Number of Weeks
(d) Number of Days
(e) Number of Hours
(f) Number of Minutes

XVI) PREP
F: (a) Source/Destination, Direction/Location /Condition, State/Instrument, Conveyance/Association/Motive, Purpose/ Time Period/Time Source/Time Destination/Material/Cause

XVII) TXT
F: (a) Attributes of SNT
(b) Attributes of SNT
...
...

(Received September 5, 1984; revised November 22, 1985)