Interaction Grammar for the Persian Language: Noun and Adjectival Phrases

Masood Ghayoomi
Nancy2 University
54506 Vandoeuvre, Nancy cedex, France
masood29@gmail.com

Bruno Guillaume
LORIA - INRIA, BP 239
54506 Vandoeuvre, Nancy cedex, France
Bruno.Guillaume@loria.fr

Abstract

In this paper we propose a modelization of the construction of Persian noun and adjectival phrases in a phrase structure grammar. This modelization uses the Interaction Grammar (IG) formalism by taking advantage of the polarities on features and tree descriptions for the various constructions that we studied. The proposed grammar was implemented with a Metagrammar compiler named XMG. A small test suite was built and tested with a parser based on IG, called LEOPAR. The experimental results show that we could parse the phrases successfully, even the most complex ones which have various constructions in them.

1 Introduction

Interaction Grammar (IG) is a grammatical formalism which is based on the notions of polarized features and tree descriptions.

Polarities express the resource sensitivity of natural language by modeling the distinction between saturated and unsaturated syntactic construction (Guillaume and Perrier, 2008).

IG focuses on the syntactic level of a natural language. This formalism is designed in such a way that it can be linked with a lexicon, independent of any formalism. The notion of polarity that is at the heart of IG will be discussed in section 2.2. In IG, the parsing output of a sentence is an ordered tree where nodes represent syntactic constituents described by feature structures.

What we are interested in is studying the construction of constituencies of the Persian language according to IG. Among various constituencies in the language, we have focused on the construction of Persian noun phrases and adjectival phrases as the first step to build a grammar for this language.

The current work covers only noun and adjectival phrases; it is only a first step toward a full coverage of Persian grammar. The grammar presented here could have been expressed in Tree Adjoining Grammar (TAG) or even in Context Free Grammar with features, but we strongly believe that the modelization of the verbal construction of Persian, which is much more complex, can benefit from advanced specificities of IG, like polarities, underspecifications and trees.

2 Previous Studies

2.1 IG for French and English

The first natural language considered within IG was French. A large coverage grammar which covers most of the frequent constructions of French, including coordination, has been built (Perrier, 2007; Le Roux and Perrier, 2007).

Recently, using the fact that the French and English languages have many syntactic similarities, Planul (2008) proposed an English IG built by modifying the French one. These two grammars were tested on the Test Suite for Natural Language Processing (TSNLP; Oepen et al, 1996). Both cover 85% of the sentences in the TSNLP.

2.2 Polarity

The notion of polarity is based on the old idea of Tesnière (1934), Jespersen (1935), and Adjukiewicz (1935) that a sentence is considered as a molecule with its words as the atoms; every word is equipped with a valence which expresses its capacity of interaction with other words, so that syntactic composition appears as a chemical reaction (Gaiffe and Perrier, 2004). Apparently, it seems Nasr (1995) was the first to propose a

Proceedings of the 7th Workshop on Asian Language Resources, ACL-IJCNLP 2009, pages 107–114,
Suntec, Singapore, 6-7 August 2009. ©2009 ACL and AFNLP

formalism that explicitly uses the polarized structure in computational linguistics. Then researches such as Muskens and Krahmer (1998), Duchier and Thater (1999), and Perrier (2000) proposed grammatical formalisms in which polarity is also explicitly used. However, Categorial Grammar was the first grammatical formalism that exploited implicitly the idea of polarity (Lambek, 1958). Recently, Kahane (2006) showed that well-known formalisms such as CFG, TAG, HPSG, and LFG could be viewed as polarized formalisms.

IG has highlighted the fundamental mechanism of neutralization between polarities underlying CG in such a way that polarities are attached to the features used for describing constituents and not to the constituents themselves. Polarization of a grammatical formalism consists of adding polarities to its syntactic structure to obtain a polarized formalism in which neutralization of polarities is used to control syntactic composition. In this way, the resource sensitivity of syntactic composition is made explicit (Kahane, 2004).

In trees expressing syntactic structures, nodes that represent constituents are labeled with polarities with the following meanings: A constituent labeled with a negative polarity (<-) represents an expected constituent, whereas a constituent labeled with the positive polarity (->) represents an available resource. Both of these polarities can unify to build a constituent which is labeled with a saturated neutral polarity (<=>) that cannot interact with any other constituents. The composition of structures is guided by the principle of neutralization that every positive label must unify with a negative label, and vice versa. Nodes that are labeled with the simple neutral polarity (=) do not behave as consumable resources and can be superposed with any other nodes any number of times; they represent constituents or features indifferently.

The notion of saturation in terms of polarity is defined as a saturated structure that has all its polarities neutral, whereas an unsaturated structure keeps positive or negative polarities which express its ability to interact with other structures. A complete syntactic tree must be saturated; that means it is without positive or negative nodes and it can not be composed with other structures: so all labels are associated with the polarity of = or <=>.

The set of polarities {-> , <- , = , <=>} is equipped with the operation of compositional unification as defined in the table below (Bonfante et al, 2004):

	<-	->	=	<=>
<-		<=>	<-	
->	<=>		->	
=	<-	->	=	<=>
<=>			<=>	

Table 1. Polarity compositions on the nodes

2.3 Tree Description Logic in IG

Another specification of IG is that syntactic structures can be underspecified: these structures are trees descriptions. It is possible, for instance, to impose that a node dominates another node without giving the length of the domination path. Guillaume and Perrier (2008) have defined four kinds of relations:

- Immediate dominance relations: *N* > *M* means that *M* is an immediate sub-constituent of *N*.
- Underspecified dominance relations: *N* >* *M* means that the constituent *N* includes another constituent *M* at a more or less deep level. (With this kind of node relations, long distance dependencies and possibilities of applying modifiers could be expressed.)
- Immediate precedence relations: *N* << *M* means that the constituent *M* precedes the constituent *N* immediately in the linear order of the sentence.
- Underspecified precedence relations: $N <<^{+} M$ means that the constituent *M* precedes the constituent *N* in the linear order of the sentence but the relation between them cannot be identified.

3 The Persian Language Properties

Persian is a member of the Indo-European language family and has many features in common with the other languages in this family in terms of morphology, syntax, phonology, and lexicon. Although Persian uses a modified version of the Arabic alphabet, the two languages differ from one another in many respects.

Persian is a null-subject language with SOV word order in unmarked structures. However, the word order is relatively free. The subject mood is widely used. Verbs are inflected in the language and they indicate tense and aspect, and agree with subject in person and number. The language does not make use of gender (Māhootiān, 1997).

In noun phrases, the sequence of words is around at least one noun, namely the head word. So, the noun phrase could be either a single unit noun, or a sequence of other elements with a noun. The syntax of Persian allows for having elements before a noun head _prenominal, and after the noun head _postnominal.

To make a phrase, there are some restrictions for the elements surrounding a head to make a constituent; otherwise the sequence of elements will be ill-formed, that is, ungrammatical.

Nouns belong to an open class of words. The noun could be a common noun, a proper noun, or a pronoun. If this noun is not a proper noun or a pronoun, some elements can come before it and some after it (Māhootiān, 1997). Some of the prenominal elements coming before a noun head are cardinal numbers, ordinal numbers, superlative adjectives, and indefinite determiners; post-nominal elements are nouns and noun phrases, adjectives and adjectival phrases, adjectival clauses with conjunctions, indefinite post-determiners, prepositional phrases, adverbs of place and time, ordinal numbers, possessive adjectives, and Ezafeh.

The syntactical structure of an adjectival phrase is simple. It is made up of a head adjective and elements that come before and after the head. An adjectival phrase is a modifier of a noun. The elements coming before a simple adjective are adverbs of quantity and prepositional phrases.

4 Required Tools

4.1 Test Suite

The test suite is a set of controlled data that is systematically organized and documented. In this case, the test suite is a kind of reference data different from data in large collections of text corpora. A test suite should have the following advantages: it should have a broad coverage on the structural level, so you can find many structures of a language with a minimal lexicon; it could be multilingual, so the structure of the languages could be compared; it should be a consistent and highly structured linguistic annotation. The differences between a test suite and a corpus are: that in test suite there is a control on the data, that the data has a systematic coverage, that the data has a non-redundant representation, that the data is annotated coherently, and that relevant ungrammatical constructions are included intentionally in a test suite (Oepen et al, 1996).

Since our end goal is to develop a fragment of Persian grammar, to the best of our knowledge no already developed test suite for our target constructions was available; so we built a very small test suite with only 50 examples based on a small lexicon _only 41 entries.

4.2 XMG

The XMG system is usually called a "meta-grammar compiler" is a tool for designing large-scale grammars for natural language. This system has been designed and implemented in the framework of Benoit Crabbé (2005).

XMG has provided a compact representation of grammatical information which combines elementary fragments of information to produce a fully redundant, strongly lexicalized grammar. The role of such a language is to allow us to solve two problems that arise while developing grammars: to reach a good factorization in the shared structures, and to control the way the fragments are combined.

It is possible to use XMG as a tool for both tree descriptions in IG and TAG. Since there isnot any built-in graphical representation for IG in XMG, LEOPAR is used to display the grammar. LEOPAR is a parser for processing natural languages based on the IG formalism.

4.3 LEOPAR

LEOPAR is a tool chain constructed based on IG (Guillaume et al, 2008). It is a parser for IG that can be used as a standalone parser in which inputs are sentences and outputs are constituent trees. But it also provides a graphical user interface which is mostly useful for testing and debugging during the stages of developing the grammar. The interface can be used for interactive or automated parsing. LEOPAR also provides several visualization modes for the different steps in the parsing process. Furthermore, it offers some tools to deal with lexicons: they can be expressed in a factorized way and they can be compiled to improve parsing efficiency.

LEOPAR is based on UTF8 encoding, so it supports Persian characters. It is also modified to take into account the right-to-left languages. For our designed grammar we have taken the advantage of this parser for IG.

5 Designing the Grammar

In this section we explicitly describe the tree construction of the Persian noun and adjectival phrase structures which are polarized. We have provided the elementary syntactic structures derived from the existing rules in the language and then polarized the features in the trees which are named *initial polarized tree descriptions.*

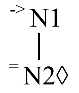
To be more comprehensible and clear, nodes are indexed for addressing. More importantly, the trees should be read from right-to-left to match the writing system in the right-to-left language.

For clarity in the tree representations in this paper, no features are given to the nodes. But while developing the grammar with XMG, polarized features are given to the nodes to put a control on constructing the trees and avoid over-generating some constructions.

There are some constructions whose tree representations are the same but represent two different constructions, so they could be described from two different points of views. Such trees are described in the sections corresponding to the relevant constructions. Some morphophonemic phenomena were considered at the syntactic level, while developing our grammar. Such a phenomenon is defined at the feature level for the lexicon which will be described in their relevant sections.

5.1 Noun Construction

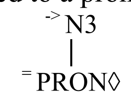
A noun phrase could consist of several elements or only one head noun element. If the element of a noun phrase (N1) is a noun, it is anchored to a lexicon item (N2) which could be a common noun, or a proper noun. The symbol ◊ has been used for the nodes that are anchored to a lexical item.



The tree of a common noun and a proper noun are the same, but features should be given to the tree to make a distinction between the anchored nouns. With the help of features, we can make some restrictions to avoid some constructions. Features and their values are not fully discussed here.

5.2 Pronoun Construction

A pronoun can appear both in subject and object positions to make a noun. In this construction, node N3 is anchored to a pronoun:



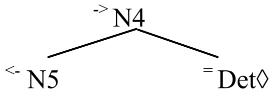
A pronoun cannot be used in all constructions. For example, N3 cannot be plugged into N5 in a determiner construction because a determiner could not come before a pronoun. To avoid this construction, some features have been used for the node N5 to stop the unification with some N nodes like N3.

5.3 Determiner Construction

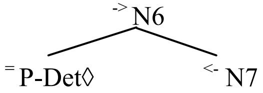
In Persian a determiner comes before a common noun or a noun phrase, and not a proper noun or a pronoun.

Persian does not benefit from the definite determiner, but there are two kinds of indefinite determiners: one comes before a noun as a separate lexical item and the other one comes after a noun (post-determiner) which is joined to the end of the noun as described below:

If the determiner comes before a noun, there must be a tree in which a Det node is anchored to a lexicon item that is a determiner and which comes immediately before a noun. In other words, some lexical items which are determiners could attach to this node:



If the determiner comes after a noun (i.e. if it is a post-determiner), then it can be joined to the end of a noun. The post-determiner (P-Det) and the preceding noun (N7), make a noun (N6):



The post-determiner has three different written forms: ‘ی’ /i/, ‘یی’ /yi/, and ‘ای’ /?i/. The reason to have them is phonological. In our formalism we have considered this phonological phenomenon at a syntactic level.

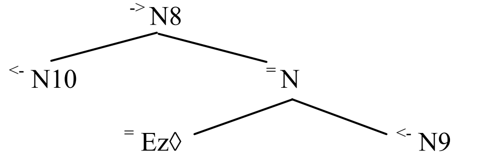
If the post-determiner construction is used after an adjective in the linguistic data, it does not belong to the adjective (since the adjective is only the modifier of the noun), but it belongs to the noun. According to the phonological context and the final sound of the adjective, the post-determiner that belongs to the noun changes and takes one of the written forms.

5.4 Ezafeh Construction

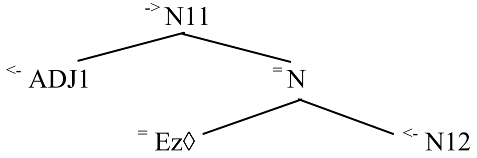
One of the properties of Persian is that usually short vowels are not written. In this language, the Ezafeh construction is represented by the short vowel ‘ـِ’ /e/ after consonants or ‘ی’ /ye/ after vowels at the end of a noun or an adjective.

Here we try to give a formal representation of such construction that is described from a purely syntactical point of view. Ezafeh (Ez) appears on (Kahnemuyipour, 2002): a noun before another noun (attributive); a noun before an adjective; a noun before a possessor (noun or pronoun); an adjective before another adjective; a pronoun before an adjective; first names before last names; a combination of the above.

Note that Ezafeh only appears on a noun when it is modified. In other words, it does not appear on a bare noun (e.g. 'کتاب' /ketāb/ 'book'). In Ezafeh construction, the node Ez is anchored to the Ezafeh lexeme. The below tree could make a noun phrase (N8) with Ezafeh construction, in which a common noun or a proper noun on N9 is followed by an Ezafeh (Ez) and another common noun, proper noun, pronoun or another noun phrase plugs to the node N10:

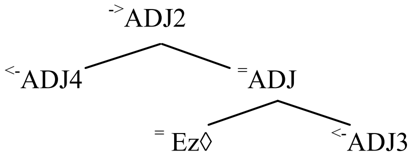


The below tree could make a noun phrase (N11) with Ezafeh construction in which a common noun or a proper noun on N12 is modified by an adjectival phrase on node ADJ1. Ezafeh has to be used after the noun to link it to the adjective:



Based on the final sound of the word which is just before Ezafeh, there are two written forms for Ezafeh, depending on whether the noun ends with a consonant or a vowel.

As we have already said, Ezafeh contraction could be used for an adjective (ADJ1). After this construction, another adjectival phrase (ADJ3 and ADJ4) with Ezafeh could appear too. It should be mentioned that ADJ4 is plugged into an adjective without Ezafeh construction:



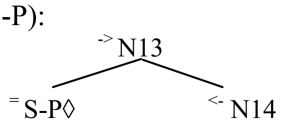
5.5 Possessive Construction

In Persian there are two different constructions for possessive. One is a separate lexical item as a common noun, a proper noun, or a pronoun. The second is a possessive pronoun that is a kind of suffix which attaches to the end of the noun. In the first construction, a noun with an Ezafeh construction is used and then a common noun, a proper noun, or a pronoun as a separate lexical item follows. In the latter construction, there is a common noun and the joined possessive pronoun. The two constructions are discussed here:

In section 5.4 we described Ezafeh construction (N8). This tree could be used for possessive construction, too. In this tree an Ezafeh is used after a common noun and Ezafeh is followed by either a common noun or a proper noun. A pronoun could not be used in N9 with Ezafeh. Such a kind of construction is avoided by defining features.

The possessive construction as a suffix could come after both a noun and an adjective. The general property of the joined possessive pronouns is that there is an agreement between the subject and the possessive pronoun in terms of number and person, no matter whether it is used after a noun or an adjective.

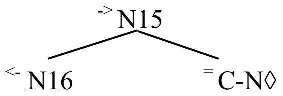
If the joined possessive pronoun (S-P) is used after a noun (N14), we would have the tree N13 in which the possessive pronoun is anchored to the suffix (S-P):



Based on the phonological reasons and considering Persian syllables, as was discussed previously in section 5.3, this suffix would have different written forms based on the phonological context it appears in: after a consonant, the vowel /ā/, or any other vowels except /ā/. For adjectives, there is no suffix possessive pronoun. In the linguistic data, this pronoun could appear after the adjective. But the point is that the adjective is only the modifier of the noun. This possessive pronoun, in fact, belongs to the noun and not the adjective, but based on the phonological rules (i.e. the final sound of the adjective) only one of the written forms would appear after that.

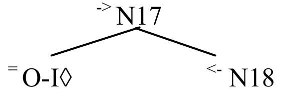
5.6 Count noun Construction

There are some nouns in Persian referred to as count nouns which have collocational relations with the head noun that is counted. So, in such a construction, the node C-N is anchored to a lexical item that is a count noun:



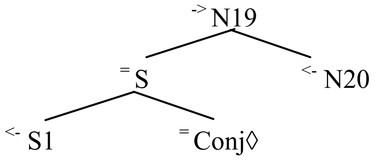
5.7 Object Construction

In Persian, a noun phrase can appear both in subject and object positions. If the noun phrase appears in a subject position, it does not require any indicator. But if the noun phrase appears in the direct object position (N18), the marker 'را' /rā/ is used to indicate that this noun phrase (N17) is a direct object. We call this marker 'Object Indicator' (O-I) so the node is anchored to the object maker. The representation of the tree for the object construction (N17) is the followings:



5.7 Conjunction Construction

In Persian, there is a construction to modify the preceding noun phrase with an adjective clause which we have named the Conjunction construction. In such a construction, there are a noun phrase (N20), a conjunctor (Conj), and a clause to modify the noun phrase (S1). In the tree, the conjunction node is anchored to a conjunctor:

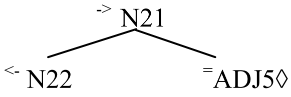


5.8 Adjective Constructions

There are two classes of adjectives: the first class comes before a noun head, the second one after.

There are three kinds of adjectives in the first class which can be differentiated from each other with the help of features. The first class of adjectives contains superlative adjectives, cardinal numbers, and ordinal numbers that modify a noun, a count noun, or a noun phrase. Usually, the adjectives coming before a noun phrase are in complementary distribution; i.e. the presence of one means the absence of the two others.

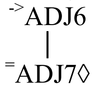
The following tree represents the adjective construction coming before a noun (N22). The adjective ADJ5 is anchored to a lexical item:



The second class of adjectives (which comes after a noun) contains mostly simple adjectives, ordinal numbers and comparative adjectives.

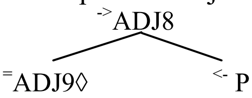
As we have already described tree N11 in section 5.4, to have an adjective after a noun the noun must have an Ezafeh construction. So, this tree represents a construction where an adjective (ADJ1) comes after a noun (N12).

To saturate ADJ1, the tree ADJ6 is required which is anchored to an adjective lexical item:

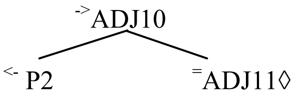


In some adjective constructions, a prepositional phrase could be used which comes before or after some adjective constituents. With the help of some features, we have made restrictions on the kind of adjective and the preposition lexical item that could plug into this node.

If a preposition is used before the adjective (ADJ9), it is a comparative adjective:

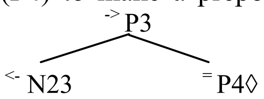


If the preposition is used after the adjective (ADJ11), it is either a comparative or a simple adjective:



5.9 Preposition Construction

In Persian a common noun, a proper noun, a pronoun, or a noun phrase could come after a preposition (P4) to make a prepositional phrase (P3):



If the preposition construction is used in an adjective construction, only some specific prepositions can be used. Once again, the restrictions are encoded with features.

6 Implementation and Results

So far we have explicitly described the noun and adjectival phrase constructions in Persian according to the constituency rules that are extracted from the linguistic data. These rules are represented by polarized trees. Since we wanted to study the noun and adjectival phrase structures, they required data. We have gathered this data for our purpose as a test suite.

To design IG for the constructions that were described, we have used XMG as the basic tool to have the initial tree descriptions. While describing the trees in XMG, several operators will be used to polarizing features. The categories of the nodes are considered as features, so the nodes are polarized. Using XMG, we have done factorizations and defined classes for general trees. Three factorized general trees are defined in our XMG coding. We have also defined 17 classes for coding of trees to represent the constructions as described.

The output of XMG is given to LEOPAR to display the graphical representations of the tree structures and also parse the data. The test suite is given to LEOPAR for parsing.

Having the developed trees and the test suite, we successfully parsed all available phrases, from the simplest to the most complex ones that had a variety of constructions in them. Example 1 has a simple construction, example 2 is of medium complexity, and example 3 is the most complex:

1. کتاب دانیال
/ketāb/ (/e/) /dāniyāl/
book (Ez) Daniel
'the book of Daniel / Daniel's book'

2. همزمان با انتشار اولین کتاب او
/hamzamān/ /bā/ /entešār/ (/e/) /avvalin/
in coincidence with publishing (Ez) the first
/ketāb/ (/e/) /?u/
book (Ez) his/her
'in coincidence with the publishing of his/her first book'

3. آن دو جلد کتاب جدید مهم دانیال را که
/ān/ /do/ /jeld/ /ketāb/ (/e/) /jadid/ (/e/)
that two volume book (Ez) new (Ez)
/mohem/ (/e/) /dāniyāl/ /rā/ /ke/
important (Ez) Daniel POBJ that
'the two new important book volumes of Daniel that'

We know from section 5.4 that Ezafeh is pronounced but not written. Since the anchored nodes require a lexical item, we put the word 'اضافه' /ezāfe/ 'Ezafeh' in the lexicon to have a real representation of Ezafeh. Also, wherever Ezafeh is used in the test suite, this word is replaced.
As a sample, we give a brief description of parsing the phrases 1 and 2 with LEOPAR and display the outputs.
In our test suite, phrase 1 is found as 'کتاب اضافه دانیال'. In this phrase, the common noun /ketāb/ is followed by a proper noun /dāniyāl/ with Ezafeh. The possessive construction (N8) would be used to parse this phrase.
In parsing this phrase, firstly LEOPAR reads the words and matches them with the lexical items available in the lexicon to identify their categories. Then it plugs these words into the nodes in the trees that have the same syntactic category and have an anchored node. Finally, it gives the parsed graphical representation of the phrase.
For this phrase, the Ezafeh construction tree is used in such a way that N2 is anchored to the word /ketāb/ and N1 plugs into N9 to saturate it. Then, N2 is again anchored to the word /dāniyāl/ and N1 plugs in to saturate N10. The final parsed phrase is such that all internal nodes are saturated and have neutral polarity, as shown in Figure 1.
As another example, consider phrase 2, which is found as 'همزمان با انتشار اضافه اولین کتاب اضافه او' in our test-suite. Since some various constructions are used to build this phrase, we could say that it

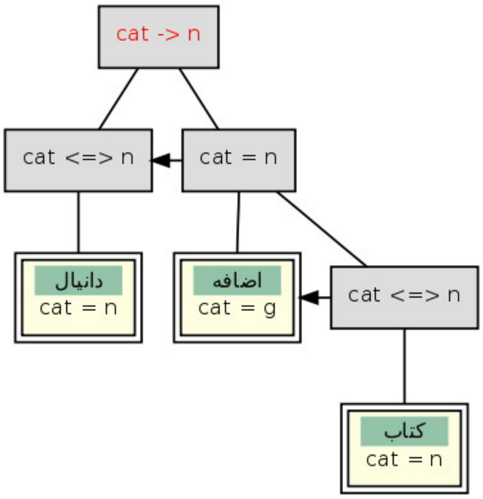


Figure 1: Parsing the phrase 'کتاب دانیال' with LEOPAR

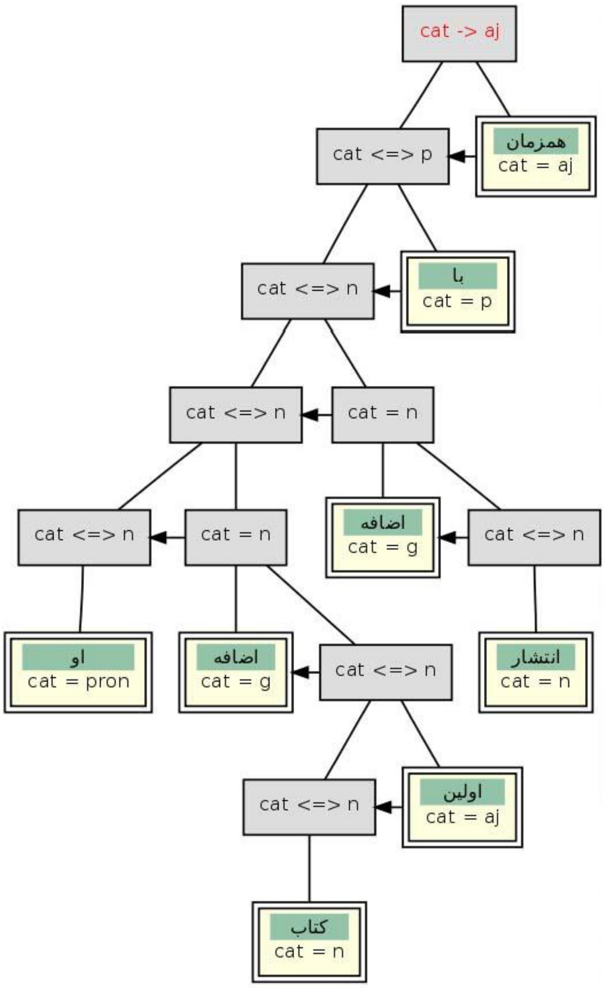


Figure 2: Parsing the phrase 'همزمان با انتشار اولین کتاب او' with LEOPAR

is a complex phrase. Firstly it takes the adjective phrase construction (ADJ10). P3, the prepositional phrase, plugs into P2. Since a noun or a noun phrase could be used after a preposition (N23), the Ezafeh construction (N8) that takes the noun plugs to this node. Another Ezafeh construction (N8) will be plugged into N10. The adjective construction (ADJ5) for ordinal numbers as the modifier of a noun (N22) could be used while a noun (N1) would plug into N22. Finally, the pronoun (N3) plugs into the unsaturated noun position in the second Ezafeh construction. Parsing the phrase with LEOPAR, the result has all internal nodes saturated and neutralized, and no polarities on the nodes are left unsaturated, as shown in Figure 2.

7 Conclusion and Future Work

In our research we have used IG to represent the construction of Persian noun and adjectival phrases in trees. XMG was used to represent the constructions using factorization and inherited hierarchy relations. Then, with the help of XMG, we defined IG by taking advantage of polarities on the features and tree descriptions for the various constructions that are introduced. Then, we used LEOPAR for the graphical representations of the trees and parsing the phrases. Finally, we applied our test suite to the parser to check whether we had the correct parsing and representation of the phrases. The experimental results showed that we could parse the phrases successfully, including the most complex ones, which have various constructions in them.

In the next step of our research, we would like to study the construction of prepositions and, more importantly, verbs in depth to make it possible to parse at the sentence level.

References

Adjukiewcz K., 1935. “Die syntaktiche konnexität” *Studia Philadelphica* 1, pp. 1-27.

Bonfante G. and B. Guillaume and G. Perrier, 2004. “Polarization and abstraction of grammatical formalism as methods for lexical disambiguation” In *Proc.s of 20th Int. Conf. on CL, Genève.*

Candito, M. H., 1996. ‘A principle-based hierarchical representa-tion of LTAGs’. *COLING-96.*

Crabbé, B., 2005. “Grammatical development with XMG”. *LACL 05.*

Duchier and Thater, 1999. “Parsing with tree descriptions: A constraint based approach” In *Proc.s of NLU and Logic Programming, New Mexico.*

Gaiffe, B. and G. Perrier, 2004. ‘Tools for parsing natural language’ *ESSLLI 2004.*

Guillaume B. and G. Perrier, 2008. “Interaction Grammars” INRIA Research Report 6621: http://hal.inria.fr/inria-00288376/

Guillaume B. and J. Le Roux and J. Marchand and G. Perrier and K. Fort and J. Planul, 2008, “A Toolchain for Grammarians” *CoLING 08, Manchester.*

Jesperson , O., 1935. *Analytic Syntax*. Allen and Uwin, London.

Kahane, S., 2004. “Grammaries d’unification polarisées” In *11iéme Conf. sur le TAL, Fés, Maroc.*

Kahane, S., 2006. “Polarized unification grammars”. In *Proce.s of 21st Int. Conf. on CL and 44th Annual Meeting of the ACL. Sydney, Australia.*

Kahnemuyipour, A., 2000. "Persian Ezafe construction revisited: Evidence for modifier phrase," *Annual Conf. of the Canadian Linguistic Association.*

Lambek, J., 1958. "The mathematics of sentence structure", *The American Mathematical Monthly* 65: 154–170.

Leopar: a parser for Interaction Grammar http://leopar.loria.fr/

Le Roux, J. and G. Perrier, 2007. “Modélisation de la coordination dans les Grammaires d’Interaction”, *Traitement Automatique des Langues* (TAL 47-3)

Māhootiān, Sh, 1997. *Persian*. Routledge.

Muskens and Krahmer, 1998. “Talking about trees and truth conditions”. In *Logical Aspects of CL, Grenoble, France*, Dec 1998.

Nasr A., 1995. “A formalism and a parser for lexicalized dependency grammars” In *Proce.s of 4th Int. Workshop on Parsing Technologies, Prague.*

Oepen, S. and K. Netter and J. Klein, 1996. “TSNLP-Test suites for natural language processing”. In *Linguistic Database*, CSLI Lecture Notes. Center for the Study of Language and information.

Perrier, G., 2000. “Interaction grammar” *Coling 2000.*

Perrier, G., 2007. "A French Interaction Grammar", *RANLP 2007, Borovets Bulgarie.*

Planul, J., 2008. *Construction d'une Grammaire d'Interaction pour l'anglais*, Master thesis, Université Nancy 2, France.

Tesnière L., 1934. “Comment construire une syntaxe” *Bulletin de la Faculté des Lettres de Strasbourg* 7-12iéme. pp. 219-229.

XMG Documentation http/wiki.loria.fr/wiki/XMG/Documentation