A Link Grammar Parser for Persian

Jon Dehdari and Deryle Lonsdale
BYU Department of Linguistics

Abstract

This paper gives an overview of an original syntactic parser for the Persian language. The parser, Persian LG, is based on Link Grammar (Sleator and Temperley 1993), a dependency-like grammar. Individual inflectional morphemes are first decomposed by a morphology component (either PC-Kimmo or Perstem), and then are syntactically linked together in an efficient and robust manner. Each component is presented in detail, with a discussion of the system's current status and possible applications.

1 Introduction

Persian is an Indo-European language with interesting morphological and syntactic properties.Verbs can express tense and aspect, and they agree with the subject in person and number. Nouns can host pronominal clitics marked for dative and genitive pronouns, as in the word بِدستت *bedæstæt* "to your hand". Verb forms like دیدیمتان *didimetān* "we saw you" host accusative pronominal clitics. Standard morphophonological changes such as epenthesis, assimilation, and deletion occur at morpheme boundaries. However, some vowel harmony also occurs, as in the word نِمی‌آیم *nemiāyæm* "I'm not coming", where the negation prefix *næ* changes to *ne* (see Mahootian 1997: 306–8). Figure 1 illustrates glossed morphological analyses for these words.

Issues naturally arise when using an orthography originally designed for a Semitic language. Some morphophonological phenomena, such as vowel harmony, do not show up in the orthography. Also, the distinction between affixes, clitics, and words

(1a) be+ dæst +æt
DAT+ hand +2.S.GEN
"to your hand"

(1b) di +d +im +etān
see +PAST +1.P.NOM +2.P.ACC
"We saw you."

(1c) næ+ mi+ ā +æm
NEG+ DUR+ come +1.S.NOM
"I am not coming."

Figure 1: Romanized Persian morphology examples: noun-enclitic dative in (1a), verb-enclitic accusative in (1b), and vowel harmony and epenthesis in (1c).

is further complicated with the use of zero-width joiners, zero-width non-joiners, word spaces, and narrow no-break spaces[1] (Megerdoomian 2000c). Since the genitive *ezāfe* marker is not normally visible in the written form, ambiguities in syntactic part-of-speech assignment and semantic roles can arise in text-based parsing.

Another consideration in parsing Persian is the morphosyntactic relation of light verb constructions. Nouns, adjectives, or prepositional phrases (among other categories) can combine with light verbs like "do" (*kærdæn*) and "have" (*dāštæn*). The resulting word pair usually derives new, non-compositional meaning. Thus دوست — داشتن *dust—dāštæn* (lit. "friend—have") means "to like". Megerdoomian (2002) offers an in-depth treatment of these constructions.

Computational processing of Persian appears to be somewhat underexplored. One notable exception is the Shiraz project[2], which employs a unification-based morphology engine (Megerdoomian 2000b) and a chart parser (Amtrup et al. 1999) for Persian-English machine translation (Amtrup et al. 2000). Another approach somewhat closer to the one we have taken underlies the Perslex engine (Riazati 1997), a Persian two-level morphology processor, the public availability of which is unclear.

[1] See also http://www.laits.utexas.edu/persian/persianword/persianwp.htm

[2] http://crl.nmsu.edu/shiraz/

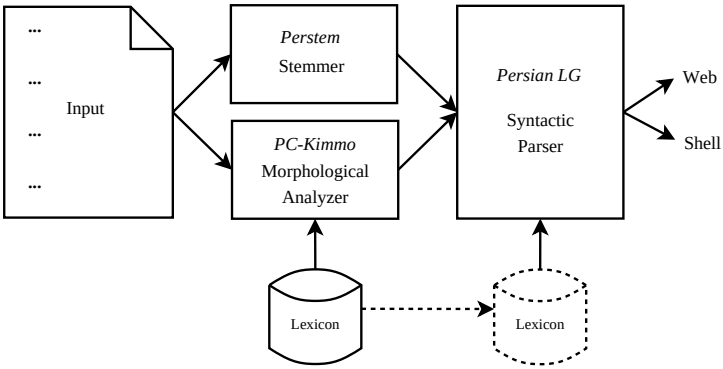

Figure 2: An overview of the parsing system.

Finally, a Persian stemming implementation has been developed for information retrieval purposes (Tashakori et al. 2002).

In this paper we discuss efforts to develop a new system for parsing Persian, called Persian LG. This enterprise was viewed as worthwhile and appropriate since it involves novel ways of integrating morphological and syntactic processing. Its modules are also built on open-source technologies, like two-level morphology and link grammar parsing, that have been used in similar applications.

2 Morphological preprocessing

The first component in our processing approach performs morphological decomposition. This can be carried out by a lexicon-dependent morphological analyzer or a lexicon-independent stemmer, as seen in Figure 2. After the input text has been morphologically decomposed, it is then syntactically parsed. The PC-Kimmo morphological analyzer and the Persian LG syntactic parser make use of a lexicon, which will be discussed below.

2.1 Morphological analyzer

The lexicon-dependent morphology engine is based on the two-level finite-state approach (Koskenniemi 1983) and uses the PC-Kimmo engine, which is capable of

```
RULE 0:i <= Vl +:0 ___ [ Pn | m A n | t A n | C A n | e:0 ] #

RULE e:0 <=> Vl:Vl +:0 0:i ___ #
```

Figure 3: Two sample PC-Kimmo rules.

recognizing (i.e. analyzing or parsing) and generating (i.e. synthesizing or composing) word-based morpheme sequences. Similar morphology engines have been developed for such languages as Arabic (Beesley 1997; Beesley and Karttunen 2003), Turkish (Oflazer 1994), Armenian (Lonsdale and Danielyan 2005), Oriya (Shabadi 2003), and many others. As mentioned above, the Shiraz (Megerdoomian 2000b) and Perslex (Riazati 1997) projects have also developed morphology engines for Persian.

Practical considerations prompted us to implement a transliteration scheme for Persian. For example, our morphology and syntax engines do not accommodate non-Roman characters. Perl software assures a straightforward conversion between commonly used character sets (e.g. ISIRI 3342[3], CP-1256[4], and UTF-8) and our romanized input/output. A strict 1:1 correspondence to the orthography of Persian allows lossless conversion.

The PC-Kimmo morphology engine has three principal components: a set of rules, a collection of morpheme lexicons, and a phrase-structure grammar. The engine mediates two levels of a word (the lexical and surface representations) via a dozen rules that specify systematic morphophonological changes. Traditional grammars were used to conceptualize and develop appropriate rules (Mace 2003; Mahootian 1997). These rules were compiled via the KGen rule compiler[5] into state transition tables which collectively specify a finite-state transducer architecture.

Figure 3 shows two sample rules. These two rules often work in tandem and with other rules to handle epenthesis, even in situations where the surrounding vowels are not visible on the surface. The first rule states that a surface letter ‘ i ’ must (⇐) delete (i.e. correspond to null) if preceded by a morpheme-final long vowel and if followed

[3] http://www.isiri.org/std/3342.htm

[4] http://www.microsoft.com/globaldev/reference/sbcs/1256.htm

[5] http://www.sil.org/pckimmo/about_pc-kimmo.html

```
PC-KIMMO>recognize bui
bu+e     smell+EZ

PC-KIMMO>recognize nmi-guiim
n+mi-+gu+im     NEG+DUR+say.PRES+1P

PC-KIMMO>recognize nmi-binmC
n+mi-+bin+m+C     NEG+DUR+see.PRES+1S+3s.object
```

Figure 4: Morphological decomposition for three words of varying complexity.

by any of the various suffixes listed, including an *ezāfe* particle. The second rule, an iff (⇔) rule, complements the first by always deleting the *ezāfe* particle (lexical 'e') when word-final and when preceded by a morpheme-final long vowel and an epenthesized 'i'. So بوی bui *buye* "smell of" might appear in a text and would resolve to ـِ + بو bu+e at the lexical level. This knowledge substantially contributes to the subsequent syntactic parsing stage.

A lexicon licenses valid morpheme sequences and specifies various properties of lexical and grammatical morphemes (Antworth 1990). The lexicon system contains seven different repositories for three types of morpheme-related information: fully vowelled lexical forms, English glosses, and featural/constraint information. Each lexical category (V, N, P, A) has its own lexicon; other lexicons have been developed for affixes, proper nouns, and function words (prepositions, conjunctions, numbers, etc.). Unfortunately, no freely usable and easily adaptable machine-readable Persian lexicon was readily available during development. In our case lexical information was derived from standard reference dictionaries, such as Steingass (1892).

The third component, the word grammar, permits display of a word's morphological structure in hierarchical format. A set of 16 context-free word-formation rules specify and constrain Persian lexical substructure.

Recognition of a word returns all parses in three possible formats: a sequence of lexical morphemes, a corresponding sequence of English morphemic glosses, and an optional word-structure parse tree. The top part of Figure 4 displays recognition results for the surface form bui , which is lexically bu+e. Gloss information is also displayed ("smell+EZ"). Figure 4 (middle) recognizes نمی‌گوییم, *nemiguyim* "We are not saying". The first rule of Figure 3 epenthesizes a ـیـ *y* between the present tense

	Baseline	*PC-Kimmo*	*Perstem*
Accuracy	–	96%	91%
Coverage	82%	92%	97%

Table 1: Evaluation of the two morphological analyzers

verb root گو *gu* "say" and the first person plural suffix یم *im*. A rule governing vowel harmony with the negative prefix نَـ *næ* would have also been employed, had the text been fully vowelled. The final part of Figure 4 shows recognition of a complex verbal form, نمی‌بینمش *nemibinæmeš* "I don't see it" .

The morphology engine has undergone considerable development, but some work remains to be done. Less commonly needed morphophonological rules still have to be written, and of course more lexicon development is necessary to extend coverage. In spite of this limited lexicon, recent evaluations have shown promising results. We used a random sampling of 500 unseen words from corpora that we built from Kayhan news[6] and BBC Persian news[7]. When words had multiple parses, the first parse was used. When words were not recognized, the word was taken in its entirety. The baseline, where no words were morphologically decomposed, was correct 82% of the time. More than 92% of the test words were morphologically analyzed correctly, as is seen in Table 1. Of the words which were morphologically decomposed, 96% were analyzed correctly.

2.2 Stemmer

While the Persian PC-Kimmo engine provides excellent accuracy, it is currently unable to perform morphological operations on words not found in its lexicon. We developed a lexicon-independent stemmer/shallow morphological parser that can be used as an alternative to the PC-Kimmo engine, or in conjunction with it. The stemmer, Perstem, is written in Perl and uses regular expression substitutions to separate

[6]http://www.kayhannews.ir

[7]http://www.bbc.co.uk/persian

نمی‌گوییمتان → nmi-guiimtAn → n+_mi-+_gu_+0_+im_+tAn → n mi gu im tAn

نمی‌گویمتان → nmi-guimtAn → n+_mi-+_gu_+0_+m_+tAn → n mi gu m tAn

کتاب‌های → ktAb-hAi → ktAb_+-hA_+e → ktAb hA e

Figure 5: The stages of stemming the words *nemi-guyimetān*, *nemi-guyæmetān*, and *ketāb-hāye*.

inflectional morphemes, and optionally remove affixes.[8] The stemmer currently has 76 substitution rules, which replace one pattern of text with another.

Figure 5 shows the decomposition stages for the verbs *nemi-guyimetān* "we do not tell you", *nemi-guyæmetān* "I do not tell you", and the nominal fragment *ketāb-hāye* "books of". The morphemes in the final stage serve as the input for the syntactic parser, where they are linked with other words and morphemes. This ensures agreement between features, such as number and person.

Currently Perstem is the primary means of morphological decomposition for Persian LG, due to its flexibility and robustness. Perstem can process about 12200 words per second on an UltraSPARC-III machine, ten times faster than the PC-Kimmo analyzer. Using the previously mentioned testing words, Perstem correctly analyzed 97% of the words.[9] Of the words that were morphologically decomposed, 91% were analyzed correctly. The use of a lexicon clearly helps eliminate incorrect analyses, but requires more processing time and extensive development time. Perstem's coverage is comparable with Megerdoomian (2004), as is the accuracy of Persian PC-Kimmo. Preliminary testing has shown that integrating the PC-Kimmo engine with Perstem markedly increases coverage with only a small loss in accuracy compared with the PC-Kimmo results. Tashakori et al. (2002) uses stemming evaluation metrics that are not comparable with the aims of this paper.

[8] Perstem may be downloaded at http://sourceforge.net/projects/perstem

[9] Evaluation data are found at http://ling.ohio-state.edu/~jonsafari

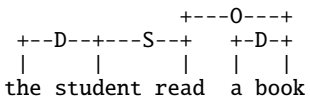

Figure 6: A sample English sentence parsed in link grammar.

3 A Persian parsing engine

After the morphological parsing is completed, the resulting output is then transferred to a parser for syntactic processing. Thus a distinction is drawn between morphology and syntax in our integrated system, with either PC-Kimmo or Perstem processing morphophonological material, and the parser treating morphosyntactic material.

We have chosen to use the the link grammar (LG) engine, an efficient, freely available parser originally developed for English (Sleator and Temperley 1993; Grinberg et al. 1995). Dependency-style parsing has seen a resurgence in recent years, and link grammar provides a well-developed framework for determining dependencies between words (Schneider 1998). It is more robust than traditional parsers and has been widely used in such NLP applications as information retrieval, speech recognition, and machine translation.[10] Written in the C programming language, it is comparatively fast and efficient (Sleator and Temperley 1993: 10–11).

The link grammar parser does not seek to construct constituents in the traditional linguistic sense; instead, it calculates simple, explicit relations between pairs of words. A link is a targeted relationship between two words and has two parts: a left side and a right side. For example, links associate such word pairs as: subject + verb, object + verb, preposition + object, adverbial modifier + adjective, and auxiliary + main verb. Each link has a label that expresses the nature of the relationship mediating the two words, as is seen in Figure 6. Potential links are specified by a set of technical rules that constrain and permit word-pair associations. In addition, it is possible to score individual linkages and to penalize unwanted links.

The Persian LG parser implementation, apparently the first for this language, follows the basic LG strategy with one important exception. Incoming words in the

[10] See bibliography at http://link.cs.cmu.edu/link/papers/

English parser are intact; the incoming Persian words, on the other hand, are morphologically decomposed via the morphological component described in the previous section. Links are therefore established between individual inflectional and lexical morphemes rather than only between separate words.

We developed a new inventory of Persian links; in principle they follow the basic premises of LG parser construction, even though many names vary from those in English. Of course, there are morphological links in the Persian system whereas the English system has none. Processing of coordinated structures was also redone for Persian for transparency purposes. A recursive algorithm parses any number of conjunctions for nominative noun phrases, accusative noun phrases, prepositional phrases, predicate adjectives, or complement phrases.

The LG parser, besides including specifications for link construction, also makes use of a set of lexicons for various word categories. Since the morphology and syntax parsing engines are pipelined, lexicon coverage across both engines obviously needs to be consistent. To assure this, a lexical database containing full vocalizations is used to generate lexicons for both engines, annotating the entries as necessary with relevant information. For the morphology engine, this includes features, categories, glosses, and lexicon membership. For the syntax engine categories, paradigmatic information and valency information are specified.

While the LG parser makes use of lexicon sets, it can guess an unknown word's category by using surrounding information. The system is currently configured to guess an unknown word's part of speech as either a noun, verb (non-light), or the non-verbal element of a light verb construction—all large open-class word sets. Thus when the parser encounters a word not found in its lexicon, it will try assigning the word one of these parts of speech. When multiple parses are grammatical, the parser prefers ones that result in the lowest cost vector (see Casbeer et al. fc: § 5).

In the two sentences of Figure 7, Perstem has successfully separated the inflectional morphemes of the final word in part *b* of both sentences. The verb, generally in the final position of the sentence, ends in the third-person plural suffix *-im* and finds no subject with matching person and number features, such as *mā* "we". This rules out a subject link (S) for either of the preceding words. Since only known light verbs can form light verb construction links (K) with non-verbal elements, the first

```
a.      fdsaf fdsafasf fdsafftim
b.      fdsaf fdsafasf fdsaff t im
             +-----------On-----------+------VMP------+
c.           +-----M-----+            +---VMT--+      |
             |           |            |        |      |
        fdsaf[?].n fdsafasf[?].n fdsaff[?].v t.vmt im.vmp

a.      fdsaf fdsafasf krdim
b.      fdsaf fdsafasf kr d im
             +----------On----------+-----VMP----+
c.           |            +----K----+-VMT-+      |
             |            |         |     |      |
        fdsaf[?].n fdsafasf[?].nk kr.vk d.vmt im.vmp
```

Figure 7: Two sentences containing unknown words.

sentence is grammatical only if an object link (O) is formed between the first and third words. Lastly, the parser guesses the second word to be a noun and forms an *ezāfe* link (M) between the first and second words so that the entire sentence is grammatically linked. The second sentence can receive this same parse; however, link grammar allows for preferred and penalized links. We have configured light verbs to prefer forming light verb construction links with nouns.[11] Thus the highest ranked parse for the second sentence uses the light verb construction (K) link.

Since the parser handles not only links between words, but also inflectional morphemes, all of the morphological links in the parse are marked with an M in their names. Figure 8 shows link grammar parses for two Persian sentences based on Megerdoomian (2000a). The first sentence, فاطمه و زن زیبای داریوش وارد شده اند *Fatemeh and Dariush's pretty wife have entered*, displays the linking of coordinated constituents in a complex noun phrase (Megerdoomian 2000a), a light verb construction, the present perfect tense, and the parser's ability to handle unknown words. The second sentence, سخت است ولی آسانتر خواهد شد *It's hard, but it will get easier*, uses a conjunction at the phrasal complement level, ولی `uli` *væli* "but". The adjective آسان `]sAn` *āsān* "easy" morphologically links to the comparative suffix تر `tr` *-tær*, and the future tense, خواهد `xuAhd` *xāhæd*.

[11] However a noun followed by the specific accusative marker *rā* will always form an object link.

```
%  getmorpholg "fATmh u zn zibAi dAriuC uArd Cdh |nd"
fATmh u zn zibA e dAriuC uArd C d h |nd

             +--------------------------------CCS--------------------------------+
             |                                                          +----VMPP----+      |
     +-CCNl--+CCNR+--M--+--EZ--+----M---+          +---K---+-VMT-+      +-VPPP-+
     |       |    |     |      |        |          |       |     |      |      |
fATmh.prop u.cc zn.n zibA.aj e.ez dAriuC.prop uArd[?].nk C.vkp d.vmt h.pspt |nd.perf

   u.cc           CCS     <---CCS--->  CCS       |nd.perf
   fATmh.prop     CCNl    <---CCNl-->  CCNl      u.cc
   u.cc           CCNR    <---CCNR-->  CCNR      zn.n
   zn.n           M       <---M----->  M         zibA.aj
   zibA.aj        EZ      <---EZ---->  EZ        e.ez
   e.ez           M       <---M----->  M         dAriuC.prop
   h.pspt         VPPP    <---VPPP-->  VPPP      |nd.perf
   C.vkp          VMPP    <---VMPP-->  VMPP      h.pspt
   uArd[?].nk     K       <---K----->  K         C.vkp
   C.vkp          VMT     <---VMT--->  VMT       d.vmt


% getmorpholg "sxt |st uli ]sAntr xuAhd Cd."
sxt |st uli ]sAn tr xuAh d C d .

    +-----------------------------------Xp-----------------------------------+
    |                       +------------------CC------------------+        |
    |                       |      +---------------P---------------+        |
    +----------Wcc----------+      |               +-----VFUT-----+        |
    |        +---P--+--CCF--+      +--AJM--+       +---VMP--+     +-VMT+   |
    |        |      |      |      |       |       |        |     |    |   |
LEFT-WALL sxt.aj |st.vip uli.cc ]sAn.aj tr.ajm xuAh.future d.vmp C.vi d.vmt .
```

Figure 8: System input/output, first through PC-Kimmo, then through the LG parser for two sentences:

فاطمه و زن زیبای داریوش وارد شده اند
Fatemeh and Dariush's pretty wife have entered (top, with individual links listed);

سخت است ولی آسانتر خواهد شد. *It's hard, but it will get easier.* (bottom).

3.1 Current Status

Persian LG currently has approximately 60 link types, or grammatical categories, specified in a file of about 850 lines. The lexicon contains about 3500 words in 25 different categories.

The parser works well for shorter-length sentences; longer ones such as those found in the previously mentioned news corpora (Kayhan and BBC) have not yet been extensively tested. In a recent test of 177 news corpus sentences (which took 17 seconds to process on a Sun UltraSPARC II), 101 sentences succeeded in parsing; the remainder was due largely to shortcomings in the morphology analysis as it currently stands. By our metrics 81% of the parsed sentences were annotated correctly. We anticipate performing more exhaustive and methodical testing in the near future. Unfortunately there is still, to our knowledge, no publicly available gold standard for parsed sentences in Persian, so no comparative evaluation is possible at this point. What's more, we are aware of no published evaluation results from the previously mentioned parsers. While development of a benchmark parsed corpus and associated annotations is beyond the scope of this paper, we believe this would be a worthwhile activity, and the software we provide as a result of this work should be helpful in such an effort.

Some *wh*-constructions have yet to be addressed, as well as infrequently occurring word orderings. One implementation issue that has yet to be addressed is morphological ambiguity. Whereas the morphology engine generates all possible analyses, only the first one is pipelined to the parser. Fortunately for Persian this has not proved very problematic.

The PC-Kimmo engine and the LG parser, though usable independently, have been seamlessly integrated into the Soar cognitive modeling system (Newell 1990). Previous work has integrated the English LG parser into the non-linguistic Soar environment in order to provide a back-end shallow semantic processor. Several applications have been built on top of the English system including student essay rating (Lonsdale and Strong-Krause 2003), named entity extraction (Lonsdale et al. 2001), and text extraction from newspaper headlines and biomedical information (Lonsdale et al. 2006). A possible direction for future work is to pass all of the morphological parses to the syntactic parser, and then to the agent-based semantic processor.

4 Conclusions

The modular system we have described integrates efficient morphology engines with a robust syntactic parser. This is significant because many of the difficulties in processing Persian, such as orthographical and morphological ambiguity can be resolved in the morphological component before reaching the syntactic parser. We foresee its possible use in a wide variety of applications such as language pedagogy, information retrieval and extraction, corpus tools, online dictionaries, and speech-based interfaces.

References

Amtrup, J. W., Megerdoomian, K., and Zajac, R. (1999). Rapid development of translation tools. In *Proceedings of Machine Translation Summit VII*, pages 385–389.

Amtrup, J. W., Rad, H. M., Megerdoomian, K., and Zajac, R. (2000). Persian-English machine translation: An overview of the Shiraz project. Memoranda in Computer and Cognitive Science MCCS-00-319, Computing Research Lab, New Mexico State University.

Antworth, E. (1990). *PC-KIMMO: A two-level processor for morphological analysis*. Number 16 in Occasional Publications in Academic Computing. Summer Institute of Linguistics, Dallas, TX.

Beesley, K. (1997). Finite-state descriptions of Arabic morphology. In *Proceedings of the Second Cambridge Conference: Bilingual Computing in Arabic and English*, Literary and Linguistic Computing Center, Cambridge University, UK.

Beesley, K. and Karttunen, L. (2003). *Finite state morphology*. CSLI Publications, Stanford University.

Casbeer, W., Dehdari, J., and Lonsdale, D. (fc). A link grammar parser for Arabic. In Mughazy, M., editor, *Perspectives on Arabic Linguistics*, volume 20, Amsterdam. John Benjamins.

Grinberg, D., Lafferty, J., and Sleator, D. (1995). A robust parsing algorithm for Link Grammars. Technical Report CMU-CS-95-125, School of Computer Science.

Koskenniemi, K. (1983). Two-level model for morphological analysis. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, pages 683–685.

Lonsdale, D. and Danielyan, I. (2005). A two-level implementation for Western Armenian morphology. *Annual of Armenian Linguistics*, 24–25:35–51.

Lonsdale, D., Hutchison, M., Richards, T., and Taysom, W. (2001). An NLP system for extracting and representing knowledge from abbreviated text. In *Proceedings of the Deseret Language and Linguistics Society 2001 Symposium*.

Lonsdale, D. and Strong-Krause, D. (2003). Automated rating of ESL essays. In *Workshop on Building Educational Applications with Natural Language Processing, HLT/NAACL-03*. Association for Computational Linguistics.

Lonsdale, D., Tustison, C., Parker, C., and Embley, D. W. (2006). Formulating queries for assessing clinical trial eligibility. In *Proceedings of the 11th International Conference on Applications of Natural Language to Information Systems*, number 3999 in Lecture Notes in Computer Science, pages 82–93. Springer, Berlin.

Mace, J. (2003). *Persian Grammar: For reference and revision*. RoutledgeCurzon, London.

Mahootian, S. (1997). *Persian*. Descriptive Grammars. Routledge, London.

Megerdoomian, K. (2000a). A computational analysis of the Persian noun phrase. Memoranda in Computer and Cognitive Science MCCS-00-321, Computing Research Lab, New Mexico State University.

Megerdoomian, K. (2000b). Persian computational morphology: A unification-based approach. Memoranda in Computer and Cognitive Science MCCS-00-320, Computing Research Lab, New Mexico State University.

Megerdoomian, K. (2000c). Processing Persian text: Tokenization in the Shiraz project. Memoranda in Computer and Cognitive Science MCCS-00-322, Computing Research Lab, New Mexico State University.

Megerdoomian, K. (2002). *Beyond Words and Phrases: A Unified Theory of Predicate Composition*. PhD thesis, University of Southern California.

Megerdoomian, K. (2004). Finite-state morphological analysis of Persian. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING)*, University of Geneva.

Newell, A. (1990). *Unified Theories of Cognition*. Harvard University Press.

Oflazer, K. (1994). Two-level description of Turkish morphology. *Literary and Linguistic Computing*, 9(2).

Riazati, D. (1997). *Computational Analysis of Persian Morphology*. PhD thesis, Royal Melbourne Institute of Technology.

Schneider, G. (1998). A linguistic comparison of constituency, dependency and link grammar. Master's thesis, University of Zurich.

Shabadi, K. R. (2003). Finite state morphological processing of Oriya verbal forms. In *Proceedings of EACL-2003 Workshop on Computational Linguistics for the Languages of South Asia: Expanding Synergies with Europe*, pages 49–56. Association for Computational Linguistics.

Sleator, D. and Temperley, D. (1993). Parsing English with a Link Grammar. In *Third International Workshop on Parsing Technologies*.

Steingass, F. J. (1892). *A Comprehensive Persian-English Dictionary*. Routledge and K. Paul.

Tashakori, M., Meybodi, M. R., and Oroumchian, F. (2002). Bon: The Persian stemmer. In *Proceedings of the First Eurasia Conference on Advances in Information and Communication Technology*, pages 487–494.