

A Reference Dependency Bank for Analyzing Complex Predicates

Tafseer Ahmed, Miriam Butt, Annette Hautli and Sebastian Sulger

Universität Konstanz

May 25th, 2012
LREC 2012

Context of Work

- computational LFG grammar in development in Konstanz

Context of Work

- computational LFG grammar in development in Konstanz
- aim: large-scale LFG grammar for parsing Urdu/Hindi ([Bögel et al. 2009], [Butt and King 2007])

Context of Work

- computational LFG grammar in development in Konstanz
- aim: large-scale LFG grammar for parsing Urdu/Hindi ([Bögel et al. 2009], [Butt and King 2007])
- grammar is part of the ParGram project

Context of Work

- computational LFG grammar in development in Konstanz
- aim: large-scale LFG grammar for parsing Urdu/Hindi ([Bögel et al. 2009], [Butt and King 2007])
- grammar is part of the ParGram project
 - ▶ collaborative, world-wide research project

Context of Work

- computational LFG grammar in development in Konstanz
- aim: large-scale LFG grammar for parsing Urdu/Hindi ([Bögel et al. 2009], [Butt and King 2007])
- grammar is part of the ParGram project
 - ▶ collaborative, world-wide research project
 - ▶ devoted to developing *parallel* LFG grammars for a variety of languages

Context of Work

- computational LFG grammar in development in Konstanz
- aim: large-scale LFG grammar for parsing Urdu/Hindi ([Bögel et al. 2009], [Butt and King 2007])
- grammar is part of the ParGram project
 - ▶ collaborative, world-wide research project
 - ▶ devoted to developing *parallel* LFG grammars for a variety of languages
 - ▶ features and analyses are kept parallel for easy transfer between languages

Context of Work

- computational LFG grammar in development in Konstanz
- aim: large-scale LFG grammar for parsing Urdu/Hindi ([Bögel et al. 2009], [Butt and King 2007])
- grammar is part of the ParGram project
 - ▶ collaborative, world-wide research project
 - ▶ devoted to developing *parallel* LFG grammars for a variety of languages
 - ▶ features and analyses are kept parallel for easy transfer between languages
 - ▶ languages involved:

Context of Work

- computational LFG grammar in development in Konstanz
- aim: large-scale LFG grammar for parsing Urdu/Hindi ([Bögel et al. 2009], [Butt and King 2007])
- grammar is part of the ParGram project
 - ▶ collaborative, world-wide research project
 - ▶ devoted to developing *parallel* LFG grammars for a variety of languages
 - ▶ features and analyses are kept parallel for easy transfer between languages
 - ▶ languages involved:
 - large-scale: English, German, French, Japanese, Norwegian

Context of Work

- computational LFG grammar in development in Konstanz
- aim: large-scale LFG grammar for parsing Urdu/Hindi ([Bögel et al. 2009], [Butt and King 2007])
- grammar is part of the ParGram project
 - ▶ collaborative, world-wide research project
 - ▶ devoted to developing *parallel* LFG grammars for a variety of languages
 - ▶ features and analyses are kept parallel for easy transfer between languages
 - ▶ languages involved:
 - large-scale: English, German, French, Japanese, Norwegian
 - smaller-scale (yet...): Welsh, Georgian, Hungarian, Turkish, Chinese, **Urdu** (among many others)

Complex Predicates?

- Urdu has about 700 basic verbs

Complex Predicates?

- Urdu has about 700 basic verbs
- vast majority of verbal predicates is constructed using complex predicates (CPs)

Complex Predicates?

- Urdu has about 700 basic verbs
- vast majority of verbal predicates is constructed using complex predicates (CPs)
- most other South Asian languages make use of CPs as well

Complex Predicates?

- Urdu has about 700 basic verbs
- vast majority of verbal predicates is constructed using complex predicates (CPs)
- most other South Asian languages make use of CPs as well
- knowing how to deal with CPs is essential for doing parsing/NLP for Hindi/Urdu and for South Asian languages in general

Complex Predicates?

- Urdu has about 700 basic verbs
 - vast majority of verbal predicates is constructed using complex predicates (CPs)
 - most other South Asian languages make use of CPs as well
 - knowing how to deal with CPs is essential for doing parsing/NLP for Hindi/Urdu and for South Asian languages in general
- provide a reference dependency bank that can guide teams working on NLP applications for South Asian languages (or really any language that has CPs)

Overview

- 1 Complex Predicates
- 2 Types of Complex Predicates
- 3 A Reference Dependency Bank for CPs
- 4 Conclusion

Overview

- 1 Complex Predicates
- 2 Types of Complex Predicates
- 3 A Reference Dependency Bank for CPs
- 4 Conclusion

Complex Predicates in General

- combinations of two or more predicates that predicate as a single unit

Complex Predicates in General

- combinations of two or more predicates that predicate as a single unit
- the arguments of the CP members map onto a monoclausal syntactic structure [Butt 1995]

Complex Predicates in General

- combinations of two or more predicates that predicate as a single unit
- the arguments of the CP members map onto a monoclausal syntactic structure [Butt 1995]
 - ▶ verb+verb, noun+verb, adj+verb, morphological causative

Complex Predicates in General

- combinations of two or more predicates that predicate as a single unit
- the arguments of the CP members map onto a monoclausal syntactic structure [Butt 1995]
 - ▶ verb+verb, noun+verb, adj+verb, morphological causative
 - ▶ examples from Urdu: 'memory (N) do (V)' = 'remember', 'telephone (N) do (V)' = 'telephone', 'fear (N) come (V)' = 'fear', 'throw (V) give (V)' = 'throw away'

Complex Predicates in General

- combinations of two or more predicates that predicate as a single unit
- the arguments of the CP members map onto a monoclausal syntactic structure [Butt 1995]
 - ▶ verb+verb, noun+verb, adj+verb, morphological causative
 - ▶ examples from Urdu: 'memory (N) do (V)' = 'remember', 'telephone (N) do (V)' = 'telephone', 'fear (N) come (V)' = 'fear', 'throw (V) give (V)' = 'throw away'
- often analyzed on a par with control constructions/auxiliaries/modal verbs, but:

Complex Predicates in General

- combinations of two or more predicates that predicate as a single unit
- the arguments of the CP members map onto a monoclausal syntactic structure [Butt 1995]
 - ▶ verb+verb, noun+verb, adj+verb, morphological causative
 - ▶ examples from Urdu: 'memory (N) do (V)' = 'remember', 'telephone (N) do (V)' = 'telephone', 'fear (N) come (V)' = 'fear', 'throw (V) give (V)' = 'throw away'
- often analyzed on a par with control constructions/auxiliaries/modal verbs, but:
- their syntax & semantics in fact differs markedly from these constructions [Butt 2010]

Overview

- 1 Complex Predicates
- 2 Types of Complex Predicates**
- 3 A Reference Dependency Bank for CPs
- 4 Conclusion

A Noun+Verb Complex Predicate

- formed by combining a noun and a verb

A Noun+Verb Complex Predicate

- formed by combining a noun and a verb
 - ▶ noun uninflected, light verb inflected

A Noun+Verb Complex Predicate

- formed by combining a noun and a verb
 - ▶ noun uninflected, light verb inflected
- both contribute to overall argument structure of clause

A Noun+Verb Complex Predicate

- formed by combining a noun and a verb
 - ▶ noun uninflected, light verb inflected
- both contribute to overall argument structure of clause
 - ▶ 1 argument from noun

A Noun+Verb Complex Predicate

- formed by combining a noun and a verb
 - ▶ noun uninflected, light verb inflected
- both contribute to overall argument structure of clause
 - ▶ 1 argument from noun
 - ▶ 2 arguments from verb

A Noun+Verb Complex Predicate

- formed by combining a noun and a verb
 - ▶ noun uninflected, light verb inflected
- both contribute to overall argument structure of clause
 - ▶ 1 argument from noun
 - ▶ 2 arguments from verb
 - ▶ combine into 3 arguments in resulting CP

A Noun+Verb Complex Predicate

- formed by combining a noun and a verb
 - ▶ noun uninflected, light verb inflected
- both contribute to overall argument structure of clause
 - ▶ 1 argument from noun
 - ▶ 2 arguments from verb
 - ▶ combine into 3 arguments in resulting CP

- example: *Dar lag* 'be frightened by'

nAdiyah kO hATHI sE Dar lag-A

Nadya.F.Sg Dat elephant.M.Sg Inst fear.M.Sg attach-Perf.M.Sg

'Nadya was frightened by the elephant.'

A Noun+Verb Complex Predicate

- formed by combining a noun and a verb
 - ▶ noun uninflected, light verb inflected
- both contribute to overall argument structure of clause
 - ▶ 1 argument from noun
 - ▶ 2 arguments from verb
 - ▶ combine into 3 arguments in resulting CP

- example: *Dar lag* 'be frightened by'

nAdiyah kO hATHI sE Dar lag-A
Nadya.F.Sg Dat elephant.M.Sg Inst fear.M.Sg attach-Perf.M.Sg
'Nadya was frightened by the elephant.'

(*lag* 'attach': thing attached and thing that it is attached at; *Dar*
'fear': thing that is feared)

A Noun+Verb Complex Predicate

"nAdiyah kO hATHI sE Dar lagA"

PRED	'lag<[1:nAdiyah], 'Dar<[21:hATHI]>'>'										
SUBJ	<table><tr><td>PRED</td><td>'nAdiyah'</td></tr><tr><td>NTYPE</td><td>[NSEM [PROPER [PROPER-TYPE name]]]</td></tr><tr><td>NSYN</td><td>proper</td></tr><tr><td>SEM-PROP</td><td>[SPECIFIC +]</td></tr><tr><td>1</td><td>CASE dat, GEND fem, NUM sg, PERS 3</td></tr></table>	PRED	'nAdiyah'	NTYPE	[NSEM [PROPER [PROPER-TYPE name]]]	NSYN	proper	SEM-PROP	[SPECIFIC +]	1	CASE dat, GEND fem, NUM sg, PERS 3
PRED	'nAdiyah'										
NTYPE	[NSEM [PROPER [PROPER-TYPE name]]]										
NSYN	proper										
SEM-PROP	[SPECIFIC +]										
1	CASE dat, GEND fem, NUM sg, PERS 3										
OBJ	<table><tr><td>PRED</td><td>'Dar'</td></tr><tr><td>NTYPE</td><td>[NSEM [COMMON count]]</td></tr><tr><td>NSYN</td><td>common</td></tr><tr><td></td><td>[CASE nom, CLAUSE-TYPE decl, GEND masc, NUM sg, PASSIVE -]</td></tr></table>	PRED	'Dar'	NTYPE	[NSEM [COMMON count]]	NSYN	common		[CASE nom, CLAUSE-TYPE decl, GEND masc, NUM sg, PASSIVE -]		
PRED	'Dar'										
NTYPE	[NSEM [COMMON count]]										
NSYN	common										
	[CASE nom, CLAUSE-TYPE decl, GEND masc, NUM sg, PASSIVE -]										
OBL	<table><tr><td>PRED</td><td>'hATHI'</td></tr><tr><td>NTYPE</td><td>[NSEM [COMMON count]]</td></tr><tr><td>NSYN</td><td>common</td></tr><tr><td>21</td><td>CASE inst, GEND masc, NUM sg, PERS 3</td></tr></table>	PRED	'hATHI'	NTYPE	[NSEM [COMMON count]]	NSYN	common	21	CASE inst, GEND masc, NUM sg, PERS 3		
PRED	'hATHI'										
NTYPE	[NSEM [COMMON count]]										
NSYN	common										
21	CASE inst, GEND masc, NUM sg, PERS 3										
LEX-SEM	[AGENTIVE -, GOAL +]										
TNS-ASP	[ASPECT perf, MOOD indicative]										
VTTYPE	[COMPLEX-PRED nv]										
104	[CLAUSE-TYPE decl, PASSIVE -]										

Figure: F-Structure for *nAdiyah kO hATHI sE Dar lagA* 'Nadya was frightened by the elephant.'

A Permissive Complex Predicate

- $V+V$ complex predicate

A Permissive Complex Predicate

- V+V complex predicate
 - ▶ infinitival main verb

A Permissive Complex Predicate

- V+V complex predicate
 - ▶ infinitival main verb
 - ▶ finite light verb

A Permissive Complex Predicate

- V+V complex predicate
 - ▶ infinitival main verb
 - ▶ finite light verb
- both verbs contribute to overall argument structure of clause

A Permissive Complex Predicate

- V+V complex predicate
 - ▶ infinitival main verb
 - ▶ finite light verb
- both verbs contribute to overall argument structure of clause
 - ▶ 2 arguments from main verb

A Permissive Complex Predicate

- V+V complex predicate
 - ▶ infinitival main verb
 - ▶ finite light verb
- both verbs contribute to overall argument structure of clause
 - ▶ 2 arguments from main verb
 - ▶ 2 arguments from light verb

A Permissive Complex Predicate

- V+V complex predicate
 - ▶ infinitival main verb
 - ▶ finite light verb
- both verbs contribute to overall argument structure of clause
 - ▶ 2 arguments from main verb
 - ▶ 2 arguments from light verb
 - ▶ combine into 3 arguments in resulting CP

A Permissive Complex Predicate

- V+V complex predicate
 - ▶ infinitival main verb
 - ▶ finite light verb
- both verbs contribute to overall argument structure of clause
 - ▶ 2 arguments from main verb
 - ▶ 2 arguments from light verb
 - ▶ combine into 3 arguments in resulting CP
- example: *dEkH dE* 'let see'

nAdiyah nE yAsIn kO kitAb dEkH-nE d-l
Nadya.F.Sg Erg Yassin.M.Sg Dat book.F.Sg see-Inf.M.Sg give-Perf.F.Sg
'Nadya let Yassin look at the book.'

(*dEkH* 'see': seer and seen item, *dE* 'give': permitter and action permitted)

Permissive Complex Predicate

"nAdiyah nE yAsIn kO kitAb dEkHnE dI"

PRED	'dE<[1:nAdiyah], 'dEkH<[21:yAsIn], [41:kitAb]>'>']
SUBJ	[
	[PRED 'nAdiyah']
	[NTYPE [NSEM [PROPER [PROPER-TYPE name]]] [NSYN proper] SEM-PROP [SPECIFIC +] 1[CASE erg, GEND fem, NUM sg, PERS 3]]
OBJ-GO	[
	[PRED 'yAsIn']
	[NTYPE [NSEM [PROPER [PROPER-TYPE name]]] [NSYN proper] SEM-PROP [SPECIFIC +] 21[CASE dat, GEND masc, NUM sg, PERS 3]]
OBJ	[
	[PRED 'kitAb']
	[NTYPE [NSEM [COMMON count]] [NSYN common] 41[CASE nom, GEND fem, NUM sg, PERS 3]]
LEX-SEM	[AGENTIVE +, GOAL +]
TNS-ASP	[ASPECT perf, MOOD indicativ]
VTYP	[COMPLEX-PRED vv-perm]
83	[CLAUSE-TYPE decl, PASSIVE -, PERS 3]

Figure: F-Structure for *nAdiyah nE yAsIn kO kitAb dEkHnE dI* 'Nadya let Yassin look at the book.'

Overview

- 1 Complex Predicates
- 2 Types of Complex Predicates
- 3 A Reference Dependency Bank for CPs**
- 4 Conclusion

Design of the Reference Dependency Bank

- contains sentences illustrating examples of all common CP types in Hindi/Urdu

Design of the Reference Dependency Bank

- contains sentences illustrating examples of all common CP types in Hindi/Urdu
- strategy for creating the dependency bank:

Design of the Reference Dependency Bank

- contains sentences illustrating examples of all common CP types in Hindi/Urdu
- strategy for creating the dependency bank:
 - ▶ sentences were parsed using the Urdu ParGram grammar → c- and f-structures
 - ▶ banked/disambiguated using LFG Parsebanker [Rosén et al. 2009]
 - ▶ converted into triples format (see PARC700, [King et al. 2003]) via XLE-internal process
 - ▶ triples conversion is flexible; features may be flattened or deleted

Design of the Reference Dependency Bank

- contains sentences illustrating examples of all common CP types in Hindi/Urdu
- strategy for creating the dependency bank:
 - ▶ sentences were parsed using the Urdu ParGram grammar → c- and f-structures
 - ▶ banked/disambiguated using LFG Parsebanker [Rosén et al. 2009]
 - ▶ converted into triples format (see PARC700, [King et al. 2003]) via XLE-internal process
 - ▶ triples conversion is flexible; features may be flattened or deleted
- triples format is theory-neutral; enables parsers to evaluate against the reference bank

CPs in the Reference Dependency Bank

- to model the verbal complex of CPs:
 - ▶ all parts of CP contributing arguments are concatenated by underscore
 - ▶ makes clear that CP is main predicate of clause

CPs in the Reference Dependency Bank

- to model the verbal complex of CPs:
 - ▶ all parts of CP contributing arguments are concatenated by underscore
 - ▶ makes clear that CP is main predicate of clause
- triples representation split in two parts:
 - ▶ list arguments of the whole (complex) predication
 - ▶ indication of which part of the CP contributes which argument
 - ▶ consecutive labeling of CP parts based on their linear order

CPs in the Reference Dependency Bank

- to model the verbal complex of CPs:
 - ▶ all parts of CP contributing arguments are concatenated by underscore
 - ▶ makes clear that CP is main predicate of clause
- triples representation split in two parts:
 - ▶ list arguments of the whole (complex) predication
 - ▶ indication of which part of the CP contributes which argument
 - ▶ consecutive labeling of CP parts based on their linear order
- triples are restricted to predicate-argument relations
- neglect the more detailed information in f-structures

CPs in the Reference Dependency Bank

nAdiyah nE yAsIn kO kitAb dEkH-nE d-I

Nadya.F.Sg Erg Yassin.M.Sg Dat book.F.Sg see-Inf.M.Sg give-Perf.F.Sg

'Nadya let Yassin look at the book.'

CPs in the Reference Dependency Bank

nAdiyah nE yAsIn kO kitAb dEkH-nE d-I

Nadya.F.Sg Erg Yassin.M.Sg Dat book.F.Sg see-Inf.M.Sg give-Perf.F.Sg

'Nadya let Yassin look at the book.'

XLE f-structure

CPs in the Reference Dependency Bank

nAdiyah nE yAsIn kO kitAb dEkH-nE d-I

Nadya.F.Sg Erg Yassin.M.Sg Dat book.F.Sg see-Inf.M.Sg give-Perf.F.Sg

'Nadya let Yassin look at the book.'

XLE f-structure

↓ triples conversion

triples format

CPs in the Reference Dependency Bank

nAdiyah nE yAsIn kO kitAb dEkH-nE d-I
Nadya.F.Sg Erg Yassin.M.Sg Dat book.F.Sg see-Inf.M.Sg give-Perf.F.Sg
'Nadya let Yassin look at the book.'

XLE f-structure

triples conversion

triples format

```
pred(root, dEkH_dE)
subj(dEkH_dE, nAdiyah)
obj-go(dEkH_dE, yAsIn)
obj(dEkH_dE, kitAb)
complex-pred-type(dEkH_dE, vv-perm)
cp-part1(dEkH_dE, dEkH)
cp-part2(dEkH_dE, dE)
arg1(dE, nAdiyah)
arg2(dE, dEkH)
arg1(dEkH, yAsIn)
arg2(dEkH, kitAb)
asp(dEkH_dE, perf).
```

application of rewrite rules

Overview

- 1 Complex Predicates
- 2 Types of Complex Predicates
- 3 A Reference Dependency Bank for CPs
- 4 Conclusion**

Conclusion I

- South Asian languages make heavy use of CPs

Conclusion I

- South Asian languages make heavy use of CPs
- essential to know about proper treatment

Conclusion I

- South Asian languages make heavy use of CPs
- essential to know about proper treatment
- essential to know about different CP types

Conclusion I

- South Asian languages make heavy use of CPs
- essential to know about proper treatment
- essential to know about different CP types
- also: essential to know what is not a CP

Conclusion I

- South Asian languages make heavy use of CPs
- essential to know about proper treatment
- essential to know about different CP types
- also: essential to know what is not a CP
 - ▶ e.g. auxiliaries, modal constructions need to be distinguished from CPs

Conclusion I

- South Asian languages make heavy use of CPs
- essential to know about proper treatment
- essential to know about different CP types
- also: essential to know what is not a CP
 - ▶ e.g. auxiliaries, modal constructions need to be distinguished from CPs
 - ▶ examples of these constructions are also included in the dependency bank

Conclusion I

- South Asian languages make heavy use of CPs
- essential to know about proper treatment
- essential to know about different CP types
- also: essential to know what is not a CP
 - ▶ e.g. auxiliaries, modal constructions need to be distinguished from CPs
 - ▶ examples of these constructions are also included in the dependency bank
- other treebanks offer only limited annotation for CPs (e.g. HUTB, [Bhatt et al. 2009])

Conclusion II

- presented a reference dependency bank for CPs (and other constructions that are often confused with CPs)

Conclusion II

- presented a reference dependency bank for CPs (and other constructions that are often confused with CPs)
- reference bank is designed in a theory-independent way

Conclusion II

- presented a reference dependency bank for CPs (and other constructions that are often confused with CPs)
- reference bank is designed in a theory-independent way
- represents a typology of CPs (reflects what we currently know about CPs...)

Conclusion II

- presented a reference dependency bank for CPs (and other constructions that are often confused with CPs)
- reference bank is designed in a theory-independent way
- represents a typology of CPs (reflects what we currently know about CPs...)
- researchers may consult this resource when working on a new language
 - ▶ for theoretical syntax research
 - ▶ for constructing analyses for treebanks
 - ▶ for evaluating new parsers

Conclusion II

- presented a reference dependency bank for CPs (and other constructions that are often confused with CPs)
- reference bank is designed in a theory-independent way
- represents a typology of CPs (reflects what we currently know about CPs...)
- researchers may consult this resource when working on a new language
 - ▶ for theoretical syntax research
 - ▶ for constructing analyses for treebanks
 - ▶ for evaluating new parsers
- freely available on the internet

http://ling.uni-konstanz.de/pages/home/pargram_urdu/main/Resources.html

References



Bhatt, R., B. Narasimhan, M. Palmer, O. Rambow, D. M. Sharma, and F. Xia. 2009.
A Multi-Representational and Multi-Layered Treebank for Hindi/Urdu.
In *Proceedings of the Third Linguistic Annotation Workshop, ACL-IJCNLP '09*, 186–189, Stroudsburg, PA, USA. Association for Computational Linguistics.



Bögel, T., M. Butt, A. Hautli, and S. Sulger. 2009.
Urdu and the Modular Architecture of ParGram.
In *Proceedings of the Conference on Language and Technology 2009 (CLT09)*.



Butt, M. 1995.
The Structure of Complex Predicates in Urdu.
CSLI Publications.



Butt, M. 2010.
The Light Verb Jungle: Still Hacking Away.
In M. Amberber, B. Baker, and M. Harvey (Eds.), *Complex Predicates in Cross-Linguistic Perspective*. Cambridge University Press.



Butt, M., and T. H. King.
2007.
Urdu in a Parallel Grammar Development Environment.
Language Resources and Evaluation 41(2):191–207.



King, T. H., R. Crouch, S. Riezler, M. Dalrymple, and R. Kaplan. 2003.
The PARC700 Dependency Bank.
In *Proceedings of the EAACL03: 4th International Workshop on Linguistically Interpreted Corpora (LINC-03)*.



Rosén, V., P. Meurer, and K. de Smedt. 2009.
LFG Parsebanker: A Toolkit for Building and Searching a Treebank as a Parsed Corpus.
In F. V. Eynde, A. Frank, G. van Noord, and K. D. Smedt (Eds.), *Proceedings of the 7th International Workshop on Treebanks and Linguistic Theories (TLT7)*, 127–133. LOT.