

UrduGram: Towards a Deep, Large-Coverage Grammar for Urdu and Hindi

Tafseer Ahmed, Tina Bögel, Miriam Butt, Annette Hautli, Ghulam Raza, Sebastian Sulger and Veronika Walther

Universität Konstanz

FB Kolloquium, May 2010

- 1 Urdu & the UrduGram Project
- 2 Urdu Transliterator
- 3 Syntax
- 4 Semantics

Urdu

Urdu is

Urdu

Urdu is

- a South Asian language spoken primarily in Pakistan and India

Urdu

Urdu is

- a South Asian language spoken primarily in Pakistan and India
- descended from (a version of) Sanskrit (sister language of Latin)

Urdu

Urdu is

- a South Asian language spoken primarily in Pakistan and India
- descended from (a version of) Sanskrit (sister language of Latin)
- structurally identical to Hindi (spoken mainly in India)

Urdu

Urdu is

- a South Asian language spoken primarily in Pakistan and India
- descended from (a version of) Sanskrit (sister language of Latin)
- structurally identical to Hindi (spoken mainly in India)
- together with Hindi the fourth most spoken language in the world (~ 250 million native speakers)

Urdu and Hindi

The two languages are regarded as *structurally identical*:

Urdu and Hindi

The two languages are regarded as *structurally identical*:

- syntax/morphology are practically identical

Urdu and Hindi

The two languages are regarded as *structurally identical*:

- syntax/morphology are practically identical
- vocabulary is practically identical (Urdu: borrowed from Persian/Arabic; Hindi: borrowed from Sanskrit)

Urdu and Hindi

The two languages are regarded as *structurally identical*:

- syntax/morphology are practically identical
- vocabulary is practically identical (Urdu: borrowed from Persian/Arabic; Hindi: borrowed from Sanskrit)
- main difference is in the script

Urdu and Hindi

The two languages are regarded as *structurally identical*:

- syntax/morphology are practically identical
 - vocabulary is practically identical (Urdu: borrowed from Persian/Arabic; Hindi: borrowed from Sanskrit)
 - main difference is in the script
- We are developing a single grammar and lexicon for both of the languages!

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

Context of Work

- Computational LFG grammar in development in Konstanz
- Aim: large-scale LFG grammar for parsing Urdu/Hindi
- 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
- 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
- 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
- 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

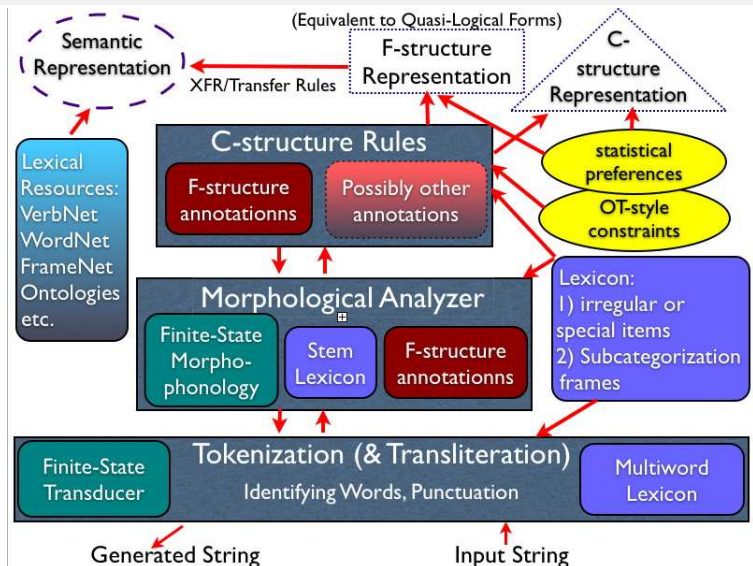
Context of Work

- Computational LFG grammar in development in Konstanz
- Aim: large-scale LFG grammar for parsing Urdu/Hindi
- 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
- 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:
 - English, German, French, Japanese, Norwegian, Welsh, Georgian, Hungarian, Turkish, Chinese, Indonesian, **Urdu** (among many others)

The ParGram Grammar Architecture



The 'Parallel' in ParGram

Analysis for transitive sentence in English ParGram grammar (F-Structure, "Functional Structure"):

The 'Parallel' in ParGram

Analysis for transitive sentence in English ParGram grammar (F-Structure, "Functional Structure"):

"Nadya saw the book."

```

[PRED      'see<[1:Nadya], [113:book]>'
  SUBJ     [
    PRED    'Nadya'
    CHECK   [_LEX-SOURCE morphology, _PROPER known-name]
    NTYPE   [
      NSEM  [PROPER [NAME-TYPE first_name, PROPER-TYPE name]]
      NSYN  proper
    ]
    1[CASE nom, GEND-SEM female, HUMAN +, NUM sg, PERS 3
  ]
  OBJ      [
    PRED    'book'
    CHECK   [_LEX-SOURCE countnoun-lex]
    NTYPE   [
      NSEM  [COMMON count]
      NSYN  common
    ]
    SPEC    [
      DET   [
        PRED    'the'
        DET-TYPE def
      ]
    ]
    113[CASE obl, NUM sg, PERS 3
  ]
  CHECK    [_SUBCAT-FRAME V-SUBJ-OBJ]
  TNS-ASP  [MOOD indicative, PERF --, PROG --, TENSE past]
  57[CLAUSE-TYPE decl, PASSIVE -, VTYPE main

```

The 'Parallel' in ParGram (cont.)

Analysis for the same transitive sentence in Urdu ParGram grammar (F-Structure, "Functional Structure"):

The 'Parallel' in ParGram (cont.)

Analysis for the same transitive sentence in Urdu ParGram grammar (F-Structure, "Functional Structure"):

"nAdiyah nE kitAb dEKHI"

```

[PRED      'dEKH<[1:nAdiyah], [20:kitAb]>'
  [PRED      'nAdiyah'
    CHECK    [_NMORPH obl]
  ]
SUBJ      NTYPE  [NSEM [PROPER [PROPER-TYPE name]]]
           [NSYN proper]
           SEM-PROP [SPECIFIC +]
           1[CASE erg, GEND fem, NUM sg, PERS 3]
  [PRED      'kitAb'
    NTYPE  [NSEM [COMMON count]]
           [NSYN common]
    20[CASE nom, GEND fem, NUM sg, PERS 3]
  ]
CHECK    [_VMORPH [_MTYPE inf]]
         [_RESTRICTED -, _SUBCAT-FRAME V-SUBJ-OBJ, _VFORM perf]
LEX-SEM  [AGENTIVE +]
TNS-ASP  [ASPECT perf, MOOD indicative]
42[CLAUSE-TYPE decl, PASSIVE -, VTYPE main]

```

The 'Parallel' in ParGram (cont.)

Analysis for the same transitive sentence in Urdu ParGram grammar (F-Structure, "Functional Structure"):

"nAdiyah ne kitAb deKHI"

```

[PRED 'deKH<[1:nAdiyah], [20:kitAb]>'
  [PRED 'nAdiyah'
    CHECK [_NMORPH obi]
  ]
SUBJ [NTYPE [NSEM [PROPER [PROPER-TYPE name]]]
      [NSYN proper]
      SEM-PROP [SPECIFIC +]
      1[CASE erg, GEND fem, NUM sg, PERS 3]
  ]
OBJ [PRED 'kitAb'
     [NTYPE [NSEM [COMMON count]]]
     [NSYN common]
     20[CASE nom, GEND fem, NUM sg, PERS 3]
  ]
CHECK [_VMORPH [_MTYPE inf]]
      [_RESTRICTED -, _SUBCAT-FRAME V-SUBJ-OBJ, _VFORM perf]
LEX-SEM [AGENTIVE +]
TNS-ASP [ASPECT perf, MOOD indicative]
42[CLAUSE-TYPE decl, PASSIVE -, VTYPE main]

```

→ *Analyses are kept parallel where possible*

The 'Parallel' in ParGram (cont.)

Analysis for the same transitive sentence in Urdu ParGram grammar (F-Structure, "Functional Structure"):

"nAdiyah ne kitAb deKHI"

```

[PRED 'dEkH<[1:nAdiyah], [20:kitAb]>'
  [PRED 'nAdiyah'
    CHECK [_NMORPH obl]
  ]
SUBJ [NTYPE [NSEM [PROPER [PROPER-TYPE name]]]
      [NSYN proper]
      SEM-PROP [SPECIFIC +]
      1[CASE erg, GEND fem, NUM sg, PERS 3]
  ]
OBJ [PRED 'kitAb'
     [NTYPE [NSEM [COMMON count]]]
     [NSYN common]
     20[CASE nom, GEND fem, NUM sg, PERS 3]
  ]
CHECK [_VMORPH [_MTYPE inf]]
      [_RESTRICTED -, _SUBCAT-FRAME V-SUBJ-OBJ, _VFORM perf]
LEX-SEM [AGENTIVE +]
TNS-ASP [ASPECT perf, MOOD indicative]
42[CLAUSE-TYPE decl, PASSIVE -, VTYPE main]

```

- *Analyses are kept parallel where possible*
- *Features are kept parallel where possible*

The 'Parallel' in ParGram (cont.)

Demo: Large-Scale English ParGram Grammar

Computational Grammars - What For?

The Motivation behind ParGram

Computational Grammars - What For?

The Motivation behind ParGram

The ParGram project is working on *Deep Grammars*

Computational Grammars - What For?

The Motivation behind ParGram

The ParGram project is working on *Deep Grammars*

- Provide detailed syntactic and semantic analyses

Computational Grammars - What For?

The Motivation behind ParGram

The ParGram project is working on *Deep Grammars*

- Provide detailed syntactic and semantic analyses
- Encode grammatical functions, tense, number etc.

Computational Grammars - What For?

The Motivation behind ParGram

The ParGram project is working on *Deep Grammars*

- Provide detailed syntactic and semantic analyses
- Encode grammatical functions, tense, number etc.
- Linguistically motivated

Computational Grammars - What For?

The Motivation behind ParGram

The ParGram project is working on *Deep Grammars*

- Provide detailed syntactic and semantic analyses
- Encode grammatical functions, tense, number etc.
- Linguistically motivated
- Usually manually constructed (→ linguistic intuition)

Computational Grammars - What For?

Possible Applications

Computational Grammars - What For?

Possible Applications

Large-Coverage, Deep Computational Grammars can be useful for:

Computational Grammars - What For?

Possible Applications

Large-Coverage, Deep Computational Grammars can be useful for:

- Meaning-Sensitive Applications

Computational Grammars - What For?

Possible Applications

Large-Coverage, Deep Computational Grammars can be useful for:

- Meaning-Sensitive Applications
 - Web-Search

Computational Grammars - What For?

Possible Applications

Large-Coverage, Deep Computational Grammars can be useful for:

- Meaning-Sensitive Applications
 - Web-Search
 - Question-Answering

Computational Grammars - What For?

Possible Applications

Large-Coverage, Deep Computational Grammars can be useful for:

- Meaning-Sensitive Applications
 - Web-Search
 - Question-Answering
 - Knowledge Representation

Computational Grammars - What For?

Possible Applications

Large-Coverage, Deep Computational Grammars can be useful for:

- Meaning-Sensitive Applications
 - Web-Search
 - Question-Answering
 - Knowledge Representation
- Text Summarization

Computational Grammars - What For?

Possible Applications

Large-Coverage, Deep Computational Grammars can be useful for:

- Meaning-Sensitive Applications
 - Web-Search
 - Question-Answering
 - Knowledge Representation
- Text Summarization
- Machine Translation

Computational Grammars - What For?

Possible Applications

Large-Coverage, Deep Computational Grammars can be useful for:

- Meaning-Sensitive Applications
 - Web-Search
 - Question-Answering
 - Knowledge Representation
- Text Summarization
- Machine Translation
- Computer-Assisted Language Learning

Computational Grammars - What For?


powerset.com

Powerset

Wikipedia Articles

companies that were bought by Microsoft

Microsoft: Companies acquired



[Visio Corporation](#)
[LinkExchange](#)
[Parlano](#)
[Jellyfish](#)

Wikipedia Articles

- [List of mergers and acquisitions by Microsoft](#) Microsoft is an American computer technology corporation based in Redmond, Washington. ... Of the **Microsoft has acquired**, 99 were based in the United States.
- [Criticism of Microsoft](#) Criticism of Microsoft has followed various aspects of its business practices. ... Microsoft has **acquired** several **companies** and products, including some that competed with earlier Microsoft products.
- [History of Microsoft](#) Activity grew quickly as developers from around the world participated, and by early 2007 commercial open source **companies**, such as **Ubuntu**, offered enterprise open source software exclusively on the Microsoft platform. ... **Microsoft** wanted to **purchase Yahoo** (first completely, later partially) in order to strengthen its search engine market vis-à-vis Google.
- [List of assets owned by Microsoft Corporation](#) Microsoft has interests in various areas: ... List of **companies acquired by Microsoft Corporation**

Computational Grammars - What For?

powerset.com

- “Semantic search engine”

The screenshot shows the PowerSet website interface. At the top, the logo 'Powerset' is visible. Below it, a search bar contains the query 'companies that were bought by Microsoft'. The results are displayed under a heading 'Microsoft: Companies acquired'. There are four items shown: 'Visio Corporation' with a logo, 'LinkExchange' with a 'NO IMAGE' placeholder, 'Parlano' with a logo, and 'Jellyfish' with a logo. Below this, there is a section titled 'Wikipedia Articles' containing a list of search results with expandable dropdown arrows. The first result is 'List of mergers and acquisitions by Microsoft', the second is 'Criticism of Microsoft', the third is 'History of Microsoft', and the fourth is 'List of assets owned by Microsoft Corporation'. Each result snippet includes the title and a short text excerpt.

Computational Grammars - What For?

powerset.com

- “Semantic search engine”
- Uses large-scale English LFG

The screenshot shows the Powerset search engine interface. At the top, the Powerset logo is visible. Below it, a search bar contains the query "companies that were bought by Microsoft". The search results are displayed under the heading "Microsoft: Companies acquired". There are four results shown, each with a logo and a name: Visio Corporation (with a logo), LinkExchange (with a "NO IMAGE" placeholder), Parlano (with a logo), and Jellyfish (with a logo). Below the search results, there is a section titled "Wikipedia Articles" which lists several articles related to Microsoft, including "List of mergers and acquisitions by Microsoft", "Criticism of Microsoft", "History of Microsoft", and "List of assets owned by Microsoft Corporation". The text in the Wikipedia articles is partially visible, showing phrases like "Microsoft is an American computer technology corporation based in Redmond, Washington. ... Of the Microsoft has acquired, 99 were based in the United States." and "Microsoft has acquired several companies and products including some that competed with earlier Microsoft products."

Computational Grammars - What For?

powerset.com

- “Semantic search engine”
- Uses large-scale English LFG
- Works on English Wikipedia

The screenshot shows the Powerset search engine interface. At the top, the search query is "companies that were bought by Microsoft". Below the query, a section titled "Microsoft: Companies acquired" displays four results with logos and names: Visio Corporation, LinkExchange, Parlano, and Jellyfish. Below this, a section titled "Wikipedia Articles" lists several articles related to Microsoft, including "List of mergers and acquisitions by Microsoft", "Criticism of Microsoft", "History of Microsoft", and "List of assets owned by Microsoft Corporation". The interface includes a search bar, navigation buttons, and a footer with the page number "12 / 60".

Computational Grammars - What For?

powerset.com

- “Semantic search engine”
- Uses large-scale English LFG
- Works on English Wikipedia
- Parses query and matches with parsed corpus

The screenshot shows the Powerset search engine interface. At the top, the search query is "companies that were bought by Microsoft". Below the query, there is a section titled "Microsoft: Companies acquired" which displays four results: Visio Corporation (with a logo), LinkExchange (with a "NO IMAGE" placeholder), Parlano (with a logo), and Jellyfish (with a logo). Below this, there is a section titled "Wikipedia Articles" which lists several articles related to Microsoft, including "List of mergers and acquisitions by Microsoft", "Criticism of Microsoft", "History of Microsoft", and "List of assets owned by Microsoft Corporation". The text in the Wikipedia articles is partially visible and contains several instances of the word "companies" and "acquired" highlighted in yellow.

Computational Grammars - What For?

powerset.com

- “Semantic search engine”
 - Uses large-scale English LFG
 - Works on English Wikipedia
 - Parses query and matches with parsed corpus
- *Can give better results than regular search engines*

The screenshot shows the Powerset search engine interface. At the top, the search bar contains the query "companies that were bought by Microsoft". Below the search bar, a section titled "Microsoft: Companies acquired" displays four results: Visio Corporation (with a logo), LinkExchange (with a "NO IMAGE" placeholder), Parlano (with a logo), and Jellyfish (with a logo). Below this, a "Wikipedia Articles" section lists several articles with snippets of text. The first article is "List of mergers and acquisitions by Microsoft", the second is "Criticism of Microsoft", the third is "History of Microsoft", and the fourth is "List of assets owned by Microsoft Corporation". Each snippet contains the word "companies" highlighted in yellow, indicating a match with the search query.

Computational Grammars - What For?

powerset.com

- “Semantic search engine”
- Uses large-scale English LFG
- Works on English Wikipedia
- Parses query and matches with parsed corpus

→ *Can give better results than regular search engines*

(Example: ‘X was bought by Y’
vs. ‘Y acquired X’)

The screenshot shows the Powerset search engine interface. At the top, there's a search bar with the query "companies that were bought by Microsoft". Below the search bar, a section titled "Microsoft: Companies acquired" displays four results with logos and names: Visio Corporation, LinkExchange, Parlano, and Jellyfish. Below this, a "Wikipedia Articles" section lists several articles with snippets of text, including "List of mergers and acquisitions by Microsoft", "Criticism of Microsoft", "History of Microsoft", and "List of assets owned by Microsoft Corporation". The snippets contain highlighted words like "acquired", "companies", and "Microsoft".

Our Overall Architecture

Our parsing architecture currently looks like this:

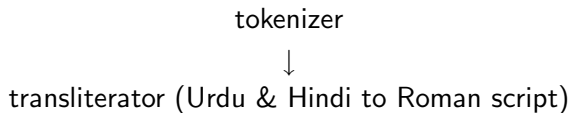
Our Overall Architecture

Our parsing architecture currently looks like this:

tokenizer

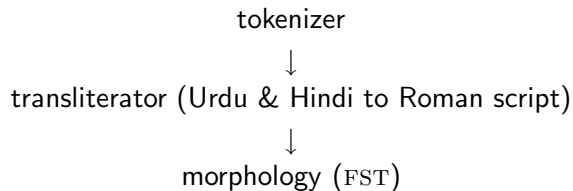
Our Overall Architecture

Our parsing architecture currently looks like this:



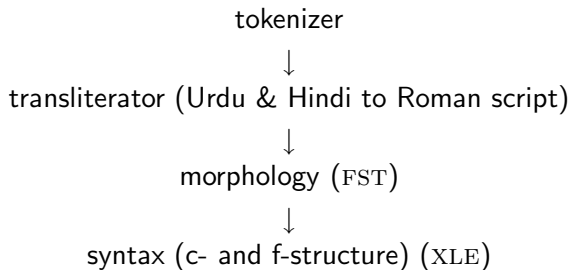
Our Overall Architecture

Our parsing architecture currently looks like this:



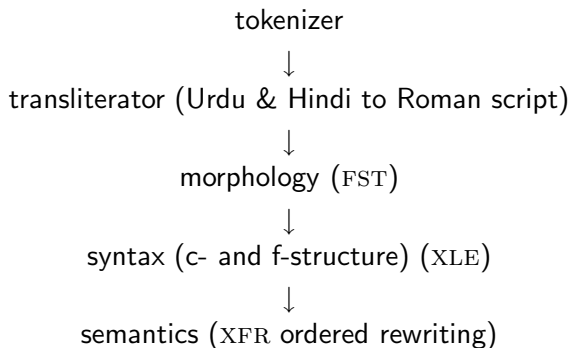
Our Overall Architecture

Our parsing architecture currently looks like this:



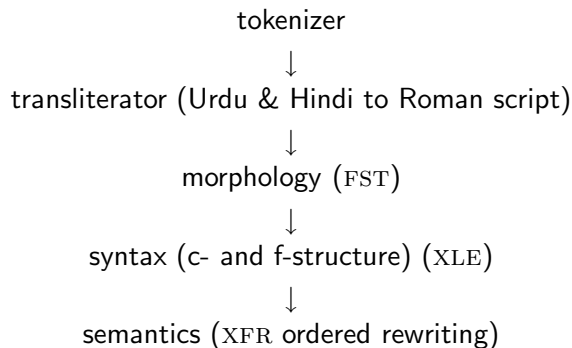
Our Overall Architecture

Our parsing architecture currently looks like this:



Our Overall Architecture

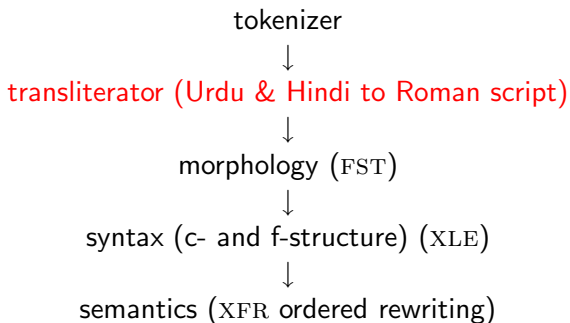
Our parsing architecture currently looks like this:



XLE is the overall development platform, with the other modules (FST and XFR) being plugged into it.

Overview

Overall Architecture



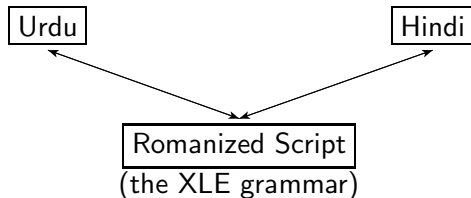
Aim of the transliterator

Our aim is to build and integrate a transliterator that allows for both, Urdu and Hindi, to be parsed AND generated with the same grammar.

ہاں بھلا کر تیرا بھلا ہوگا
اور درویش کی صدا کیا ہے

हां भला कर तिरा भला होगा
और दरवेश की सदा क्या है

couplet by the poet Mirza Ghalib



→ Right now we are working on the Urdu-Roman transliterator.

Transliteration scheme

An excerpt from our scheme table:

Unicode Urdu character	Latin letter in transliteration scheme	Phoneme
ب	b	/b/
پ	p	/p/
ت	t	/t/
ٹ	T	/t/
ج	j	/j/
چ	c	/tʃ/

Basic idea of the transliterator

- use finite state transducer to allow for generation and parsing.

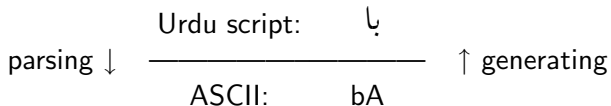
Basic idea of the transliterator

- use finite state transducer to allow for generation and parsing.

parsing ↓ $\frac{\text{Urdu script: } \text{بَ}}{\text{ASCII: } \text{bA}}$ ↑ generating

Basic idea of the transliterator

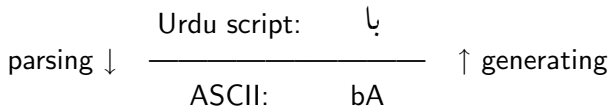
- use finite state transducer to allow for generation and parsing.



- The same concept will be used to create a transliterator for Hindi/Devanagari

Basic idea of the transliterator

- use finite state transducer to allow for generation and parsing.



- The same concept will be used to create a transliterator for Hindi/Devanagari
- This way we can parse Urdu script and generate Hindi script (and vice versa)

Position of the transliterator

- the transliterator is composed with the tokenizer (separates the words within a sentence)

Position of the transliterator

- the transliterator is composed with the tokenizer (separates the words within a sentence)
- tokenizer and transliterator are placed in front of the morphology

Position of the transliterator

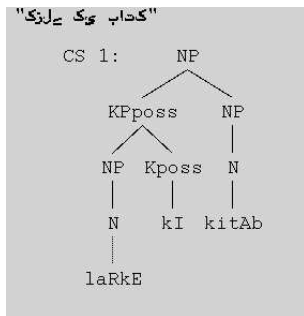
- the transliterator is composed with the tokenizer (separates the words within a sentence)
- tokenizer and transliterator are placed in front of the morphology

<i>Transliterator</i>	Input ↓ Output	کتاب ↓ kitAb
<i>Morphology</i>	Input ↓ Output	kitAb ↓ kitAb+Noun+Fem+Sg+Count
<i>XLE</i>

Example

→ The transliterator at this position works quite well:

(1) laRkE kI kitAb
 boy GEN book
 'The boy's book'



→ Problem: long sentences or highly ambiguous words (when looking at script) need some time to parse.

Problems of the script - an example

The problem of the vowels ...

- Diacritics represent short vowels

<i>Urdu script</i>	<i>Roman script</i>
بَ	ba
بِ	bi
بُ	bu

Problems of the script - an example

The problem of the vowels ...

- Diacritics represent short vowels

<i>Urdu script</i>	<i>Roman script</i>
بَ	ba
بِ	bi
بُ	bu

- (2) nAdyA nE yasIn kO kitAb dEkHnE dI
 Nadya ERG Yasin DAT see let
 'Nadya let Yassin see the book'

نادیا نی یسین کو کتاب دیکھنی دی

Problems of the script - an example

The problem of the vowels ...

- Diacritics represent short vowels

<i>Urdu script</i>	<i>Roman script</i>
بَ	ba
بِ	bi
بُ	bu

- (2) nAdyA nE yasIn kO kitAb dEkHnE dI
 Nadya ERG Yasin DAT see let
 'Nadya let Yassin see the book'

نادیا نی یسین کو کتاب دیکھنی دی

- **Unfortunately**, these diacritics tend to be left out.

نادیا نی یسین کو کتاب دیکھنی دی

Consequences

- If the input is without diacritics, e.g. کتاب ...

Urdu script	letter combination	representation	translation
کتاب	ktAb	kitAb	'book'

Consequences

- If the input is without diacritics, e.g. کتاب ...

Urdu script	letter combination	representation	translation
کتاب	ktAb	kitAb	'book'

- .. then there are all kinds of possible combinations:
kitAb, kutaAb, kitAbu, ikatAubi, ukitAbia, akatAbu, aukatAib

Consequences

- If the input is without diacritics, e.g. کتاب ...

Urdu script	letter combination	representation	translation
کتاب	ktAb	kitAb	'book'

- .. then there are all kinds of possible combinations:
kitAb, kutaAb, kitAbu, ikatAubi, ukitAbia, akatAbu, aukatAib

(demo)

Solution

In order to restrict this overgeneration the possible letter combinations need to be constrained:

Solution

In order to restrict this overgeneration the possible letter combinations need to be constrained:

- which vowels are actually allowed to cooccur?
→ *ai*, but not *ia*?

Solution

In order to restrict this overgeneration the possible letter combinations need to be constrained:

- which vowels are actually allowed to cooccur?
→ *ai*, but not *ia*?
- which consonants are actually allowed to cooccur?
→ initial *kr*, but not *gr*?

Solution

In order to restrict this overgeneration the possible letter combinations need to be constrained:

- which vowels are actually allowed to cooccur?
→ *ai*, but not *ia*?
- which consonants are actually allowed to cooccur?
→ initial *kr*, but not *gr*?
- certain combinations with semi-vowels or consonants are not allowed:
→ a short vowel followed by *v* may not be followed by *u* or *i*

Solution

In order to restrict this overgeneration the possible letter combinations need to be constrained:

- which vowels are actually allowed to cooccur?
 - *ai*, but not *ia*?
- which consonants are actually allowed to cooccur?
 - initial *kr*, but not *gr*?
- certain combinations with semi-vowels or consonants are not allowed:
 - a short vowel followed by *v* may not be followed by *u* or *i*
- certain positions are prohibited:
 - a word can never end in a short vowel or begin with a short vowel that is only represented with a diacritic

Solution

- write rules and filters out of these constraints and apply them to the transliterator

(demo)

Solution

- write rules and filters out of these constraints and apply them to the transliterator

(demo)

- Problem: these “rules” cannot be found in the literature - they are a product of extensive manual labor

Solution

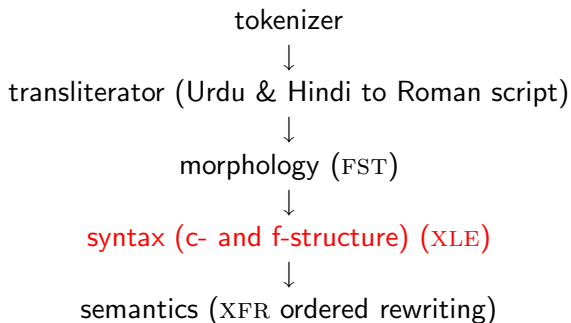
- write rules and filters out of these constraints and apply them to the transliterator

(demo)

- Problem: these “rules” cannot be found in the literature - they are a product of extensive manual labor
 - However, the transliterator works quite well now
- Some sentences are still a little slow (but I keep looking for possible restrictions)
- continue with generation of Urdu and the Hindi transliterator

Overview

Overall Architecture



Syntax

- syntax component is at the core of Urdu grammar

Syntax

- syntax component is at the core of Urdu grammar
- theoretical background: LFG

Syntax

- syntax component is at the core of Urdu grammar
- theoretical background: LFG
- well-studied (~ 30 years) framework with computational usability

Syntax

- syntax component is at the core of Urdu grammar
- theoretical background: LFG
- well-studied (~ 30 years) framework with computational usability
- c- and f-structures used for syntactic representation

Syntax

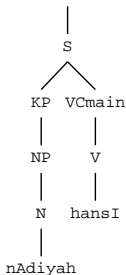
- syntax component is at the core of Urdu grammar
- theoretical background: LFG
- well-studied (~ 30 years) framework with computational usability
- c- and f-structures used for syntactic representation
 - c-structure: basic constituent structure (“tree”) and linear precedence (~ what parts belong together)

Syntax

- syntax component is at the core of Urdu grammar
- theoretical background: LFG
- well-studied (~ 30 years) framework with computational usability
- c- and f-structures used for syntactic representation
 - c-structure: basic constituent structure (“tree”) and linear precedence (~ what parts belong together)
 - f-structure: encodes syntactic functions and properties

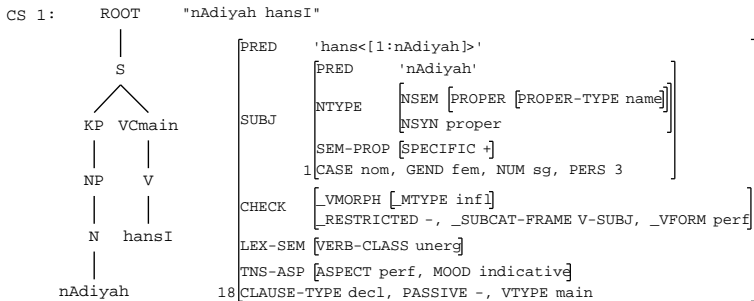
Syntax

CS 1: ROOT "nAdiyah hansI"



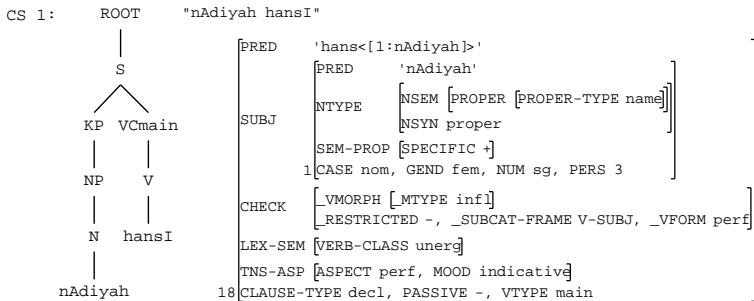
[PRED 'hans<[1:nAdiyah]>']
SUBJ	[PRED 'nAdiyah' NTYPE [NSEM [PROPER [PROPER-TYPE name]] NSYN proper]]]
	SEM-PROP [SPECIFIC +] 1[CASE nom, GEND fem, NUM sg, PERS 3]	
CHECK	[VMORPH [_MTYPE infl] _RESTRICTED -, _SUBCAT-FRAME V-SUBJ, _VFORM perf]]]
	LEX-SEM [VERB-CLASS unerg]	
	TNS-ASP [ASPECT perf, MOOD indicative]	
18	[CLAUSE-TYPE decl, PASSIVE -, VTYPE main]]

Syntax



- current size: 53 phrase-structure rules, annotated for syntactic function (usual size of large-scale grammars: 350–400 rules)

Syntax



- current size: 53 phrase-structure rules, annotated for syntactic function (usual size of large-scale grammars: 350–400 rules)
- coverage: basic clauses with free word order, NP syntax, tense and aspect, causative verbs, complex predicates, relative clauses, passives, semantically-based case marking

Discontinuous NPs in Urdu

- 1 Well known discontinuities
- 2 NP-internal discontinuity in Urdu
- 3 LFG implementation
- 4 Conclusion

Extraction from DP

(2) a.

Er hat **viele Bücher über Logik** gekauft.

He has many books on logic bought

'He has bought many books about logic.'

b. **Bücher über Logik** hat er **viele** gekauft.c. **Über Logik** hat er **viele Bücher** gekauft. (German)(3) **mantiq=par nidA=nE Ek kitAb**

logic=Loc.on Nida=Erg one book.F.3Sg

xarld-I he.

buy-Perf be.Pres

'Nida has purchased a book on logic.' (Urdu)

Quantifier Float

- (4) a. **They all** have bought a car.
 b. **They** have **all** bought a car.
- (5) **Am** all=nE **bahut** kHA-E
 mango.PI Ali=Erg many eat-Perf
 'Ali ate many mangoes.' (Urdu)

Constituent-level discontinuities in Urdu

NP-internal discontinuity

- Discontinuous NP
- Discontinuous AP

When NP-internal discontinuity occurs in Urdu

The NP-internal discontinuity in Urdu can occur when the argument-taking noun is modified by:

- ARGUMENT-TAKING ADJECTIVES
- ARGUMENT-TAKING SPECIFIER NOUNS

Argument-taking adjectives in Urdu

Nr.	Type of Argument	Example of Adjective Phrase
(i)	Dative Marked	sadr=kO hAsil president=Dat possessed 'possessed by the president'
(ii)	Ablative Marked	adliyah=sE xAif courts=Abl afraid 'afraid of courts'
(iii)	Locative Marked	buxAr=mEN muftalA fever=Loc.in suffered 'suffered with fever'
(iv)	Adpositional	sihat=kE liyE muzir health=Gen for harmful 'harmful for health'

Simple examples of argument-taking nouns

(6) a. *istisnA*
 'immunity'

b.
muqaddamAt=sE istisnA
 court-case.Pl=Abl immunity
 'immunity from court-cases'

c.
muqaddamAt=sE Alnl istisnA
 court-case.Pl=Abl constitutional immunity
 'constitutional immunity from court-cases'

Simple examples of argument-taking nouns

- (7) a. barlfiNg
 'briefing'
- b.
 salAmtl=par barlfiNg
 security=Loc briefing
 'briefing on security'
- c.
 salAmtl=par tafslll barlfiNg
 security=Loc detailed briefing
 'detailed briefing on security'

Simple examples of argument-taking nouns

(8) a. mutAlbA
 'demand'

b.

Arml-clf=sE mutAlbA
 army-chief=Abl demand
 'demand to the army-chief'

c.

Arml-clf=sE qAnUnl mutAlbA
 army-chief=Abl legal demand
 'legal demand to the army-chief'

Examples of discontinuous NPs

(9) a1. $sadr=kO_1$ $hAsil_1$ $muqaddamAt=sE_2$
 president=Dat possessed court-cases=Abl

$AlnI$ $istisnA_2$
 constitutional immunity

'Constitutional Immunity from court-cases possessed
 by the president'

a2. $[NP[_{AP}[_{KP} sadr=kO]] hAsil][[_{KP} muqaddamAt=sE] AlnI istisnA]$

b. $muqaddamAt=sE_2$ $sadr=kO_1$ $hAsil_1$ $AlnI$ $istisnA_2$

c. $sadr=kO_1$ $muqaddamAt=sE_2$ $hAsil_1$ $AlnI$ $istisnA_2$

d. $*hAsil_1$ $muqaddamAt=sE_2$ $sadr=kO_1$ $AlnI$ $istisnA_2$

Hierarchical structure of AP in NP

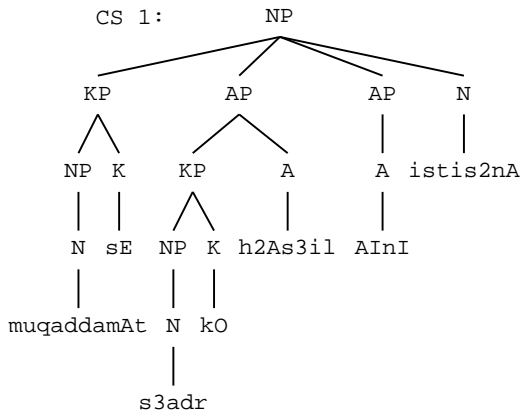


Figure: Hierarchical structure of AP in NP

Examples of discontinuous NPs

(10) a1.

Arml-clf=sE₂ salAmtl=par₁ barlfiNg₁=kA mutAlbA₂
 army-chief=Abl security=Loc.on briefing=Gen demand
 'The demand to the army chief for briefing on security'

a2. [NP[_{KP} Arml-clf=sE][_{KP}[NP[_{KP} salAmtl=par] barlfiNg]=kA]
 mutAlbA]

b. salAmtl=par₁ Arml-clf=sE₂ barlfiNg₁=kA mutAlbA₂

Examples of discontinuous NPs

- (11) [_{NP}[_{KP} Arml-clf=sE] [_{KP}[_{NP}[_{KP} mulkl salAmtl=par]
 army-chief=Abl of-country security=Loc.on
 tafslII barlfiNg]=kA] qAnUnI mutAlbA]
 detailed briefing=Gen legal demand
 'The legal demand to the army chief for a detailed
 briefing on security of the country'

LFG implementation of NP-internal discontinuity

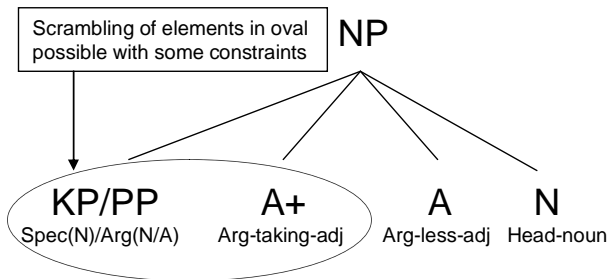


Figure: Word Order in Noun Phrases of Urdu

Implementation Issues

- Free word order in an NP
- Relating arguments with corresponding heads
- Head last constraint

LFG instruments used

- Shuffle operator (';'):
To accommodate free word order of different elements in the noun phrases.
- Non-deterministic operator ('\$'):
Relating the corresponding arguments to the corresponding heads.
- Head Precedence Operator ('<h'):
To make it sure that the head must not precede its arguments in the noun phrases.

An excerpt from Grammar Rules

NP → KP*: { (^ ADJUNCT \$ OBL)=!
 | (^ ADJUNCT \$ OBJ- GO)=!
 | (^ OBL) =!
 | (^ OBJ-GO) = ! }
 , “for scrambling”
 AP*: ! \$ (^ ADJUNCT)
 N : ^ = !

KP*: { (^ ADJUNCT \$ OBL)=!
 (^ ADJUNCT) <h (^ ADJUNCT \$ OBL)
 | }

Figure: Grammar Rules

C-structure for a discontinuous NP

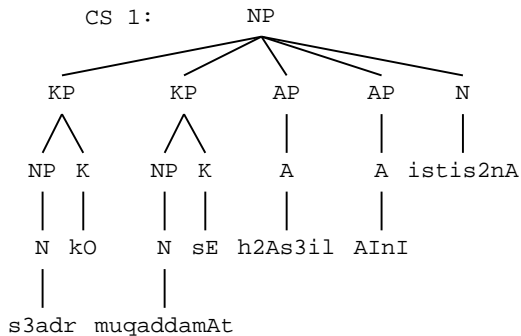


Figure: C-structure

F-structure for a discontinuous NP

"s3adr kO muqaddamAt sE h2As3il AInI istis2nA"

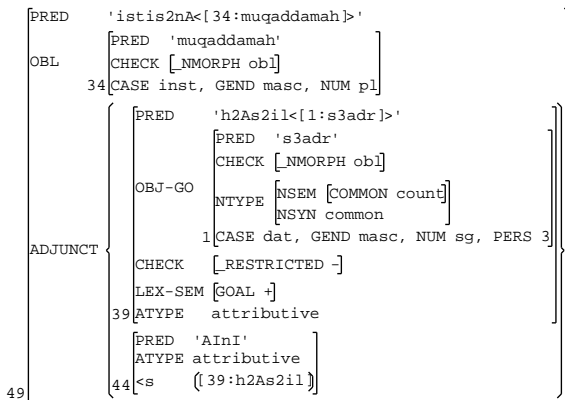


Figure: F-structure

Summary

Urdu is a typical language in which discontinuous NPs are found both at:

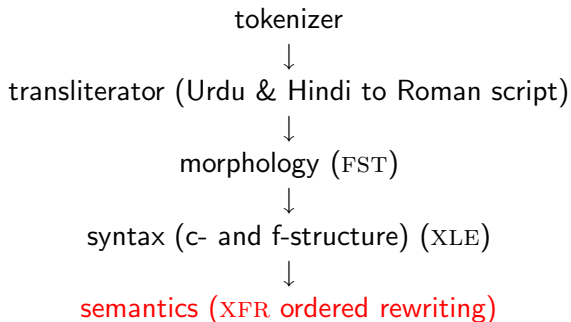
- Clause-level
- Constituent-level

Constituent-level discontinuity in Urdu can be implemented in LFG framework by making use of:

- Shuffle operator (';')
- Non-deterministic operator ('\$')
- Head-precedence operator ('<h')

Overview

Overall Architecture



Intro

Aim: a large-coverage computational semantic analyzer on the basis of a deep syntactic analysis

- use f-structures as starting point
 - apply XFR semantic rules → from f-structure facts to a semantic representation (Crouch and King, 2006)
 - judgment on the semantic well-formedness of a sentence
 - The girl laughs. → semantically well-formed
 - #The tree laughs. → semantically ill-formed
 - we need lexical information about the words in a sentence
- 1 lexical resource for Urdu verbs
 - more information on the verb and its arguments
 - 2 general lexical resource for Urdu nouns, adjectives etc.

Intro

F-structure for *nAdiyah hansI* (Nadya laughed).

"nAdiyah hansI"

[PRED	'hans<[1:nAdiyah]>']
	PRED	'nAdiyah']
SUBJ	NTYPE	[NSEM [PROPER [PROPER-TYPE name]] NSYN proper]
	SEM-PROP	[SPECIFIC +]]
	1	[CASE nom, GEND fem, NUM sg, PERS 3]
CHECK	[_VMORPH [_MTYPE inf] _RESTRICTED -, _VFORM perf]]
LEX-SEM	[VERB-CLASS unerg]]
TNS-ASP	[ASPECT perf, MOOD indicative]]
18	[CLAUSE-TYPE decl, PASSIVE -, VTYPE main]

XFR semantic rule:

PRED(%1, hans), SUBJ(%1, %subj), -OBJ(%1, %obj)

==>

word(%1, hans, verb), role(Agent, %1, %subj).

Developing an Urdu VerbNet (1)

- following the methodology of the English VerbNet (Kipper-Schuler 2006)
 - categorization of English verbs in 250 classes
 - information on event structure and argument structure of verbs
 - provides the general architecture for a VerbNet in any language
 - e.g. parts of the entry for 'laugh' in the English VerbNet

```

MEMBER: laugh
THEMROLE: Agent [+animate]
THEMROLE: Theme [+communication]
THEMROLE: Cause
THEMROLE: Recipient [+animate]
FRAME: [NP V // Basic Intransitive]
  EXAMPLE: Paul laughed.
  SYNTAX: %Agent V
FRAME: [NP V NP // TransitiveCognate Object]
  EXAMPLE: Paul laughed a cheerful laugh.
  SYNTAX: %Agent V %Theme
FRAME: [NP V PP.recipient // PPRecipient-PP]
  EXAMPLE: Paul laughed at Mary.
  SYNTAX: %Agent V {{+dest_dir}} %Recipient
  
```

Developing an Urdu VerbNet (2)

Difficulty: resource sparseness of Urdu

Approach 1:

- translating the entries in the English VerbNet to Urdu
- figure out problematic cases

Approach 2:

- fully rely on corpus work
- extend tool for automatic subcategorization extraction (Ghulam, 2010)

Can we benefit from a Hindi lexical resource?

Hindi WordNet

Facts:

- inspired in methodology and architecture by the English WordNet (Fellbaum 1998)
- **S: (n) book** (a written work or composition that has been published (printed on pages bound together)) *"I am reading"*
 - *direct hyponym / full hyponym*
 - *part meronym*
 - *has instance*
 - *direct hypernym / inherited hypernym / sister term*
 - **S: (n) publication** (a copy of a printed work offered for distribution)
 - **S: (n) work, piece of work** (a product produced or accomplished through the effort or activity or a *one of his more memorable works*"; *"the symphony was hailed as an ingenious work"*; *"he was in work of an active imagination"*; *"erosion is the work of wind or water over time"*)
 - **S: (n) product, production** (an artifact that has been created by someone or some process) *"it most of their agricultural production"*
 - **S: (n) creation** (an artifact that has been brought into existence by someone)
 - **S: (n) artifact, artefact** (a man-made object taken as a whole)
 - **S: (n) whole, unit** (an assemblage of parts that is regarded as a single entity) *"the team is a unit"*
 - **S: (n) object, physical object** (a tangible and visible entity; an entity) *"balls and other objects"*
 - **S: (n) physical entity** (an entity that has physical existence)
 - **S: (n) entity** (that which is perceived or known or inferred nonliving))

Hindi WordNet

ONTOLOGY NODES searches for NOUN (संज्ञा) form of *किताब*

Sense 1

पुस्तक, *किताब*, किताब: लिखी हुई या छपी हुई बहुत से पन्नों वाली वह वस्तु जिसमें दूसरों के पढ़ने के लिए विचार, विवेचन आदि हों; "अच्छी पुस्तक पढ़ने से ज्ञान बढ़ता है";

o मानवकृति (Artifact) (ARTFCT उदाहरण:- पुस्तक, कुसीर, नाव इत्यादि)

o वस्तु (Object) (OBJCT उदाहरण:- पुस्तक, छाता, पत्थर इत्यादि)

o निजीरव (Inanimate) (INANI उदाहरण:- पुस्तक, घर, घूँप इत्यादि)

o संज्ञा (Noun) (N उदाहरण :- गाय, दूध, मिठाई इत्यादि)

o TOP (Top Level Node)

- developed at the Indian Institute of Technology, Bombay, India
- separated into four independent “semantic nets”
 - verbs, nouns, adjectives and adverbs
- about 3.900 verbs, 57.000 nouns, 13.700 adjectives and 1.300 adverbs
- words are grouped according to their meaning similarity (“synsets”)

Hindi WordNet

Issues

- far less specific concepts than in the English WordNet

Hindi WordNet:

TOP › Noun › Inanimate › Object › Artifact › [kitAb](#)

TOP › Noun › Inanimate › Object › Artifact › [mez](#)

English WordNet:

entity › physical entity › object › whole unit › artifact › creation › product
› piece of work › publication › [book](#)

entity › physical entity › object › whole unit › artifact › instrumentality ›
furnishing › piece of furniture › [table](#)

Benefits for an Urdu VerbNet

Preliminary experiments for Urdu/Hindi verbs

- Resources that we have:
 - the database from Hindi WordNet
 - a list of Urdu verbs
- out of 3.900 Hindi verbs, we have found 534 verbs in an Urdu verb list (Humayoun, 2006)
- complex predicates are included in Hindi WordNet, but not in the Urdu wordlist
- total of around 700 Urdu verbs → more than 2/3 of Urdu verbs are found
- all found verbs seem to be valid
→ extract verb information from Hindi WordNet for the Urdu VerbNet

Urdu Lexical Semantics

Polysemy:

An extreme case - EAT expressions in Hindi/Urdu (Hook and Pardeshi, 2009):

- employing 'eat' in idiomatic expressions
- about 160 EAT expressions for Hindi/Urdu
- variety of uses due to loan translations from Persian

Urdu Lexical Semantics

h2asan=ne kEk=ko kHAyA
 h2asan.Erg cake.Acc eat.Perf.Sg.Masc
 'Hasan ate the cake.'

eat = \langle Agent, Theme \rangle

inqilAbl fikar zang kHA jAEgl
 revolutionary thought rust eat go.Fut
 'Revolutionary thinking will gather rust.'

eat (gather rust) = \langle Patient, Theme \rangle

is sAl=kl mandl sheyar-bAzAr kHA gAyl
 this year.Gen slowdown.Fem stockmarket eat go.Fut.Fem
 'This year's slowdown wrecked (lit. devoured) the stock market.'

eat (wreck) = \langle Agent, Theme \rangle

Urdu Lexical Semantics

How do we approach polysemy in the computational semantics?

- extensive corpus work to find polysemous verbs
- assign different thematic roles to polysemous verbs?
- put all combinations in the Urdu VerbNet, but mark the “original” use?
- analysis for all sentences, mark idiomatic and semantically ill-formed sentences as such?

Wrap up

What we have talked about:

- architecture of the Urdu LFG Grammar
- ongoing work
 - transliteration
 - discontinuous NPs
 - computational semantics
- challenges ahead

Demo

Thank you!