An Open Source Urdu Resource Grammar

Shafqat M Virk  
Department of Applied IT  
University of Gothenburg  
virk@chalmers.se

Muhammad Humayoun  
Laboratory of Mathematics  
University of Savoie  
mhuma@univ-savoie.fr

Aarne Ranta  
Department of CS & Eng  
University of Gothenburg  
aarne@chalmers.se
Plan

- Introduction
  - Urdu Language
  - Grammatical Framework (GF)
- Urdu Resource Grammar
  - Morphology
  - Syntax
- Attempto (An Application Grammar)
- Future Work
- Questions
Urdu Language

- Indo-European → Indo-Iranian → Indo-Aryan family
- Widely spoken in south Asia
- Closely related to Hindi
  - Phonology, morphology, syntax and day-to-day vocabulary.
  - Differs considerably in their script and scholarly writings
  - Urdu is written in a Perso-Arabic script from right to left; whereas Hindi is written in Devanagari script from left to right.

- Urdu-Hindi together
  - One of the most widely spoken language in the world with 1,017,290,000 speakers
    (Native + second language, after Chinese Rahman, 2004)

Brahui language is spoken in Pakistan, and is Dravidian
Picture from google
Grammatical Framework (GF)

- A tool for working with grammars
- Programming language for writing grammars
- A number of multilingual text generation applications (Phrasebook, Attempto, WebAlt etc) have been developed using GF and/or GF resource library

(Grammatical Framework, Ranta 2004)
http://www.grammaticalframework.org/
Levels of GF Grammars

- GF Grammars have two levels
  - Abstract Syntax
  - Concrete Syntax
Abstract Syntax

- Defines a set of Categories* and Tree building functions
- Independent of language
- Common to all languages

*Term Category is used to model different parts of speech
Abstract Syntax

- **Categories**
  - cat CN
  - cat NP
  - cat A
  - cat AP
  - cat V2

- **Functions**
  - fun PositA : A -> AP ;  -- black
  - fun AdjCN : AP -> CN -> CN ;  -- black cat
  - fun Compl : V2 -> NP -> VP ;  -- eats bread
Concrete Syntax

- Contains linearization rules for categories and trees
- Language dependent
- Each language has its own concrete syntax
Concrete Syntax [Urdu]

- **Categories**
  - \texttt{lincat CN = \{s : Number \rightarrow Case \rightarrow Str ; g : Gender\} ;}
  - \texttt{lincat AP = \{ s: Number \rightarrow Gender \rightarrow Case \rightarrow Degree \rightarrow Str \};}

- **Functions**
  - \texttt{PositA a = a ;}
  - \texttt{lin AdjCN ap cn = \{ s = \n,c => ap.s \! n \! cn.g \! c \! Posit ++ cn.s \! n \! c ; g = cn.g \} ;}
  - \texttt{Compl v2 np = np ++ v2 ; (bread eat, \text{رُتَّيِ: kʰ'ata, روئی کهاتا})}
Concrete Syntax [Urdu]

- **Categories**
  - lincat CN = \{ s : Number => Case => Str ; g : Gender \} ;
  - lincat AP= \{ s: Number => Gender => Case => Degree => Str \} ;

- **Functions**
  - PositA a = a ;
  - lin AdjCN ap cn = 
    s = \n, c => ap.s ! n ! cn.g ! c ! Posit ++ cn.s ! n ! c ;
    g = cn.g 
    
  - Compl v2 np = np ++ v2 ; (bread eat, روتی: کھاتا، روتی کھیتا)
Concrete Syntax [English]

- **Categories**
  - `lincat CN = {s : Number => Case => Str ; g : Gender};`
  - `lincat AP = {s : Agr => Str ; isPre : Bool};`

- **Functions**
  - `PositA a = { s = \_ => a.s ! AAdj Posit Nom ; isPre = True};`
  - `AdjCN ap cn = {
    s = \n,c => preOrPost ap.isPre (ap.s ! agrgP3 n cn.g) (cn.s ! n ! c);
    g = cn.g
  };

  - `Compl v2 np = v2 ++ np ;       eat bread`
Types of Grammars

- Resource Grammars
- Applications Grammars
Resource Grammars

- General purpose grammars that cover general aspects of a language linguistically
- Resource grammars encodes syntactic features of language
Application Grammars

- Typically limited to specific domains
- Encode semantic structures
- Can use resource grammars as libraries
Urdu Resource Grammar

- A resource grammar consists of
  - Lexicon
  - Grammar

- GF library currently has resource grammars for 15 languages

- Urdu is 16th in total and first South Asian language

- Almost 2700 lines of code and development time is almost seven months
Lexicon

- Test Lexicon of 350 Words
- Almost 100 Structural Words (Closed Word Category)
- The rules of defining Urdu morphology are borrowed from (Humayoun et al. 2006)
  - An Urdu morphology was developed in Haskell using Functional morphology toolkit
  - Now we have developed in GF
Morphology + Syntax

- Nouns and Noun Phrases
- Verbs and Verb Phrases
- Adjectives and Adjectival Phrases
- Clauses
- Sentences
Urdu Nouns

- Urdu Nouns inflect in
  - Number (Singular, Plural)
  - Case (Direct, Oblique, Vocative)

- Inherent Gender

Noun = \{s : Number => Case => Str ; g : Gender\}
Urdu Nouns

- Urdu Nouns inflect in
  - Number (Singular, Plural)
  - Case (Direct, Oblique, Vocative)

- Inherent Gender

Noun = \{s : Number => Case => Str ; g : Gender\}

We* have divided Nouns into 15 different groups, based on how they end, and there is one group for worst case.

* Humayoun et al 2006
Noun Phrases

(M) H (M)

NP : Type = \{s : NPCase \rightarrow Str ; a : Agr\} ;

- NPCase = NPC Case / NPErg / NPAbl / NPIIns / NPLoc1 / NPLoc2;
  - NPErg: Ergative case with case marker ‘ne: ﷲ’
  - NPAbl: Ablative with case marker ‘se: ﻦ’
  - NPIIns: Instrumental case with case marker ‘se: ﻦ’
  - NPLoc1: Locative case with case marker ‘mi: ﺢ’
  - NPLoc2: Locative case with case marker ‘pr ﺛ’
Verbs

- Urdu Verb inflects in
  - Gender (Masculine, Feminine)
  - Number (Singular, Plural)
  - Person (First, Second \{casual,familiar,respectfull\}, Third \{near,distant\})
  - Tense (Subjunctive, Perfective, Imperfective)

Verb = \{s : VerbForm => Str\}
VerbForm = VF VTense UPerson Number Gender
| Inf
| Root
Verbs

VF Subj Pers1 Sg Masc => kh-aw^N
VF Subj Pers1 Sg Fem => kh-aw^N
VF Subj Pers1 Pl Masc => kh-ay^N
VF Subj Pers1 Pl Fem => kh-ay^N
VF Subj Pers2_Casual Sg Masc => kh-a
VF Subj Pers2_Casual Sg Fem => kh-a
VF Subj Pers2_Casual Pl Masc => kh-aw^N
VF Subj Pers2_Casual Pl Fem => kh-aw^N

..........................

VF Imperf Pers1 Sg Masc => kh-ata
VF Imperf Pers1 Sg Fem => kh-aty
VF Imperf Pers1 Pl Masc => kh-atE
VF Imperf Pers1 Pl Masc => kh-atyN

..........................

Inf => kh-ana
Root => kh-a
Verb Phrases

VPH : Type = {
    s   : VPHForm => {fin, inf : Str} ;
    obj : {s : Str ; a : Agr} ;
    vType : VType ;
    comp : Agr => Str ;
    embComp: Str ;
    ad  : Str ;
} ;
Verb Phrases

VPH : Type = {
  s : VPHForm => {fin, inf : Str} ;
  obj : {s : Str ; a : Agr} ;
  vType : VType ;
  comp : Agr => Str;
  embComp : Str;
  ad : Str;
} ;

- s: {fin : Copula
  inf : actual form of verb}
- obj: object of the verb
- vType : Type of verb, will be used in Ergativity
- comp: Complement of verb
- embComp: Used in case of embedded sentences
- ad: adverb
Verb Phrases

He says that she runs.

She wants to run.

Noun ++ VP.obj ++ VP.adverb ++ VP.complement ++ VP.verb ++ VP.copula

He says that she runs.

She wants to run.

Noun ++ VP.obj ++ VP.adverb ++ VP.complement ++ VP.verb ++ VP.copula ++ VP.embComp
Adjectives

- Urdu Adjectives inflect in
  - Number (Singular, Plural)
  - Gender (Masculine, Feminine)
  - Case (Direct, Oblique, Vocative)
  - Degree (Posit, Compar, Superl)

Adjective = { s: Number => Gender => Case => Degree => Str };
Adjectival Phrases

AP = { s: Number => Gender => Case => Degree => Str };

Sg Mas Dir Posit => kala
Sg Mas Dir Compar => bht kala
Sg Mas Dir Superl => sb sE kala

Sg Fem Dir Posit => kaly
Sg Fem Dir Compar => bht kaly
Sg Fem Dir Superl => sb sE kaly
Clauses

Clause : Type = {s : VPHTense => Polarity => Order => Str} ;
- VPHTense =
  VPGenPres
  | VPPastSimple
  | VPFut
  | VPContPres
  | VPContPast
  | VPContFut
  | VPPerfPres
  | VPPerfPast
  | VPPerfFut
  | VPPerfPresCont
  | VPPerfPastCont
  | VPPerfFutCont
  | VPSubj
- Polarity = Pos | Neg ;
- Order = ODir | OQuest ;
S  = \{s : \text{Str}\}

\text{UseCl}   : \text{Temp} \to \text{Pol} \to \text{Cl} \to S

\text{UseCl} \text{\ temp p cl =}

\{ s = \text{case <temp.t,temp.a> of}

\text{<Pres,Simul> => temp.s ++ p.s ++ cl.s ! VPGenPres ! p.p ! ODir;}
\text{<Pres,Anter> => temp.s ++ p.s ++ cl.s ! VPPerfPres ! p.p ! ODir;}
\text{<Past,Simul> => temp.s ++ p.s ++ cl.s ! VPImpPast ! p.p ! ODir;}
\text{<Past,Anter> => temp.s ++ p.s ++ cl.s ! VPPerfPast ! p.p ! ODir;}
\text{<Fut,Simul> => temp.s ++ p.s ++ cl.s ! VPFut ! p.p ! ODir;}
\text{<Fut,Anter> => temp.s ++ p.s ++ cl.s ! VPPerfFut ! p.p ! ODir;}
\text{<Cond,Simul> => temp.s ++ p.s ++ cl.s ! VPSsubj ! p.p ! ODir;}
\text{<Cond,Anter> => temp.s ++ p.s ++ cl.s ! VPSsubj ! p.p ! Odir}

\}
Ergativity

- Final verb agreement is with direct subjective except in the transitive perfective tense
- In transitive perfective tense verb agreement is with direct object

Girl ate apple VS Girl ate bread.

لُزُكيّ نَهُ سَيْبُ كِهَايَ 
لُزُكيّ نَهُ رُوُثُيّ كِهَايَ 

{Fem: روُثُيّ,Fem: كِهَايَ} {Masc: سَيْبُ,Masc: كِهَايَ}
Ergativity

mkClause : NP -> VPH -> Clause = \np,vp -> {
  s = \vt,b,ord =>
  let
    subjagr : NPCase * Agr = case vt of {
      VPPast => case vp.subj of {
        (Vtrans| VTransPost) => <NP Erg, vp.obj.a> ;
        _ => <NPC Dir, np.a>
      };
      _ => <NPC Dir, np.a>
    };
    ......................
    ......................
Attempto

- A grammar for Controlled Language
- Implemented for English then was ported to Finnish, French, German, Italien, Swedish
- Ported to Urdu

Attempto Home Page: http://attempto.ifi.uzh.ch/site/
Future Work

- Bigger Lexicon (A lexicon of 6600 words has been completed recently)
- Language Specific Module (under construction)
- Hindi Resource Grammar (almost completed)
- Application Grammars (SMS translator)
Questions/Suggestions