Developing Punjabi Morphology, Corpus and Lexicon

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

Aarne Ranta
Department of CS and Engineering
Chalmers University of Technology and
University of Gothenburg, Sweden.
aarne@chalmers.se

Abstract. We describe an implementation of morphology, development of a corpus and building of a lexicon for Punjabi language. Such resources are building blocks for various language technology tasks ranging from part of speech tagging to machine translation. Their importance is further increased by the fact that Punjabi is an under resourced language. We release these resources as open-source.

Keywords: Morphology, Corpus building, Lexicon extraction, Punjabi, Shahmukhi.

1 Introduction

Punjabi is an Indo-Aryan language, widely spoken in the Punjab region of south Asia. It is spoken by 88 million people (Lewis 2009), which makes it approximately the 13th most widely spoken language in the world. Furthermore, it is the most widely spoken language of Pakistan as regards the number of native speakers.

Punjabi is normally written in two scripts: *Gurmukhi* (in India) and *Shahmukhi* (in Pakistan). The resources reported in this paper are developed using *Shahmukhi*. *Shahmukhi* is a variant of the Perso-Arabic scripts. Therefore it possesses most of their inherent characteristics such as right to left writing, an optional use of diacritic marks and short vowels being not considered as letters of their own but applied above or below a consonant by using appropriate diacritics. It has 49 consonants, 16 diacritical marks and 16 vowels; see Malik (2006) for a detailed account of both scripts.

In this paper, we report three important resources which are building blocks for various language technology tasks ranging from part of speech tagging to machine translation.

First, we report an implementation of inflectional morphology for Punjabi which is described in Section 3. We use GF (Grammatical Framework, Ranta 2004) which is a framework for developing multilingual grammar applications. GF also provides a built-in morphological analyzer and generator for the implemented languages. An overview of GF is given in Section 2.

Second, we report the development of a corpus containing 0.9 million words (941,284), which is collected partly from Wikipedia, as described in section 4.

Finally, we report a lexicon of 13,600 words (named entities:63%, lemmas of inflected words:37%; a lemma is also known as a dictionary form or a base form.). It is automatically extracted from the corpus and manually cleaned, as reported in Section 4. We have used a method proposed by (Forsberg et al., 2006) for this extraction. We describe our procedure in section 5.

We release these resources as open-source[1]. Their importance is further increased by the fact that Punjabi is an under resourced language. For these components, our source of study is the *Mahji* dialect of Punjabi which is spoken in and around Lahore and is considered to be the standard form (Akhtar 1999:10). We used simple Roman[2] transcription for Punjabi.

[1] Downloadable from homepage: www.lama.univ-savoie.fr/~humayoun/punjabi

[2] 'a', 'i', 'o' are used for short vowels; 'aa', 'ii' and 'oo' are used for the long ones. Capital letters are used for retroflex consonants. Capital 'N' used after a vowel shows nasalization. ch is pronounced as in the English word "chin".

2 The Grammatical Framework

GF, Grammatical Framework (Ranta 2004), is a programming language for multilingual grammar applications. It is related to LKB[3], XLE[4] as for its purpose, but based on functional programming and type theory. GF provides a built-in morphological analyzer and generator for the implemented languages. Since GF is a typed functional programming language, it uses similar techniques to those used by Zen toolkit (Huet 2005) and Functional Morphology (Forsberg and Ranta 2004) for its morphology component.

The morphology component of a GF grammar consists of an *inflection engine* and a *lexicon*. The inflection engine is a set of functions that take a lemma of a word and compute a finite table with all possible word forms. Each function represents a paradigm of some part of speech. However, it is the lexicon that connects the lemma to its paradigm. In GF, it is presented as a list of word-paradigm pairs.

On the other hand, every GF grammar has two components: *abstract syntax* and *concrete syntax*. An abstract syntax defines semantic conditions to form abstract syntax trees of a language with grammatical functions (fun) constructing trees belonging to categories (cat). A concrete syntax is a set of linguistic objects (strings, inflection tables, records) associated to abstract syntax trees, providing rendering and parsing. There may be multiple concrete syntaxes for an abstract syntax, allowing multilingual translation.

On usage, we can roughly divide GF grammars into resource grammars and application grammars. Resource grammars are general purpose grammars (Ranta 2009a) that are linguistic-oriented and they try to cover general aspects of a language. On the other hand, application grammars are domain specific and may use these resource grammars as software API. They are not written from scratch for each domain, but they use resource grammars as libraries (Ranta 2009b). Hence one could create an application specific grammar with a very limited linguistic knowledge. Example of such grammar applications are KeY[5], WebALT[6], MathNat (Humayoun and Raffalli 2010), etc.

The morphology we report here could be used for the implementation of Punjabi grammar under GF resource grammar library (Ranta 2009a) and also in other grammar frameworks.

3 Morphology

Morphology is a key component that enables the analysis and synthesis of all word forms in a language. Punjabi is quite similar to other Indo-European languages as regards morphology. For instance, it has concatenative inflectional morphology. However some differences can be found in case of causative verbs that also exhibit stem-internal changes in some cases. In this section we discuss Punjabi parts of speech and present our solution that explains them in GF.

Like other Arabic script based languages, a Punjabi word written in *Shahmukhi* may contain multiple tokens. However, we do not deal with such words and leave them to be dealt at syntax level. The only exception is the named entities that we report in the lexicon.

We define morphology for adjectives, adverbs, nouns, verbs and closed classes which are given in Sections 3.1 to 3.3, where the words are divided into sixteen paradigms. We've devised seven paradigms for nouns, four for verbs, three for adjectives, one for adverbs and one for named entities. Each paradigm is defined as a finite function that takes the lemma and produces a table with all possible word forms.

In abstract syntax the words are treated as constants of some lexical category. For example in the lexicon, words such as *munDaa* (boy), *nachnaa* (to dance) and *apeRna* (to reach) are declared in Figure 1. The corresponding rules in concrete syntax for Punjabi are given in figure 2, where mkN01,

[3] Lexical Knowledge Builder: http://wiki.delph-in.net/moin/LkbTop
[4] Xerox linguistics environment: http://www2.parc.com/isl/groups/nltt/xle/
[5] The Key project, Integrated Deductive Software Design: http://www.key-project.org/
[6] Web Advanced Learning Technologies: http://webalt.math.helsinki.fi/content/index_eng.html

mkV01 and mkV04 are functions taking a lemma and producing a table of all word forms. The numbers such as 01, 04 are the paradigm numbers given to them. We will describe these paradigms in more details in next sub-sections.

```
fun
    n1_boy      : N ;      -- Noun
    v1_dance    : V ;      -- Verb
    v4_reach    : V ;      -- Verb
```

Figure 1: Abstract syntax

```
lin
    n1_boy      = mkN01 "منڈا"
    v1_dance    = mkV01 "نچنا" "نچانا" "نچوانا";
    v4_reach    = mkV04 "اپڑنا"
```

Figure 2: Concrete Syntax

3.1 Nouns

Punjabi nouns are inflected in number and case. They also have an inherent gender that can be masculine or feminine. Morphologically speaking, it has five cases: direct, oblique, vocative, ablative, and locative/instrumental; the last two cases are vestigial (Shackle 2003:599). Since locative/instrumental case is extremely rare to find, we omit it and define just the first four cases in our implementation.

In concrete syntax, we define a category N as a record having a table "s" from number to case to string (Number => Case => Str) and an inherent parameter "g" of type gender, as shown below[7]:

```
lincat
  N = {s:Number=>Case=>Str; g:Gender} ;
param
  Number = Sg | Pl;
  Case = Dir | Obl | Voc | Abl ;
  Gender = Masc | Fem ;
```

Nouns can be divided into different groups based on their inflection which are mentioned by Bhatia (1993:164-166) and Shackle (2003:600-601). However, we have made suitable changes with respect to morphological context wherever needed. This results in seven groups whose description and inflections are given in Table 2 of Section 7. As a running example, we give below the definition of function mkN01 which describes the first paradigm for nouns mentioned in Table 2 and Figure 2.

```
oper  mkN01 : Str → N ;
      mkN01 xa =    let end = last xa ;  --returns the last character
                        x    = if (end=="ع") then xa else (tk 1 xa)
       in mkN  xa  (x+"ے")  (x+"ا")  (x+"یں")  (x+"ے")  (x+"اں")  (x+"و")  ("")  Masc ;
```

In the code above, mkN01 takes a string and returns the inflection table of that noun. In fourth line we say that if a word ends with "ع" then return that string unchanged[8]; otherwise take away one character from last (i.e. tk 1 xa) and return the remaining string. The "+" sign concatenates two strings. We also have used a general function for nouns (mkN) that takes nine arguments (eight strings and a gender) and produces a record as shown below:

```
oper mkN:(x1,_,_,_,_,_,_,x8:Str) → Gender → N =
   \sg_dir, sg_obl, sg_voc, sg_abl,   pl_dir, pl_obl, pl_voc, pl_abl, gender -> {
     s = table {
        Sg => table { Dir => sg_dir ;  Obl => sg_obl ; Voc => sg_voc ; Abl => sg_abl } ;
        Pl  => table { Dir => pl_dir ;  Obl => pl_obl  ; Voc => pl_voc ; Abl => pl_abl }
```

[7] Abbreviations are following: Abl=Ablative, Dir=Directive, Fem=Feminine, Masc=Masculine, Obl=Oblique, Pl=Plural, Sg=Singular, Voc=Vocative.

[8] If the singular directive form ends with "ع", root form is the same as its singular directive form. e.g. burqaa' (robe)

```
        } ; g = gender
    } ;
```

In a similar fashion, we construct paradigms for remaining nouns by using inflections given in Table 2 of Section 7.

3.2 Verbs

Verbs are the most complex part of speech in Punjabi. There are three classes of verbal sentences in Punjabi: simple, conjunct and compound (Bhatia 1993:85). We are interested only in simple verbs as they consist of one token and therefore can be captured morphologically. In contrast, conjunct and compound verbs can only be identified by the analysis of multiple words and mostly have a semantic role. Therefore, they should be handled in syntax.

Simple verbs can be classified into the following four classes. *First,* verbs having a basic stem form, direct and indirect causatives. The basic stem form can be intransitive, transitive or di-transitive. From it two other forms of that verb (direct and indirect causatives) can be formed. These three are regular verbs having a distinct meaning. As an example, consider a verb *nachnaa* (to dance) in table 1, which was first mentioned in figure 2.

Table 1: A verb having direct and indirect causatives

	Root	**Infinitive**	**Oblique**
Basic form	nach	nachnaa (to dance)	nachan
Direct causative	nachaa	nachaanaa (made to dance by self)	nachaan
Indirect causative	nachvaa	nachvaanaa (made to dance by someone)	nachvaan

Second, verbs having only basic and direct causative forms, while the indirect causative does not exist. An example of such verb is *piinaa* (to drink) whose direct causative infinitive form is *peaanaa* (made to drink).

Third, verbs having only a basic and indirect causative forms, while the direct causative does not exist. The verb *niighalnaa* (to swallow) and its indirect causative infinitive *niighalwaanaa* (made to swallow) is an example.

Finally, verbs having only basic stem form, while direct and indirect causatives do not exist. An example of such class is *oodiikanaa* (to wait). Our experiment on lexicon extraction (Section 5) shows that second and third class of verbs are very few as compared to the first and fourth.

Punjabi verb conjugations are regular. Furthermore the basic stem forms and causatives inflect in a similar way. Verbs inflect for tense/mood (subjunctive, perfective, imperfective), person (first, second {casual, formal}, third {near, distant}), number and gender. We show below the data structure that we use for the first verb class as a running example[9].

```
param
        Tense       = Subj | Perf | Imperf;
        Person      = Pers1 | Pers2Casual | Pers2Familiar | Pers3Near | Pers3Far;
        VerbForm1   = Basic Tense Person Number Gender
                    | Caus1 Tense Person Number Gender
                    | Caus2 Tense Person Number Gender
                    | Inf | Inf_Fem | Inf_Obl | Ablative | Root
                    |Caus1_Inf | Caus1_Fem | Caus1_Obl | Caus1_Abl
                    |Caus2_Inf | Caus2_Fem | Caus2_Obl | Caus2_Abl ;
oper V1 = {s:VerbForm1 => Str} ;
```

[9] Abbreviations: Subj=Subjunctive, Perf=Perfective, Imperf=Imperfective, Pers1=First person, Pers2Casual=Second person casual, Pers2Familiar=Second person familiar, Pers3Near=Third person near, Pers3Far=Third person far, Basic=Basic form of verb, Caus1=Direct causative, Caus2=Indirect causative, (Inf=Infinitive, Inf_Fem=Feminine infinitive form, Inf_Obl=Infinitive oblique) of basic verb, (Caus1_Inf=Infinitive, Caus1_Fem=Feminine, Caus1_Obl=oblique, Caus1_Abl=ablative) of direct causative, etc.

We have defined below a function mkV01 which takes three verb forms and builds the complete inflection table, producing 193 word forms. In the **bold-face** lines of code below, we can see that the same function "mkCmnVF" is used for basic, direct and indirect verb forms. It is so because they inflect in a similar way. The inflection table of basic stem form for verb "*nachnaa*" is shown in Table 3 of Section 7.

```
mkV01 : (x1,x2,x3: Str) -> V1 = \inf, caus1, caus2 ->
    let root   = (tk 2 inf); root1 = (tk 2 caus1); root2 = (tk 2 caus2)
    in {s = table {
        Root            => root;
        Inf             => inf;                 Inf_Fem     => ((tk 1 inf) + "ی");
        Inf_Obl         => (tk 1 inf);          Ablative    => ((tk 1 inf) + "وں");
        Caus1_Inf       => caus1;               Caus1_Fem   => ((tk 1 caus1) + "ی");
        Caus1_Obl       => (tk 1 caus1);        Caus1_Abl   => ((tk 1 caus1) + "وں");
        Caus2_Inf       => caus2;               Caus2_Fem   => ((tk 1 caus2) + "ی");
        Caus2_Obl       => (tk 1 caus2);        Caus2_Abl   => ((tk 1 caus2) + "وں");
        BVF t p n g     => mkCmnVF root   t p n g;
        Caus1 t p n g   => mkCmnVF root1 t p n g;
        Caus2 t p n g   => mkCmnVF root2 t p n g
  } };
```

The way these verbs are defined in the lexicon is shown in Figure 2. In a similar way the data structures VerbForm2, VerbForm3 and Verbform4 are defined for constructing the V2, V3, and V4 categories which represent the second, third and fourth verb class respectively, discussed in the beginning of this section.

3.3 Adjectives, Adverbs and Closed Classes

We may divide adjectives into two groups: marked and unmarked. Marked adjectives inflect in number, case, and gender, and mostly end at (aa, ا). We define their inflection below for an example word *niilaa* (blue).

	Masculine				**Feminine**			
	Dir	**Obl**	**Voc**	**Abl**	**Dir**	**Obl**	**Voc**	**Abl**
Sg	niilaa	niile	niile/niilea	niileoN	niilii	niilii	niilii/niiliie	niiliioN
Pl	niile	niileaaN	niileo	Ø	niilii	niiliaaN	niilileo	Ø

Unmarked adjectives do not inflect at all. Examples of them are: *khush* (happy), *dilchasp* (interesting), *laal* (red), etc.

Punjabi adverbs are unmarked and normally do not inflect. Examples of them are: *chhetii* (quickly), *holii* (slow), etc. We also define some of the closed classes such as pronouns, postpositions, conjunctions, negations, questions, numerals, names of months, etc.

4 Corpus Development

It has already been realized that free resources on Internet such as blogs, web pages, Wikipedia, etc could be used to build a large corpus. A reference to such work among many is (Scannell 2007) which uses web crawling for automatic text collection.

Even this method cannot help much for Punjabi written in *Shahmukhi*. It is so because; first, extremely few people write Punjabi in *Shahmukhi* on the internet; second, most of the *Shahmukhi* texts we find on the Internet is published in graphic format which makes it unfit for language processing.

However, Punjabi *Shahmukhi* version of Wikipedia[10] is a hope for this grave situation. Although it is rather small and normally has a large number of named entities, but it could still be used as a corpus. Furthermore, we were able to get hold of some Punjabi texts from modern literature[11].

To have a plain text from Wikipedia, we ripped off html/Wikipedia tags[12]. The modern literature of Punjabi was already in text form and required no ripping. Column **A** shows their word counts.

Then we rip the corpora on sentences and deleted multiple occurrences of the sentence. This step is necessary for the Wikipedia text to get rid of template sentences which appear on every page. The results are shown in column **B**.

Finally, we tokenized on words and deleted multiple occurrences; the results are shown in column **C**. At this stage, we also deleted all non-Shahmukhi characters such as: digits, western alphabets, punctuation marks for both Roman and Arabic, miscellaneous symbols, etc. Percentages given in column **B** and **C** are with respect to column **A**.

Due to the different nature of the corpora, we decided to run this experiment three times; one for Wikipedia text, second for modern literature and third for combined corpora as shown below.

	Word count		
	A: Plain	**B:** Unique sentences	**C:** Unique words
Wikipedia	327,372	307,047 (93.79%)	22,167 (6.77%)
Modern literature	651,227	625,111 (95.98%)	46,881 (7.19%)
Wikipedia and Modern literature	**978,599**	**941,248 (96.18%)**	**51,607 (5.27%)**

Like other Perso-Arabic script based languages, Punjabi in *Shahmukhi* exhibits both space insertion and deletion problems. For instance during the procedure, many erroneous entries were found in which spaces were lacking between words. Similarly many erroneous words were found which were parts of multi-token words and should not be separated on a white space.

The space deletion problem is partially solved with the fact that there are two letters which always end a word; (baRii yai, ے) and (hamza, ء). For instance, we corrected 4,711 words ending with (baRii yai, ے) from the corpus having unique sentences (mentioned in column B). Similarly there are letters such as (alif, ا) and (alif-maddaa, آ) that cannot appear next to each other. So if they appear (such entries were few), we should insert a space between them. To extract multi-token words, a list of common prefixes and suffixes could be used. However we have not done it for the work reported here. Proper names are the only category that may contain multi-token words and we've extracted them mostly from Wikipedia.

A corpus of 0.9 million (941,284) words is made publicly available. As we are unaware of the copyright information of the modern literature, we've alphabetically sorted the sentences of both corpora to mix them well.

5 The Lexicon

Building a lexicon manually is a tedious and laborious task. Therefore we have performed an experiment of automatic extraction of Punjabi lexicon using the *extract* tool (Forsberg et al. 2006). In this tool, the user provides rules about how words of different paradigms inflect. The tool then analyses morphological endings of words accordingly and collects a lexicon tagged with those paradigm names.

[10] Dump downloaded from: http://dumps.wikimedia.org/pnbwiki/20100402/

[11] Most of the material is published as graphic images on the website of Academy of the Punjab in North America: http://www.apnaorg.com/. For detailed bibliography, see section 7. Thanks to M. G. Abbas Malik for sharing these texts with us.

[12] To remove tags, we modified the script found at http://wiki.apertium.org/wiki/Building_dictionaries

We now present here two rules from the list that is defined for Punjabi, reported in section 7.2. These are written in propositional logic. The rules given below represent the first paradigm for nouns given in table 2 in section 7 and the first verb class mentioned in section 3.2.

rule n1 = x+"ا" {(x+"ا" & (x+"ے" | x+"یا" | x+"یوں" | x+"یاں")) & (~x+"انا" | ~x+"ندا" | ~x+"دا") };
rule v1= x +"نا" x+"انا" x+"وانا" {(x+"نا" | x+"دا" | x+"اندا" | x+"انا" | x+"وانا") };

The tool extracts the lexicon for both rules in the following form which we later translate into the GF lexicon format.

n1 آرا	v1 نچوانا نچانا نچنا
n1 استرا	v1 اُبھروانا اُبھارنا اُبھرنا
n1 منڈا	v1 اپڑوانا اپڑانا اپڑنا

The result for extracted lexicon may vary with the knowledge of the lexical distribution of the language and the level of strictness in the rules. A stricter definition will result in fewer words but more accuracy and vice versa. For example, the following rule will return fewer results as compared to the rule 'n1' given above, because the conditions are made stricter by replacing disjunction (|) with conjunction (&).

rule n1_ = x+"ا" {(x+"ا" & x+"ے" & x+"یا" & x+"یوں" & x+"یاں") & (~x+"انا" | ~x+"ندا" | ~x+"دا") };

For our rules, we tried to be in the middle of accuracy and coverage. To have a high level of confidence on correctness of reported lexicon, we manually checked it from word to word; and all incorrect entries were removed resulting in a lexicon of 13,600 words (named entities:63%, lemmas:37%).

In the process of corpus and lexicon building, we made the following observations. First, most high frequent words are postpositions, auxiliaries, particles and pronouns. That is why unique words in corpus are considerably less than the total words (5.27% of total words).

Second, as already said, Punjabi in Shahmukhi exhibits both space insertion and deletion problems. This is a part of the reason why the extracted and manually checked lexicon (13,600 words) is considerably less (26.3%) then the unique word count of corpus (51,607 words). Another reason could be the fact that we use Wikipedia text which normally contains a large number of named entities. For instance, in 500 most frequent words, we found 26 proper nouns.

Third, it is not possible to get a fully vocalized corpus as Punjabi is mostly written without or with a varying number of diacritic marks. Because of that, sometimes we have found more orthographic versions per word with different diacritic information. We have saved all such version in the lexicon as sometimes they may represent different words having different meanings.

6 Related Work and Conclusion

Most of the applications related to Punjabi language processing only support *Gurmukhi* script. Further most of the attention is given to Punjabi which is written and spoken in India. A review of such applications is given by Lehal (2009).

We may also find some transliteration systems[13] to interchange *Shahmukhi* and *Gurmukhi*, but they only seem to be a partial solution. In our opinion, it is so because Punjabi is not only written in two different scripts but also it borrows vocabulary from different sources. For instance, Punjabi spoken and written in Pakistan borrows words mostly from Arabic, Persian and Urdu. On the other hand, Punjabi spoken and written in India borrows mostly from Sanskrit and Hindi.

[13] One such system is (Malik 2006).

With the best of our knowledge, we are unaware of any morphological analyzer, lexicon or corpus publicly available for Punjabi written in *Shahmukhi* and spoken in Pakistan. It is one of the main motivations to develop these resources and publish them as open source.

In this paper we've presented an implementation of Punjabi morphology with fairly good coverage. Being a high level programming language, GF provided us sufficient freedom and ease. Further, dealing word classes and their parameters as algebraic data types, and the inflection tables (paradigms) as finite functions satisfying the completeness, makes this implementation elegant, modular, extensible and reusable.

We also discussed some of the problems related to corpus and lexicon development for *Shahmukhi*. One such problem is the fundamental limitation of our lexicon, as it is not fully vocalized. Further, for morphological analysis of a word, system requires an exact match. Another shortcoming of our work is the lack of support for multi-token words which is postponed to be dealt at syntax level.

In future, these components could be utilized for a Punjabi grammar under GF resource library (Ranta 2009a) but also adapted to other grammar frameworks.

References

Akhtar, R. Nasim, 1999. *Aspectual Complex Predicates in Punjabi.* Department of Language and Linguistics, University of Essex.

Bhatia, Tej K., 1993. *Punjabi: a cognitive-descriptive grammar*. Routledge, ISBN 9780415003209.

Forsberg M., Hammarström H., Ranta A., 2006. *Lexicon Extraction from Raw, Text Data.* In: Salakoski, T.and Ginter, F. and Pyysalo, S. and Pahikkala, T.(eds.) Advances in Natural Language Processing: Proceedings of the 5th International Conference, FinTAL, Finland, August 23-25, 2006, pp. 488-499 SPRINGER, LNCS 4139.

Forsberg, M. and Ranta, A., 2004. *Functional Morphology*. In ICFP 2004, Showbird, Utah, pages 213–223.

Huet, G., 2005. *A Functional Toolkit for Morphological and Phonological Processing, Application to a Sanskrit Tagger*. The Journal of Functional Programming, 15(4):573–614.

Humayoun, M. and Raffalli, C., 2010. *MathNat - Mathematical Text in a Controlled Natural Language.* Special issue: Natural Language Processing and its Applications. Journal on Research in Computing Science. Volume 46. ISSN:1870-4069.

Lehal G. S., 2009. *A Survey of the State of the Art in Punjabi Language Processing*, Language In India, Volume 9, No. 10, pp. 9-23 (2009).

Lewis, M. Paul (ed.), 2009. *Ethnologue: Languages of the World, Sixteenth edition*. Dallas, Tex.: SIL International. Online: http://www.ethnologue.com/.

Malik, M. G. Abbas., 2006. *Punjabi Machine Transliteration.* Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pp 1137-1144.

Ranta A., 2009a. *The GF Resource Grammar Library: A systematic presentation of the library from the linguistic point of view*. Linguistics in Language Technology, 2(2).

Ranta A., 2009b. *Grammars as Software Libraries. In Y. Bertot, G. Huet, J-J. Lévy and G. Plotkin (eds), From Semantics to Computer Science*, Cambridge University Press, Cambridge, pp. 281-308.

Ranta A., 2004. *Grammatical Framework: A Type-Theoretical Grammar Formalism.* The Journal ofFunctional Programming 14(2) (2004) 145–189.

Scannell, K. P., 2007. *The Crúbadán Project: Corpus building for under-resourced languages*, C. Fairon, H. Naets, A. Kilgarriff, G-M de Schryver, eds., "Building and Exploring Web Corpora", Proceedings of the 3rd Web as Corpus Workshop in Louvain-la-Neuve, Belgium.

Shackle, C., 2003. *Panjabi*, in Cardona, George; Jain, Dhanesh, The Indo-Aryan Languages, Routledge, pp. 581–621, ISBN 9780415772945.

7 Appendix

Table 2: Noun groups with inflections

#	Paradigm	Num	Root	Dir	Obl	Voc	Abl
1	Masculine nouns ending in (alif,ا), (hey,ہ) and (ein, ع) e.g. munDaa (boy), k'oRaa (horse).	*Sg* *Pl*	munD	munDaa munDe	munDe munDeaaN	munDeaa munDeo	munDeoN Ø
2	Masculine nouns such as shehar (city), din (day), hath (hand), mard (man).	*Sg* *Pl*	shehar	shehar shehar	shehar sheharaN	sheharaa/ shehar sheharo	sheharoN Ø
3	Feminine nouns ending in (ii, ی) e.g. sakhii (friend), kuRii (girl), bolii (language).	*Sg* *Pl*	kuRii	kuRii kuRiaaN	kuRii kuRiaaN	kuRie kuRio	Ø Ø
4	Feminine nouns such as gal (thing), saltanat (kingdom), kitaab (book).	*Sg* *Pl*	kitaab	Kitaab kitaabaaN	kitaab kitaabaaN	kitaabe/ kitaab kitaabo	kitaaboN Ø
5	Feminine nouns such as maaN (mother), kaaN (crow), gaaN (cow). Rare to find.	*Sg* *Pl*	maa	maaN maavaaN	maaN maavaaN	maaN/ maae maavo	maavoN Ø
6	Masculine nouns such as kuNvaaN (well). Rare to find.	*Sg* *Pl*	kuNv	kuNvaaN kuNveN	kuNveN kuNveaaN	kuNveN kuNweo	kuNvoN Ø
7	Masculine nouns such as aaTaa (flour), baabaa (old man). Rare to find.	*Sg* *Pl*	aaT	aaTaa aaTaa	aaTe aaTe	aaTe/ aaTeaa aaTe/ aaTiaa	aaTeoN Ø

Table 3: Conjugation of verbs nachnaa (to dance) and khaanaa (to eat)

<table>
<tr><td colspan="6">nachnaa (to dance) and khaanaa (to eat)</td></tr>
<tr><td colspan="6">Root: nach, khaa*</td></tr>
<tr><td colspan="6">Subjunctive</td></tr>
<tr><td rowspan="2"></td><td rowspan="2">First Person</td><td colspan="3">Second Person</td><td colspan="2">Third Person</td></tr>
<tr><td>Casual</td><td colspan="2">Familiar</td><td>Near</td><td>Distant</td></tr>
<tr><td>Sg. Masc/Fem</td><td>nachaaN/khaawaaN</td><td>nach/khaa</td><td colspan="2" rowspan="2">nacho/khaa’o</td><td colspan="2">nachee/khaa’ee</td></tr>
<tr><td>Pl. Masc/Fem</td><td>nachiiee/khaaiiee</td><td>nacho/khaa’o</td><td colspan="2">nachan/khan, khaawan</td></tr>
<tr><td colspan="6">Perfective</td></tr>
<tr><td>Sg. Masc.</td><td colspan="2">nacheaa/ khaaeaa</td><td colspan="2">nachee/ khaa’ee</td><td colspan="2">nacheaa/ khaaeaa</td></tr>
<tr><td>Sg. Fem.</td><td colspan="2">nachii/ khaaii</td><td colspan="2">nachiiaaN/ khaaii’aaN</td><td colspan="2">nachii/ khaa’ii</td></tr>
<tr><td>Pl. Masc.</td><td colspan="3">nachee/ khaa’ee</td><td colspan="2">nachee/ khaa’ee</td></tr>
<tr><td>Pl. Fem.</td><td colspan="3">nachiiaaN/ khaaii’aaN</td><td colspan="2">nachiiaaN/ khaa’iiaaN</td></tr>
<tr><td colspan="6">Imperfective</td></tr>
<tr><td>Sg. Masc.</td><td colspan="2">nachedaa/ khaandaa, khaaondaa</td><td colspan="2">nachedee/ khaandee, khaaondee</td><td colspan="2">nachedaa/ khaandaa, khaaondaa</td></tr>
<tr><td>Sg. Fem.</td><td colspan="2">nachedii/ khaandii, khaaondii</td><td colspan="2">nachediiaaN/ khaandiiaaN, khaaondiiaaN</td><td colspan="2">nachedii/ khaandii, khaaondii</td></tr>
<tr><td>Pl. Masc.</td><td colspan="5">nachedee/ khaandee, khaaondee</td></tr>
<tr><td>Pl. Fem.</td><td colspan="5">nachediiaaN/ khaandiiaaN, khaaondiiaaN</td></tr>
<tr><td colspan="6">* If a root ends with one of (ا , و , آ, ی) then it inflects similar to verb khaanaa. To be precise, this is how direct and indirect causatives inflect</td></tr>
</table>

7.1 Details of collected modern literature

1. *Jornal puncham*. 2004. *Sucheet kitaab ghar.11 Shraf mansion, 16 Queens Road Lahore, Pakistan.* We have four different editions.
2. Two book by Najam Hussain Syed (*saedhaaN*, *saaraaN*)
3. Plays by Lakhat pasha (*thall* and *ik probharaa akharr*)
4. *Kahaniyan Sujag* - miscellaneous short stories by famous writers. Some writers are: Farkhanda Lodhi, Akbar Lahori, Jaswant singh kanwal, Azra waqar, Rana Gulam Shabbir, Azra waqar, paraim parkash, etc.

7.2 Rules to extract lexicon

Nouns: *Correspond to the paradigms in table 2.*

rule n1 = x+"ا" { (x+"ا" & (x+"ے" | x+"یا" | x+"یوں"| x+"یاں")) & (~ x+"انا" | ~ x+"ندا" | ~ x+"دا") };
rule n1 = x { x+ & (x+"ے" | x+"یا" | x+"وں")}; *--brqe (robe)*
rule n2 = x { (x & (x+"ا" | x+"وں" | x+"اں")) & (~ x+"ی" | ~ x+"یاں") };
rule n3 = x+"ی" { (x+"ی" & x+"یاں") };
rule n4 = x { (x & x+"ان" & (x+"وں" | x+"و" | x+"ے")) };
rule n5 = x+"ں" { ((x+"ں" & x+"واں") & (x+"ئے" | x+"وو")) };
rule n6 = x+"واں" { (x+"واں" & (x+"ویں" | x+"ویاں" | x+"ویو")) };

Verbs:

rule v1= x +"نا" x+"انا" x+"وانا" {(x+"نا" | x+"دا" | x+"ندا" | x+"انا" | x+"وانا") };
rule v2= x +"نا" x+"انا" {(x+"نا" | x+"انا") & ~ (x+"وانا")};
rule v3= x +"نا" x+"وانا" {((x+"نا" & x+"وانا") & ~(x+"انا")) };
rule v4 = x+"نا" { (x+"نا" & ~(x+"انا") & ~(x+"وانا")) };

Adjectives:

rule adj1 = x+"ا" { (x+"ا" & x+"ے" & x+"ی") };

7.3 100 highest frequency words of Punjabi

تے **29077**	دے **21644**	اے **15663**	دی **15441**	دا **14108**	سی **12190**
نوں **10876**	چ **10513**	کے **8524**	توں **7381**	نے **7353**	وی **7092**
نال **6920**	اوہ **6258**	وچ **6167**	اک **5368**	نہیں **4889**	ایہہ **4745**
ای **4598**	نیں **4277**	ہے **3801**	میں **3650**	پر **3642**	سن **3578**
کہ **3404**	اس **3235**	نی **3064**	کر **3019**	لئی **2969**	گیا **2933**
اوس **2918**	وِچ **2910**	تاں **2879**	دیاں **2830**	ایس **2644**	ہو **2632**
کوئی **2394**	اوہناں **2375**	گئی **2070**	کیتا **1924**	گل **1814**	اِک **1801**
جاندا **1732**	شہر **1663**	اوہدے **1663**	والے **1581**	تک **1536**	دتا **1529**
مینوں **1474**	ناں **1402**	اوہنوں **1389**	آپنے **1389**	میرے **1385**	جدوں **1374**
گئے **1368**	گھر **1350**	ہن **1344**	کم **1343**	کرن **1336**	رہیا **1330**
ہور **1314**	ہویا **1294**	ول **1291**	ہوئی **1271**	لگ **1225**	کسے **1222**
جس **1220**	پئی **1220**	سلطنت **1219**	لے **1213**	جا **1178**	کیہ **1156**
اپنے **1150**	جے **1149**	کیتی **1147**	پھیر **1143**	بعد **1113**	نُوں **1087**
پیا **1083**	ویلے **1080**	دو **1079**	جی **1078**	ہووے **1076**	کجھ **1065**
جیویں **1048**	ہوئے **1046**	رہی **999**	گئیا **997**	ہاں **993**	ہوندا **981**
سارے **949**	بن **948**	پاکستان **940**	ہی **935**	اُتے **928**	سنگھ **926**
اج **915**	جان **908**	ہر **907**	لیا **900**		