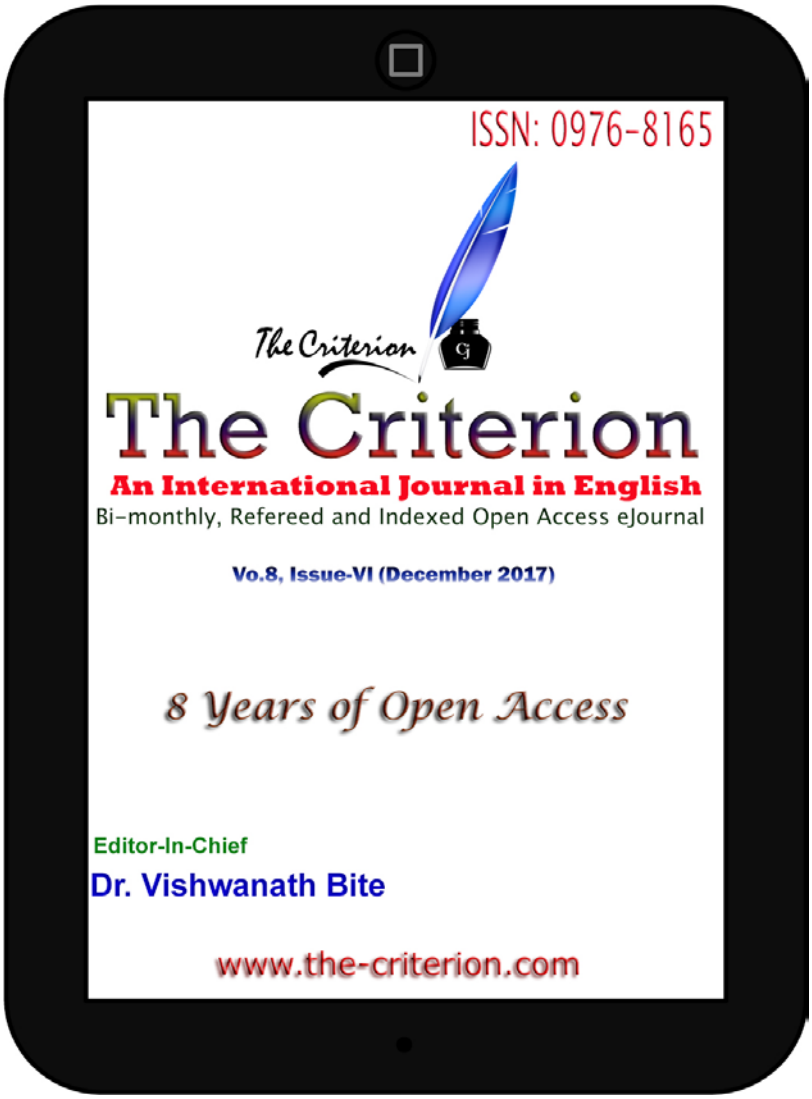

About Us: http://www.the-criterion.com/about/

Archive: http://www.the-criterion.com/archive/

Contact Us: http://www.the-criterion.com/contact/

Editorial Board: http://www.the-criterion.com/editorial-board/

Submission: http://www.the-criterion.com/submission/

FAQ: http://www.the-criterion.com/fa/

Generating Dogri Morphological Analyzer Using Apertium Tool: An Overview

Sunil Kumar
Senior Resource Person (Academics)
National Translation Mission
Central Institute of Indian Languages,
Mysore

Article History: Submitted-06/12/2017, Revised-13/12/2017, Accepted-15/12/2017, Published-31/12/2017.

Abstract:

Computational morphology is a subfield of computational linguistics (also called "natural language processing" or language engineering). Computational morphology concerns itself with computer applications that analyze words in a given text, such as determining whether a given word is verb or a noun. Almost all practical applications that deal with natural language must have a morphological component. After all, an application must first recognize the word in question before analyzing it syntactically, semantically, or whatever the case may be. The term morphology is generally attributed to the Johann Wolfgang von Goethe (1749–1832) a German poet, playwright, novelist, and philosopher who coined it early in the nineteenth century in a biological context. Its etymology is Greek: morph- means 'shape, form', and morphology is the study of form or forms. In linguistics morphology is thestudy of the smallest grammatical units of language, and of their formation into words, including composition, derivation and inflection. Although linguists may argue for other definition of morphology, they mostly agree with morphology is the study of meaningful parts of words (McCarthy, 1991). So, Morphological Analyzer is an integral part of any Natural Language Process system, especially in the context of Indian languages. The task of the Morphological Analyzer is to identify the structural components of a word and used for information extraction. In the Dogri word /jAgatO/, for example there are two meaningful units: {jAgat) and the vocative case marker {O}. Such units called morphemes, minimal meaning-bearing unit in a languageand are the smallest units of morphological analysis. Morphemes that precede the stem or root are called prefixes, such as {ana} in Dogri /anapaDha'EA/. Those that follow are called suffixes, such as {O} in/jAgatO/.This paper describes an ongoing effort to develop Dogri morphological analyzer, using an open source platform-Apertium (LT-Toolbox).

Keywords: Morphological Analyzer, Apertium, Dogri morphology, Paradigm, Dictionary building.

An Overview of Dogri:

Dogri is one of the modern Indo-Aryan languages along with Punjabi which have developed tonal contrasts. It has three tones: low / ˩ / mid / - / and high / ˥ /. Dogri is a morphologically rich language having the pre-dominant word order of Subject-Object-

Verb (SOV) with a flexibility to rearrange the constituents as many Indian languages allow. Nouns are generally inflected for number, gender and case. There are two numbers –singular and plural; two genders-masculine and feminine; and three cases- simple, oblique and vocative. The oblique forms occur when a noun or noun phrase is followed by a postposition. Nouns are inflected according to their gender and the word final sound. Dogri is primarily spoken in the Jammu and Kashmir state and the adjoining areas of Himachal Pradesh, Punjab and across the border in Sialkot and Shakargarh tehsils presently in Pakistan. As language part of the Census of India 2011 is not available so according to the Census of India 2001 the number of Dogri speakers are 22,82,589. The History of modern Indo-Aryan languages such as Hindi, Marathi, Gujarati, Assamese, Bengali, Odia, Punjabi and Dogri can be traced to its earlier stages-Old Indo-Aryan language (1500 B.C. to 600 B.C.) and Middle Indo-Aryan language (600B.C. to 1000 A.D.). Dogri has its own script namely "dogreakkhar" or "dogre" based on Takri script which is closely related to the Sharada script employed by Kashmiri language. This script was the official language of Jammu & Kashmir state during the regime of Maharaja Ranbir Singh (1857-1885 A.D.) After the independence the state government constituted a committee on 29th October, 1953 headed by Sh. GirdhariLalDogra presented a report and accordingly the state government decided to adopt Devnagri as well as Persian script for Dogri and it was incorporated in the State Constitution in 1957. So at present the Devnagri script is mainly used in India and the Nasta'liq form of Perso-Arabic in Pakistan.

Types of Morphology

Inflectional morphology: in inflectional morphology the new word formed is of same class as that of previous word i.e. if previous word is of class noun then after adding affixes to it will remain a noun. This can be better understood by an example as कुड़ी (Girl) becomes कुड़ियां (Girls) on adding इयांas suffix.

Derivational morphology: in derivational morphology new word formed will be of different category than the previous word for example- बेशर्म (brazen) (Adj) becomes बेशर्मी (brazenness) (Noun) on adding ई as suffix.

Since, it is the initial stage of this task; we have confined our work to inflectional morphology so derivational and compound morphology will be implemented latter.

Apertium or LT-toolbox and Word and Paradigm Model (WPM)

Lttoolbox is a lexical and morphological analyzer package from the Apertium machine translation system. It was developed through a number of projects like Open-Source Machine Translation for the Languages of Spain and EurOpenTrad: Open-Source Advanced Machine Translation for the European Integration of the Languages of Spain by the Transducers Research Group. It is used for lexical processing, morphological analysis and generation of words. The Word and Paradigm model has been used here for

the present approach which involves a practical adoption of the lttoolbox in order to build improvised open source morphological analyzers and generators for Dogri language. The tool uses the computational algorithm called Finite State Transducers for both analysis and generation. It confirmed that the relevant module can be utilized to develop morphological analyzers and generators for all Indian languages. The only effort that one has to be made is to add language specific data to the ready-made tool.

POS Tag set used for morph:

We have used the morpho-syntactic Parts-Of-Speech tag set for morph based on the three level-hierarchies which includes Category, Type and Attributes/Values for capturing inflectional morphology of Dogri.

1. **Noun: -**

Category	Type	Attributes	Examples
Noun(N)	Common (NC)	Gender, Number, Case, Case marker,	जागतें/NC.mas.pl.obl.0
	Proper(NP)	Gender, Number, Case, Case marker.	राम/NP.mas.sg.0.0
	Verbal (NV)	Case, Case marker.	पीने/NV.obl.0
	Spatio-temporal (NST)	Case, Case marker,Dimension.	बाह्रा/NST.obl.abl.prx

2. **Pronoun:**

Category	Type	Attributes	Examples
Pronoun (P)	Pronominal (PPR)	Gender, Number, Person, Case, Case marker, Emphatic, Dimension, Honorificity.	असें/PPR.0.pl.1.obl.erg.n.n.n
	Reflexive (PRF)	Gender, Number, Case, Case marker	आपूं/PRF.0.0.dir.0
	Reciprocal (PRC)	Case.	इक-दुए/PRC.obl
	Relative (PRL)	Gender, Number, Case, Case marker, Emphatic, Honorificity.	जेह्ड़ा/PRL.mas.sg.obl. gen.n.n
	Wh-pronoun (PWH)	Gender, Number, Case, Case marker, Emphatic, Honorificity.	कोह्दा/PWH.mas.sg.obl.gen.n.n

3. **Demonstrative:**

Category	Type	Attributes	Examples
	Absolutive (DAB)	Gender, Number, Dimension, Emphatic.	एह/DAB.0.0.prx.n

Demonstrative (D)	Relative Demonstrative (DRL)	Gender, Number	जेह्ड़ा /DRL.mas.sg
	Wh-demostrative (DWH)	Gender, Number	केह्ड़ा /DWH.mas.sg.

4. Nominal Modifier:

Category	Type	Attributes	Examples
Nominal Modifier (J)	Adjective (JJ)	Gender, Number, Case,	शैल/JJ.0.0.dir.
	Quantifier (JQ)	Gender, Number Case, Emphatic, Numeral.	मता/JQ.mas.sg.dir.n.0
	Intensifier (JINT)	Gender, Number, Case.	बड़ी/ JINT.fem.sg.dir

5. Verb:

Category	Type	Attributes	Examples
Verb (V)	Main Verb (VM)	Gender, Number, Person, Tense, Aspect, Mood, Negation, Finiteness, Honorificity.	जंदा/VM.mas.sg.0.0.pft.dcl.n.fnt.n
	Auxiliary Verb (VA)	Gender, Number, Person, Tense, Aspect, Mood, Negation, Finiteness, Honorificity.	ऐ/VA.0.sg.3.prs.prg.dcl.n.fnt.n

6. Adverb:

Category	Type	Attributes	Examples
Adverb(A)	Manner (AMN)	Gender, Number, Case.	बल्लें/AMN.0.0.0

7. Postposition:

Category	Type	Attributes	Examples
Post- position(PP)	Case(PP)	Gender, Number, Case marker, Honorificity.	दियां/PP.fem.pl.gen.n

8. Particle:

Category	Type	Attributes	Examples
	Co-ordinating (CCD)		ते/CCD
	Subordinating (CSB)		जेकर/CSB

Particle (C)	Interjection (CIN)	Gender, Number, Case Marker.	अड़िये/CIN.fem.sg.0
	(Dis)Agreement (CAGR)		नेईं/CAGR
	Emphatic (CEMP)		गै/CEMP
	Topic (CTOP)		ते/CTOP
	Delimitive (CDLIM)		मात्र/CDLIM
	Honorific (CHON)		होरCHON/
	Dedative (CDED)		बारै/CDED
	Exclusive (CEXCL)		बगैर/CEXCL
	Interrogative (CINT)		कीह्/CINT
	Dubitative (CDUB)		खबरै/CDUB
	Similative (CSIM)	Gender, Number, Case.	जनेह् /CSIM
	Others (CX)	Gender, Number, Case.	आह्ला/CX.mas.sg.dir

Transliteration scheme Tool:

Since the tool supports Roman characters only, Indic words need to be transliterated first before one starts the actual work. For this purpose a well defined transliteration module is required. There are many Romanizing tools available based on different schemes. One such tool has been developed by LDC-IL for romanizing all Indian Scripts. In this model, we have used the Linguistic Data Consortium for Indian Languages transliteration tool which is primarily based on different mapping schemes but script grammar rules have been also incorporated to set constraints on the mapping process for better results. The transliteration tool has made the present task very easy. The Roman-Devanagari transliteration scheme for Dogri is as under:

अ-a, आ-A, इ-i, ई-I, उ-u, ऊ-U, ए-E, ऐ-ai, ओ-O, औ-au, ऋ-x, अ:- aH, अं-aM. क-ka, ख-kha, ग-ga, घ-gha, ङ-ng'a, च-ca, छ-cha, ज-ja, झ-jha, ञ-nj'a, ट-Ta, ठ-Tha, ड-Da. ढ-Dha, ण-Na, त-ta, थ-tha, द-da, ध-dha, न-na, प-pa, फ-pha,ब-ba, भ-bha, म-ma, य-ya, र-ra, ल-la, व-va,श-sha, स-sa, ह-ha, ड़-D'a, ढ़-D'ha, क्ष-kSa, त्र-tra, ज्ञ-nj'a, Whereas " represents ' {sur} symbol of Dogri and \s stands for SDevanagariavagraha character.

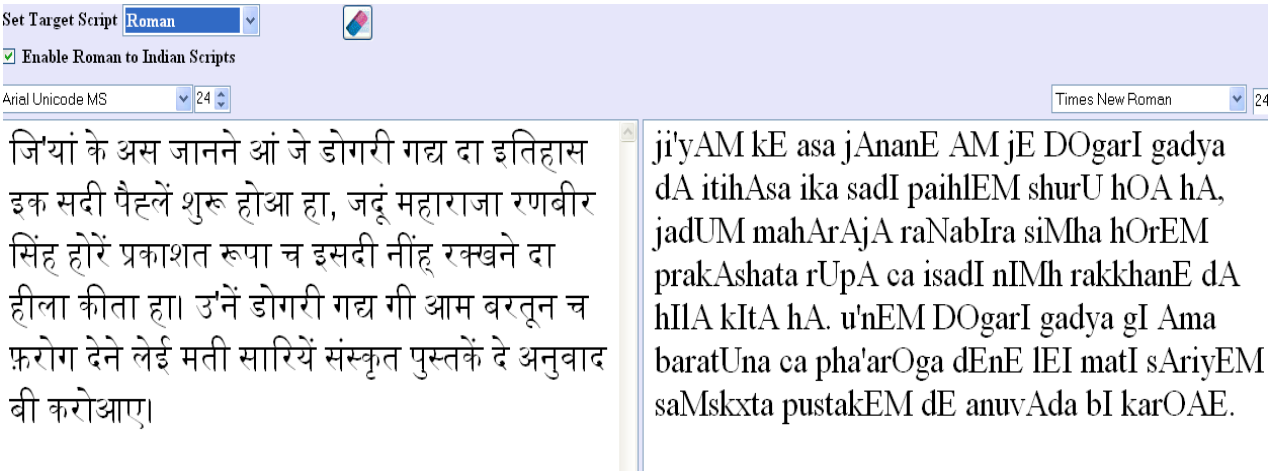

Fig.1*THE SNAPSHOT OF TRANSLITERATION INTERFACE*

Input tool interface:

A Graphical User Interface (GUI) based tool has been developed to facilitate the inputting of paradigms and the dictionary entries; a task which otherwise turned unwieldy given the XML format. Besides facilitating the inputting of paradigms and their features the tool also reduced the effort of manually lemmatizing the dictionary entries, as the entry automatically gets lemmatized according to the paradigm type selected for the word. The tool is divided into two parts, enabling the user to input the paradigms as well as inputting the dictionary entries, either manually, by typing word by word or by loading from an already existing word list. In this tool we can either input the word directly into the text box or load it from the existing word list and get its lemma easily, just by assigning an appropriate paradigm from the drop down list.

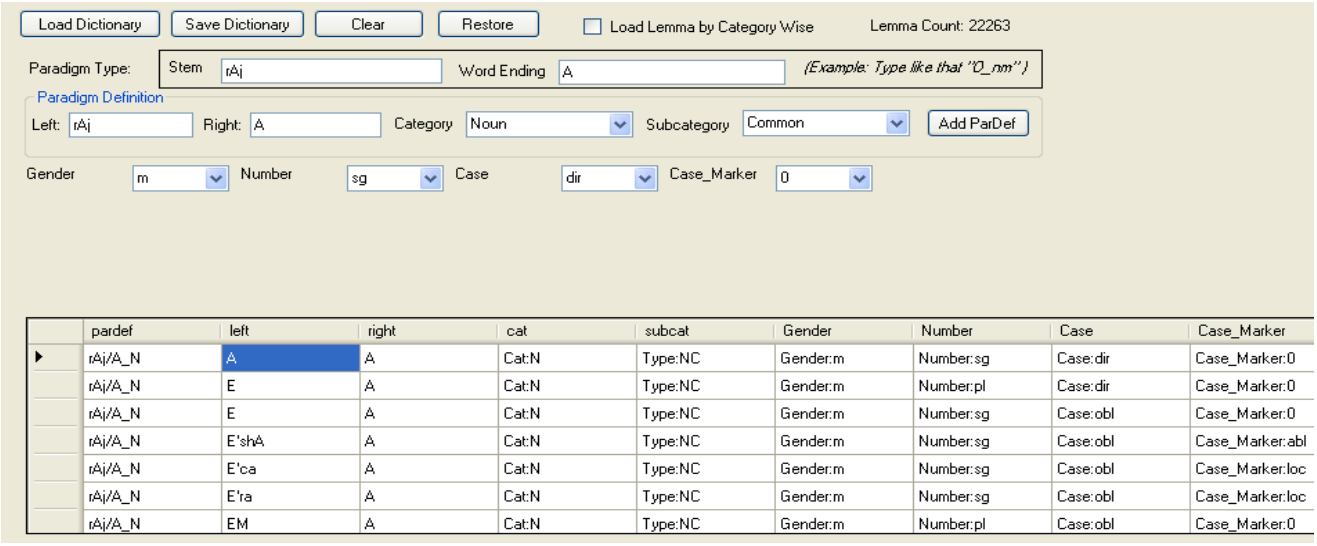

Fig.2*The snapshot of Input tool interface*

Computing Dogri morphology:

The process of computing morphology of Dogri, simply, involves the manipulation of strings or words forms (letter transducers), with or without much consideration to real

morphemic division. For illustration, we will show the implementation of inflectional morphology which can be divided into the following steps:

Step-1. Defining Elements:- All the characters and terms <sdefs> (Characters, Categories, Subcategories, Attributes and Attribute values) that are used in Transliteration scheme and Morphological analysis are defined in the XML code as shown below for Noun:

```
<?xml version="1.0"?
<dictionary>
<alphabet>abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ-
'</alphabet>
<sdefs>
<sdef n="cat:N" c="Noun"/>
<sdef n="subcat:NC" c="Common noun"/>
<sdef n="subcat:NP" c="Proper noun"/>
<sdef n="subcat:NV" c="Verbal noun"/>
<sdef n="subcat:NST" c="Spatio-Temporal noun"/>
<sdef n="gender:m" c="masculine"/>
<sdef n="gender:f" c="feminine"/>
<sdef n="number:sg" c="singular"/>
<sdef n="number:pl" c="plural"/>
<sdef n="case:d" c="direct"/>
<sdef n="case:o" c="oblique"/>
<sdef n="Case_Marker:abl" c="ablative"/>
<sdef n="Case_Marker:loc" c="loCative"/>
<sdef n="Case_Marker:voc" c="voCative"/>
</sdefs>
```

- **Paradigms**

A Paradigm in this model referred to a complete set of related inflectional and productive derivational word forms of a given category. This refers to the features and feature values of the root such as category, number, gender, person and case marking in the case of nouns and Gender, Number, Person, Tense, Aspect, Mood, Negation, Finiteness, Honorificity in the case of verb. To capture their each and every morphological variation they can be categorized on various paradigms based on their vowel ending, number, genderand case information. All the paradigms of a language are defined such that the words are divided into the left string and the right string separated by a slash; wherein, the left string is the unchangeable base or stem, where as the right and changeable string are the inflections. By choosing a word-form for defining a paradigm, we are actually naming/tagging that very paradigm by the name of the word-form chosen for example rAjA (King). The variable part of string on the right side of the splitter gets substituted while generating the inflectional paradigm (all possible word-forms in construction) for the same word as shown in the Dogri sample noun paradigm below, all the paradigms are defined within the <pardefs> XML tag.

```
<pardef n="rAj/A_N">
<e><p><l>A</l><r>A<s n="Cat:N"/><s n="subcat:NC"/><s n="Gender:m"/><s
n="Number:sg"/><s n="Case:dir"/><s n="Case_Marker:0"/></r></p></e>
<e><p><l>E</l><r>A<s n="Cat:N"/><s n="subcat:NC"/><s n="Gender:m"/><s
n="Number:pl"/><s n="Case:dir"/><s n="Case_Marker:0"/></r></p></e>
```

<e> <l>E</l><r>A<s n="Cat:N"/><s n="subcat:NC"/><s n="Gender:m"/><s n="Number:sg"/><s n="Case:obl"/><s n="Case_Marker:0"/></r> </e>
<e> <l>E'shA</l><r>A<s n="Cat:N"/><s n="subcat:NC"/><s n="Gender:m"/><s n="Number:sg"/><s n="Case:obl"/><s n="Case_Marker:abl"/></r> </e>
<e> <l>E'ca</l><r>A<s n="Cat:N"/><s n="subcat:NC"/><s n="Gender:m"/><s n="Number:sg"/><s n="Case:obl"/><s n="Case_Marker:loc"/></r> </e>
<e> <l>E'ra</l><r>A<s n="Cat:N"/><s n="subcat:NC"/><s n="Gender:m"/><s n="Number:sg"/><s n="Case:obl"/><s n="Case_Marker:loc"/></r> </e>
<e> <l>EM</l><r>A<s n="Cat:N"/><s n="subcat:NC"/><s n="Gender:m"/><s n="Number:pl"/><s n="Case:obl"/><s n="Case_Marker:0"/></r> </e>
<e> <l>EM'shA</l><r>A<s n="Cat:N"/><s n="subcat:NC"/><s n="Gender:m"/><s n="Number:pl"/><s n="Case:obl"/><s n="Case_Marker:abl"/></r> </e>
<e> <l>EM'ca</l><r>A<s n="Cat:N"/><s n="subcat:NC"/><s n="Gender:m"/><s n="Number:pl"/><s n="Case:obl"/><s n="Case_Marker:loc"/></r> </e>
<e> <l>EM'ra</l><r>A<s n="Cat:N"/><s n="subcat:NC"/><s n="Gender:m"/><s n="Number:pl"/><s n="Case:obl"/><s n="Case_Marker:loc"/></r> </e>
<e> <l>A</l><r>A<s n="Cat:N"/><s n="subcat:NC"/><s n="Gender:m"/><s n="Number:sg"/><s n="Case:obl"/><s n="Case_Marker:voc"/></r> </e>
<e> <l>EA</l><r>A<s n="Cat:N"/><s n="subcat:NC"/><s n="Gender:m"/><s n="Number:sg"/><s n="Case:obl"/><s n="Case_Marker:voc"/></r> </e>
<e> <l>EO</l><r>A<s n="Cat:N"/><s n="subcat:NC"/><s n="Gender:m"/><s n="Number:pl"/><s n="Case:obl"/><s n="Case_Marker:voc"/></r> </e>

- **Dictionary building**

From a conventional dictionary, root word dictionary is different in this model. The dictionary for Morphological Analysis contains roots and their corresponding paradigm. An entry in the dictionary contains that part of the lemma which is common to all the inflected forms of the word. This step involves lemmatization.The coverage of the system is directly related to the size of the dictionary. As this is an ongoing work, we hope to expand and make the system more robust, by increasing the dictionary size. As can be seen in the Dogri dictionary sample below, the left hand side element in the entry is the lemma and the right side element is the paradigm type that the word follows.

Dogri Sample Dictionary:

<e lm="rAj">rAj<par n="rAj/A_N"/></e>
<e lm="bhAsh">bhAsh<par n="bhAsh/A_N"/></e>
<e lm="jAgat">jAgat<par n="jAgat/a_N"/></e>
<e lm="bhain">bhain<par n="bhain/a_N"/></e>
<e lm="dhOb">dhOb<par n="dhOb/I_N"/></e>
<e lm="kuD'">kuD'<par n="kuD'/I_N"/></e>
<e lm="sAdh">sAdh<par n="sAdh/u_N"/></e>
<e lm="DAk">DAk<par n="DAk/U_N"/></e>
<e lm="ja">ja<par n="ja/u_N"/></e>
<e lm="maiMh">maiMh<par n="/maiMh_N"/></e>
<e lm="darEyA">darEyA<par n="darEyA/_N"/></e>
<e lm="gau"><par n="/gau_N"/></e>
<e lm="mA">mA<par n="mA/M_N"/></e>
<e lm="rAt">rAt<par n="rAt/a_N"/></e>

```
<e lm="katAb"><i>katAb</i><par n="katAb/a_N"/></e>
<e lm="laD'"><i>laD'</i><par n="laD'/I_N"/></e>
<e lm="nUMh"><i></i><par n="/nUMh_N"/></e>
<e lm="bhrA"><i></i><par n="/bhrA_N"/></e>
<e lm="muni"><i>muni</i><par n="muni/_N"/></e>
```

Conclusion:

The above discussion reveals that developing a morphological-analyzer for Dogri using LT-Toolbox is an efficient way given the elegance of the combinatory effect of word and paradigm model and the finite state transducers on speedy processing. Using finite state based model has one more advantage that it gives us two in one. I-e Further, using an open source tool in creation of computational resource for resource poor Dogri language is need of the hour, as it involves low cost and less time. Finally, for developing morphological analyzer with this approach one doesn't require any programming skill but a little working knowledge of computers and morphology. So, every one irrespective of having programming background can develop language specific resources and can contribute in the technological development of Dogri language.

Works Cited:

Agnihotri 2006, *Hindi: An Essential Grammer,*Routledge London and New York

Gupta, Veena.1995. *DogriVyakaran.* Academy of Art, Culture and Language, Jammu.

Kachru, Yamuna. 2006. *Hindi*. John Benjamins: Amsterdam/Philadelphia.

Kiraz, George Anton, *Computational Nonlinear Morphology, With Emphasis on Semitic Languages*, Cambridge University Press2001

Koul, Omkar N. 2008 *Modern Hindi Grammar*, Dunwoody Press

Masica, Colin. 1993. *The Indo-Aryan Languages*. CUP: Cambridge

McGregor, William B., *Linguistics, An Introduction*, Continuum International Publishing Group, 2009

Schmidt, Ruth Laila. 1999. *Urdu: An Essential Grammar*. Routledge: London

Sharma, Aryendra. 1994. *A Basic Grammar for Modern Hindi*. Central Hindi Directorate.

Sharma Atreyee, *Part -of -Speech Tagging,* Knowledge Sharing Event-4 LDCI-IL , CIIL, Mysore

Stump, G. T. (2001*).Inflectional Morphology.A Theory of Paradigm Structure*. Cambridge: Cambridge University Press.

Varshney, Radhey L, *An Introductory Textbook of Linguistics & Phonetics*, Students Store, 2006

Web Sources:

Pathania, Shashi. Dubey, Preeti and Devanand, Comparative Study of Hindi and Dogri Languages with Regard to Machine Translation, LANGUAGE IN INDIA Volume 11: 10 October 2011
http://www.languageinindia.com/oct2011/v11i10oct2011.pdf, Accessed on20 Oct. 2017

Rathor, Rajeev: Morphological POS Tagger for Hindi Language, Thapar University, Patiala, http://tudr.thapar.edu:8080/jspui/bitstream/10266/554/3/T554.pdf, Accessed on 20 Oct. 2017