Chapter 4

AutoParSe for Generation of Morphological Lexicon

4.1 Introduction

Morphological analysis is required for many NLP applications such as Spell Checkers, Text to Speech Systems, Rule Based Machine Translation, etc. Morphological Analyzer is a tool which is used to perform morphological analysis. Finite State Transducers are ideal for developing a Morphological Analyzer for a language because they are computationally efficient, inherently bidirectional and can also be used for word generation. Finite State Transducer based approach for developing Morphological Analyzer are based on word and paradigm model, wherein a word lemma is mapped to a corresponding Morphological Paradigm. A Morphological Paradigm is used to generate all possible word forms for a given word lemma. To develop a Morphological Analyzer using Finite State Transducer based approach, the following two resources are required:

- Morphological Paradigm List
- Morphological Lexicon
The Morphological Paradigm List is prepared using partial paradigms generated from SAM based Morphology Learning as an initial seed set which was enhanced referring to grammar books, morphology related linguistics thesis in Konkani and elaborate discussions with linguists. Lemmas\footnote{Citation form of words} in the language are then mapped to appropriate Morphological Paradigms to create a Morphological Lexicon. Mapping of lemmas to Morphological Paradigms is time consuming when done manually. Carrying out such mapping manually also requires manpower with necessary language skills which is difficult to get without adequate funding. Based on the morphological richness of the language and level of expertise and availability of the linguists, it takes about 3 to 6 months to map words to paradigms.

In this chapter, we present Semi-Supervised Algorithms for Automatic Paradigm Selection (AutoParSe) designed to facilitate the development of Morphological Lexicon. Here we propose the concept of paradigm differentiating measure (pdm) which has been used to map lemmas to Morphological Paradigms.

### 4.2 Related Work

Automatic mapping of word to a paradigm have been done earlier for other languages. An n-gram-based model has been developed \citep{Lindén2009, Sánchez-Cartagena2012} to select a single paradigm in cases where more than one paradigm generates the same set of word forms. These systems use POS information or some additional user input from native language speakers to map words to paradigms, instead of a corpus alone.

Akshar Bharati \citep{Bharati1995} group have worked extensively on the development of Morphological Analyzer for Indian Languages such as Hindi. Unsupervised
improvements to existing Morphological Analyzers (Bharati et al., 2001) have been implemented wherein equivalence classes of feature structures have been used to improve the coverage of existing Morphological Analyzer in Hindi.

Lexicon acquisition methods (Attia et al., 2012, Carlos et al., 2009, Clement et al., 2004, Forsberg et al., 2006) exist for many languages that extract lemmas from a corpus and map them to morphological paradigms. Functional Morphology has been used to define morphology for languages like Swedish and Finnish, and tools based on Functional Morphology, namely Extract (Forsberg et al., 2006) which suggest new words for a lexicon and map them to paradigms, have been developed. Functional Morphology based tools use constraint grammars to map words correctly to paradigms. To be able to use a tool like Extract, the morphology of the language has to be fitted into the Functional Morphology definition.

### 4.3 Terminology and Notations Used

#### Definition 4.3.1 (Root, Stem, Base, Prefix and Suffix)

A Root is the basic part of a lexeme\(^2\) which cannot be further analyzed, using either inflectional or derivational morphology. Root is that part of word-form that remains when all derivational and inflectional affixes have been removed.

A Stem is that part of the word form that remains when inflectional suffixes have been removed.

A Base \((b_i)\) is that part of the word form to which affixes of any kind can be added. It is a generic term which could refer to a Root or a Stem.

A Prefix is a bound morpheme that is attached at the beginning of a Base.

A Suffix \(s_i \in \Sigma^*\) is a bound morpheme that is attached at the end of a Base.

**Example 1:** In the word *untouchables*, the morphemes are \(un+touch+able+s\) wherein *touch* is the Root, *untouchable* is the Stem or the Base for the word *untouchables*,

---

\(^2\)Lexeme is the basic unit of meaning. It is an abstract unit of morphological analysis in linguistics, that roughly corresponds to a set of forms taken by a single word
touchable is the Base for the word untouchable, the morpheme 'un-' is a Prefix, the morpheme '-able' is a Derivational Suffix and the morpheme '-s' is an Inflectional Suffix.

**Example 2:** In the word touched, the morphemes are touch+ed wherein touch is the Root, and is also the Stem or the Base for the word touched and the morpheme '-ed' is an Inflectional Suffix.

**Example 3:** In the word wheelchairs, the morphemes are wheel+chair+s which has two Roots namely wheel and chair, wheelchair is the Stem or the Base for the word wheelchairs and the morpheme '-s' is an Inflectional Suffix.

**Example 4:** For the word went, the Root is go and is an example of irregular morphology which does not have a regular Stem or Suffix. Such a type of change in the Root is referred to as Suppletive Root Allomorphy.

**Definition 4.3.2 (Formative Suffix)**
A Formative Suffix is a Suffix between the Base and final Suffix. Formative Suffix is used to form a Stem in case of nouns and as a Phonetic Joiner Suffix in case of verbs.

**Example 1:** In the word फातराक (phataraaka) the morphemes are फातर+◌ा+क (phatara+a+ka) wherein फातर (phatara) is the Root, फातरा (phataraa) is the Stem, ‘-◌ा’ is the Formative Suffix, and the morpheme ‘-क(ka)’ is the Inflectional Suffix.

**Example 2:** In the word घोड्याक (ghoDyaaka) the morphemes are घोड+◌्यां+क (ghoD+yaa+ka) wherein घोड (ghoD) is the Root, घोड़ा (ghoDo) is the Base, घोंड़ा (ghoDyaa) is the Stem, ‘-◌्या(yaa)’ is the Formative Suffix, and ‘-क(ka)’ is the Inflectional Suffix.

**Definition 4.3.3 (Rule)**
An ordered 3-tuple (α, β, γ) is said to be a Rule used to convert a string $x_i$ to a string $y_i$ where α=”ADD/DELETE” is an operation performed on input string $x_i$; β=position at which the operation specified in α is to be performed on string $x_i$; γ=$z_i$ is the argument for the operation to be performed.

---

3 फातराक (phataraaka) (for the stone)
4 घोड्याक (ghoDyaaka) (for the horse)
Example: If $x_i=\text{धांवप}(dhaa.Nvapa)$\(^5\) and Rule= ("DELETE", "END", "प(pa)") where $\alpha=\text{"DELETE"}$; $\beta=\text{"END"}$; $\gamma=\text{"प(pa)"}$ with respect to Definition 4.3.3 $y_i=\text{धांव}(dhaa.Nva)$.

**Definition 4.3.4 (Base Formation Rule (BFR))**

An ordered n-tuple of Rules which is used to convert lemma $l_i$ to base $b_i$ is said to be a **Base Formation Rule** BFR.

Example: If $l_i=\text{भास}(bhasa)$\(^6\) and BFR= ("DELETE", "END", "स(sa)"),("ADD", "END", "श(sha)") with respect to Definition 4.3.4 $b_i=\text{भाश}(bhasha)$.

**Definition 4.3.5 (Inflectional Set)**

A set $W_{p_i,l_j}$ of all possible word forms generated by a Morphological Paradigm\(^7\) with $p_i$ as paradigm identifier, for a lemma $l_j$ is said to be the **Inflectional Set** for lemma $l_j$ with respect to paradigm $p_i$.

Example: If $p_i=P10$, a verb Morphological Paradigm and $l_j=\text{walk}$ with respect to Definition 4.3.5 $W_{p_i,l_j}=$\{walk, walks, walking, walked\}.

**Definition 4.3.6 (Suffix Evidence Set)**

A set $SES_{p_i,l_j}$ referred to as **Suffix Evidence Set** for lemma $l_j$ corresponding to paradigm $p_i$, is an intersection of set $W_{p_i,l_j}$ and set $C$

Example: If $W_{p_i,l_j}=$\{walk, walks, walking, walked\} and $C=$\{talk,talked,walk, walker, walking, walked\} with respect to Definition 4.3.6 $SES_{p_i,l_j}=$\{walk, walking, walked\}.

**Definition 4.3.7 (Suffix Evidence Value)**

A set $SEV_{p_i,l_j}$ referred to as **Suffix Evidence Value** for lemma $l_j$ corresponding to paradigm $p_i$, is a cardinality of the set $SES_{p_i,l_j}$

Example: If $W_{p_i,l_j}=$\{walk, walks, walking, walked\} and $C=$\{talk,talked,walk, walker, walking, walked\} with respect to Definition 4.3.6 $SES_{p_i,l_j}=$\{walk, walking, walked\} and with respect to Definition 4.3.7 $SEV_{p_i,l_j}=3$.

---

5\text{धांवप(dhaa.Nvapa) (to run)}

6\text{भास(bhasa) (language)}

7Morphological Paradigm has been defined in the following section
4.4 Morphological Lexicon Generation using AutoParSe

AutoParSe has been used to generate Morphological Lexicons for the two main inflecting Part of Speech (POS) categories in Konkani namely nouns and verbs. AutoParSe is a data driven method which makes use of Corpus to map lemmas in a language to corresponding Morphological Paradigms.

4.4.1 Morphological Paradigm

A Morphological Paradigm is used to generate the inflectional word forms for a given input lemma. We define a Morphological Paradigm as follows:

Definition 4.4.1 (Morphological Paradigm)

An ordered tuple \((\phi, \chi, \{(\psi_1, \omega_1, \gamma_1), \ldots, (\psi_n, \omega_n, \gamma_n)\})\) where

- \(\phi = p_i\), a unique identifier for the \(i^{th}\) paradigm,
- \(\chi = \text{pos}_i\), the Part of Speech category corresponding to the paradigm,
- \(\psi_j = BFR\) the Base Formation Rule corresponding to the \(j^{th}\) Base,
- \(\omega_j = S_k\) a set of (suffix, grammatical feature) ordered pairs corresponding to the \(j^{th}\) Base and
- \(\gamma_j\) = A boolean flag which is set to 1 if corresponding suffixes uniquely identify the paradigm i.e. corresponding \((\psi_j, \omega_j)\) forms the paradigm differentiating measure\(^9\).

- \(n\) is the total number distinct bases for the paradigm,

is said to be a Morphological Paradigm which is used to generate the Inflectional Set i.e. all the inflectional word forms, for the input lemma.

Example: When the paradigm is given by

Here suffix could be made up of more than one suffix concatenated with each other

\(^9\)paradigm differentiating measure has been defined in the following subsection.
• ϕ = P11,

• χ = noun,

• ψ₁ = ("DELETE", "END", "स(a)"),("ADD", "END", "श(sha)") is the BFR corresponding to the first Base.

• ω₁ = {((◌े)(e), singular oblique case), (◌ेक(eka), singular oblique accusative case), (◌ेकू च(ekaUch), singular oblique accusative case with emphatic clitic), ... }

• γ₁ = 1

• ψ₂ = ("DELETE", "END", "∅") is the BFR corresponding to the second Base.

• ω₂ = {((◌ो)(o), plural direct case), (◌ोच (och), plural direct case with emphatic clitic), ...

• γ₂ = 0

• n = 2,

If the input lemma = भास(bhaasa), then the first Base is भाश(bhaasha) and the second Base is भास(bhaasa). The word forms generated by the above paradigm are as follows: {भाशे(bhaashe), भाशेक(bhaasheka), भाशेकू च(bhaashekaUch), ... भासो(bhaaso), भासोच(bhaasoch), ...}

Types of Morphological Paradigms: Word forms generated by Morphological Paradigms can be represented at two levels namely Surface Level and Lexical Level. At the Surface Level, a Morphological Paradigm generates a set of word forms which can be expressed in an abstract manner as \{bᵢ . sⱼ : where bᵢ is the Base; sⱼ is the Suffix\}. At the Lexical Level, a Morphological Paradigm generates a set of word forms which can be expressed in an abstract manner as \{lᵢ + grammatical features : where lᵢ is the lemma\}. 
Example: If the input lemma $l_i = \text{dance}$, Word forms generated at Surface Level are \{dancing, danced, dances, \ldots\} where $b_i = \text{danc}$.

Word forms generated at Lexical Level are \{dance + present continuous, dance + past perfect, dance + present, \ldots\}.

Morphological Paradigms can differ from each other either at the Surface Level or at the Lexical Level

**Surface Level difference between Morphological Paradigms:** Two Morphological Paradigms are said to differ at surface level when they generate different set of word forms at the Surface Level for a given input lemma. Surface level difference implies that at least one of the following two conditions is true.

- \(\exists\) at least one BFR that is not the same amongst them.
- \(\exists\) at least one suffix which is not the same amongst them.

**Lexical Level difference between Morphological Paradigms:** Two distinct Morphological Paradigms are said to differ at lexical level when they generate same set of word forms at the Surface Level. Lexical level difference implies the following condition is true

- \(\exists\) at least one word form which has different grammatical features in the two paradigms.

Each Morphological Paradigm is unique either at the Surface or Lexical level. The feature which makes the Morphological Paradigm unique is referred to as *paradigm differentiating measure* and is defined as follows

**Definition 4.4.2 (Paradigm Differentiating Measure)**

The ordered tuple \((\psi_j, \omega_j)\) with respect to Definition 4.4.1 is called *paradigm differentiating measure* if it occurs only once across all possible paradigms.

**Example 1:** If set A and B represent two sets of word forms generated by two different paradigms $p_1$ and $p_2$ respectively which differ at the surface level, for a given lemma. Let set A and B be given as follows:
A= \{ (b_1.s_1,f_1), (b_1.s_2,f_2), (b_1.s_3,f_3), (b_1.s_4,f_4), (b_1.s_5,f_5) \}
B= \{ (b_1.s_1,f_1), (b_1.s_6,f_2), (b_1.s_3,f_3), (b_1.s_4,f_4), (b_1.s_5,f_5) \}

where \( b_j \) is a base obtained using \( \psi_j \), \( s_j \) is the suffix obtained using \( \omega_j \) and \( f_j \) is the corresponding grammatical feature.

From set A and B we observe that the word forms differ only at the second entry namely \((b_1.s_2,f_2) \in A\) and \((b_1.s_6,f_2) \in B\) hence the corresponding \((\psi_1, \omega_2)\) in \( p_1 \) and \((\psi_1, \omega_2)\) in \( p_2 \) are the paradigm differentiating measure.

**Example 2:** If set C and D represent two sets of word forms generated by two different paradigms \( p_1 \) and \( p_2 \) respectively which differ only at the lexical level, for a given lemma. Let set C and D be given as follows:

\[
\begin{align*}
C &= \{ (b_1.s_1,f_1), (b_1.s_1,f_2), (b_1.s_3,f_3), (b_1.s_4,f_4), (b_1.s_5,f_5) \} \\
D &= \{ (b_1.s_1,f_1), (b_1.s_3,f_2), (b_1.s_3,f_3), (b_1.s_4,f_4), (b_1.s_5,f_5) \}
\end{align*}
\]

where \( b_j \) is a base obtained using \( \psi_j \), \( s_j \) is the suffix obtained using \( \omega_j \) and \( f_j \) is the corresponding grammatical feature.

From set C and D we observe that the word forms are same at surface level but corresponding grammatical features differ only at the second entry namely \((b_1.s_1,f_2) \in A\) and \((b_1.s_3,f_2) \in B\) hence the corresponding \((\psi_1, \omega_2)\) in \( p_1 \) and \((\psi_1, \omega_2)\) in \( p_2 \) are the paradigm differentiating measure.

### 4.4.2 Verb Morphological Lexicon Generation using AutoParSe

AutoParSe has been used to generate a Verb Morphological Lexicon. A Verb Morphological Lexicon is a resource which maps Verb lemmas to corresponding Verb Morphological Paradigms. This resource is required to create a Finite State Transducer (FST) based Morphological Analyzer.

#### Problem Statement:
Given a set of verb lemmas in \( LX_V = \{l_i : i = 1 \text{ to } n\} \), where \( n \) is number of lemmas in Lexicon \( LX_V \); a set of unique corpus words \( W_C = \{(w_i,f_{w_i}) : i = 1 \text{ to } m\} \), where \( m \) is number of unique words in Corpus \( C \); a set of Verb Suffix Group \( VSG = \{(g_i,S_{g_i},CR_{g_i}) : g_i \text{ is the suffix group identifier, } i = 1 \text{ to } p\} \), where \( p \) is
number of Suffix Groups, \( S_g \) is the suffix set corresponding to the \( i^{th} \) Suffix Group and \( CR_g \) is the constraint rule if any corresponding to the \( i^{th} \) Suffix Group} and a set of Verb Paradigm List \( P_{LV} = \{(p_i, \{(BFR_j, j\cdot s_k, g_{jk})\}) : p_i \) is the paradigm identifier, \( BFR_j \) is BFR corresponding to the \( j^{th} \) Base, \( g_{jk} \) is the \( k^{th} \) suffix group corresponding to the \( j^{th} \) Base, \( j\cdot s_k \) is the Phonetic Joiner Suffix for the \( k^{th} \) suffix group corresponding to the \( j^{th} \) Base, \( 0 \leq j \leq n \) where \( n \) is the number of Bases corresponding to the paradigm and \( 0 \leq k \leq m \) where \( m \) is the number of Suffix Groups corresponding to the \( j^{th} \) Base of the paradigm} generate Verb Morphological Lexicon set \( LX_{VM} = \{(l_i, p_j) : l_i \in LX_V, \text{ and } p_j \in P_{LV}\} \)

### 4.4.2.1 Konkani Verb Inflections

In general, verb lemmas in Konkani appear in the gerundial form in lexical resources such as WordNet, dictionary or thesaurus. In their gerundial form, verbs end with the character प (pa) (Ending character of verb lemmas/ citation forms).

**Example:** चलप (chalapa) (to walk) and धांवप (dha.nvapa) (to run) are Konkani verb lemmas.

To generate an Inflectional Set corresponding to an input verb lemma we need to follow the following step

1. Obtain verb Base/ Bases\(^{10}\) from verb lemma using appropriate BFR

2. Attach appropriate suffixes to the verb Base/ Bases obtained in step 1

**Example:** If input verb lemma = धांवप (dha.nvapa), appropriate BFR = ("DELETE", "END", "प(pa)") verb Base = धांव (dha.Nva) appropriate suffixes = लो, तलो, ि◌ँल्लो, ता, ...(lo, talo, illo, taa, ...) Inflectional Set = {धांवलो, धांवतलो, धांविल्लो, धांवता, ...(dha.Nvalo, dha.Nvatalo, dha.Nvailllo, dha.Nvataa, ...} \)

\(^{10}\)A verb lemma can have more than one Base Forms
4.4.2.2 Paradigm Selection for Konkani Verbs

A Konkani verb lemma is mapped to exactly one Morphological Paradigm. The verb Morphological Paradigms are such that they all differ from each other at the surface level. However it is not possible to implement a Rule Based System to map verb lemmas to Morphological Paradigms due to ambiguity in paradigm selection. This ambiguity is presented next.

Ambiguity in Paradigm Selection for Konkani Verbs:

To select a Morphological Paradigm for an input verb lemma, we first need to identify which BFR is compatible to the given input lemma. BFR are based on ending characters in the verb lemma. However there is no one-one correspondence between the verb lemma ending characters and BFR. This gives rise to ambiguity in choosing the paradigm with an appropriate BFR.

Example: If verb lemma ends with characters वप (vapa) as in case of the verb lemma धांवप (dha.nvapa) (to run), then three BFR are compatible which gives rise to three possible Bases namely धांव, धांय, धां (dha.Nva, dha.Nya, dha.N). Amongst these three possible Bases only धांव (dha.Nva) is the correct choice. However no linguistic rule can be used to arrive at the correct base धांव (dha.Nva), thus causing ambiguity in choosing a correct BFR for the input verb lemma.

4.4.2.3 Framework for Verb Morphological Lexicon Generation

The Framework used for Verb Morphological Lexicon Generation is illustrated in Figure 4.1. It takes as input a Verb Lexicon ($L_{V}$) resource, uses three resources namely Verb Suffix Group ($VSG$), Verb Paradigm List ($P_{L}V$) and Corpus ($C$) and generate an output resource namely Verb Morphological Lexicon ($L_{VM}V$). It consists of three modules namely Inflectional Set Generator, Candidate Paradigm Generator and Paradigm Selector. The details of resources used and modules are described next.
4.4.2.4 Resources in Verb Morphological Lexicon Generation

The resources used in Verb Morphological Lexicon Generation Framework are as follows:

- **Verb Lexicon** \((LX_V)\): Verb Lexicon resource comprises of verb lemmas. This resource has been extracted from the Konkani WordNet. An entry in the \((LX_V)\) resource is of the form \(l_i\). The following example illustrates an entry in this resource.

  **Example:** Verb Lexicon entry: धावप

- **Verb Suffix Group** \((VSG)\): Verb Suffix Group resource comprises of Suffix Groups. Suffix groups are created such that if one suffix of the group can be attached to a Base all other members of the same group can also be attached to the same Base.
An entry in the \( VSG \) is of the form \((g_i, S_{g_i}, CR_{g_i})\) where

- \( g_i \) is a group identifier,
- \( S_{g_i} \) is the set of verb suffixes belonging to the group and
- \( CR_{g_i} \) is constraint rule if any corresponding to the group.

The following example illustrates an entry in this resource.

**Example 1:** Verb Suffix Group entry:

\[
(g_i, S_{g_i}, CR_{g_i}) = (G-1, \{\text{तना(tanaa), ताना(taanaa), तली(taII), तली(taLI.N), ... }\}, \emptyset)
\]

- G-1 is the group identifier,
- All suffixes in \( S_{g_i} \) are regular verb suffixes which begin with the letter \( त(ta) \) and
- \( CR_{g_i} = \emptyset \) implies there is no constraint defined for this group.

**Example 2:** Verb Suffix Group entry:

\[
(g_i, S_{g_i}, CR_{g_i}) = (G-2, \{\text{◌्ठना(Thanaa), ◌्ठाना(Thaanaa), ◌्ठली(ThaII), ◌्ठली(ThaLI.N), ... }\}, \text{BaseEnd=”ठ(Tha)”})
\]

- G-2 is the group identifier,
- All suffix in this group are morphophonemic variation of regular verb suffixes which begin with the letter \( ठ(Tha) \) and
- \( CR_{g_i} = \text{BaseEnd=”ठ(Tha)”} \) implies this suffix group can be attached to those lemmas whose Base ends with the letter \( ठ(Tha) \).

- **Verb Paradigm List \((P_L V)\):** Verb Paradigm List resource comprises of Verb Morphological Paradigms. This resource is used to define the Verb Paradigms and makes use of the \((VSG)\) resource to do so.

An entry in the \((P_L V)\) is of the form \((p_i, \{(BFR_j, j_{sk}, g_{jk})\})\) where
- $p_i$ is the paradigm identifier, $0 \leq j \leq r$ where $r$ is the number of verb paradigm,

- $BFR_j$ is BFR corresponding to the $j^{th}$ Base, $0 \leq j \leq n$ where $n$ is the number of Bases corresponding to the paradigm,

- $g_{jk}$ is the $k^{th}$ suffix group corresponding to the $j^{th}$ Base, and $0 \leq k \leq m$ where $m$ is the number of Suffix Groups corresponding to the $j^{th}$ Base of the paradigm,

- $j s_{jk}$ is the Phonetic Joiner Suffix for the $k^{th}$ suffix group corresponding to the $j^{th}$ Base.

**Example:** Verb Paradigm List entry:

$$(p_i, \{(BFR_j, j s_k, g_{jk})\}) = (P-1, \{("DELETE", "END", "प(pha)"), \emptyset, G-1), ...\})$$

where

- $p_i$=P-1 is the paradigm identifier,

- $BFR_j= ("DELETE", "END", "प(pha)"),$

- $j s_k=\emptyset$ and

- $g_{jk}=G-1$ is the corresponding Verb Suffix Group identifier.

- **Corpus C:** Corpus resource comprises of words found in a Corpus of the language. An entry in the Corpus resource is of the form $(w_i,f_{w_i})$. The following example illustrates an entry in the resource.

**Example:** Corpus entry: (फातर$^{11}$, 1164)

The output resource generated by Verb Morphological Lexicon Generation Framework is as follows:

$^{11}$फातर(phaatara) (stone)
• **Verb Morphological Lexicon** ($LX_{VM}$): Verb Morphological Lexicon resource comprises of verb lemmas $l_i$ along with the corresponding paradigm identifier $p_j$. An entry in the $LX_{VM}$ resource is of the form $(l_i,p_j)$. The following example illustrates an entry in this resource.

**Example:** Verb Morphological Lexicon entry: (धांवप, P-13)

### 4.4.2.5 Modules in Verb Morphological Lexicon Generation

The modules in Verb Morphological Lexicon Generation Framework are as follows:

#### Candidate Paradigm Generator

This module is used to generate an intermediate resource namely Candidate Paradigm List ($CP$), which is a list of paradigms which are compatible with the input lemma.

The algorithm for the Candidate Paradigm Generator is illustrated in Figure 4.2

**Algorithm: Candidate Paradigm Generator for Verb**

**Input:** Verb Lemma $l_i$, Set of verb Morphological Paradigms $P_{LV}$, Set of Verb Suffix Group ($VSG$).

**Output:** Candidate Paradigm List $CP$.

/* Compute compatible candidate paradigms */

$CP = \emptyset$ // Candidate Paradigm set initialized to NULL

For each $p_i$ in $P_{LV}$

- BFR= getBaseFormationRule($p_i$)
- If isBFRCompatible($l_i$,BFR) // Compatible with Base Formation Rule
  - If isMorphoPhonemicCompatible($l_i$, $p_i$) // Does not violate morphophonemic rule
    - Add $p_i$ to $CP$
  - End If
- End If

End For

**Figure 4.2:** Algorithm: Candidate Paradigm Generator for Verb.

A paradigm is said to be compatible with an input lemma if it satisfies the following conditions:

- The input lemma $l_i$ is compatible with all the $BFR$ in the corresponding paradigm i.e. for a Rule in BFR, when $\alpha=$"DELETE", $\beta=$"END" and $\gamma \neq \emptyset$, the Rule...
is compatible with input lemma only when ending characters in lemma match string specified in $\gamma$.

**Example:** If the input lemma $l_i = \text{तापप}(taapapa)(\text{to scold})$, a paradigm whose $BFR$ has Rule ("DELETE", "END", "वप(vapa)") will not be compatible with input lemma $l_i$ as its ending characters do not match $\gamma$ which is "वप(vapa)"

- The input lemma $l_i$ does not violate morphophonemic rules when attaching a Suffix to the corresponding Base i.e. if the language has a morphophonemic rule which states that if the Base ends with a particular character the regular suffix undergoes a transformation.

**Example:** If the input lemma $l_i = \text{उठप}(uThapa)(\text{to get up})$, since morphophonemic rule in Konkani states that if the Base ends with the character ठ(Tha) the suffixes starting with त(ta) have to be transformed to ◌्ठ(Tha), those paradigms with suffix group which start with त(ta) will not be compatible.

**Inflectional Set Generator**

This module is used to generate the Inflectional Set intermediate resource for a given input lemma $l_i$ corresponding to a Morphological Paradigm with paradigm identifier $p_i$. It makes use of a Finite State Machine based sequencing of suffixes to generate the different possible suffixes. This suffixes are grouped together into suffix groups are stored in the $VSG$ resource. Corresponding to each suffix group there is a joiner suffix ($j_sj_k$). The inflectional word forms are obtained by attaching the joiner suffix ($j_sj_k$) to the verb Base, followed by the suffixes in the corresponding suffix group. The algorithm for the Inflection Set Generator is illustrated in Figure 4.3

**Paradigm Selector**

This module is used to select the relevant Morphological Paradigm for the input lemma $l_i$ from amongst the Candidate Paradigms in $CP$. The following steps are followed to select a relevant Morphological Paradigm
Chapter 4. AutoParSe for Generation of Morphological Lexicon

Algorithm: Inflectional Set Generator for Verb

**Input:** Paradigm identifier \( p_i \), Verb Lemma \( l_j \), Set of verb Morphological Paradigms \( P_{LV} \), Set of Verb Suffix Group (VSG).

**Output:** Inflectional Set \( W_{p_i l_j} \).

/* Compute Inflectional Set for lemma \( l_j \) corresponding to paradigm \( p_i \) */

\[
W_{p_i l_j} = \emptyset \\
// Inflectional Set is initialized to NULL
\]

BFR-List = getBaseFormationRules(\( p_i \))

For each \( BFR_j \) in BFR-List

\[
b_j = \text{applyBFR}(l_i, BFR_j) \quad // \text{b}_j \text{ is the } j^{th} \text{ Base corresponding to the paradigm}
\]

\[
g_jk = \text{getSuffixGroup}(p_i, BFR_j) \quad // \text{get suffix group identifier corresponding to the}
\]

\[
S_{jk} = \text{getSuffixSet}(g_jk) \quad // \text{get suffix set corresponding to the } j^{th} \text{ Base and } k^{th} \text{ Suffix Group}
\]

\[
js_{jk} = \text{getJoinerSuffix}(p_i, BFR_j, g_jk) \quad // \text{get joiner suffix set}
\]

For each \( s_{jk} \) in \( S_{jk} \) // for each suffix in the suffix set

\[
w_{jk} = b_j.js_{jk}.s_{jk}
\]

\[
W_{p_i l_j} = W_{p_i l_j} \cup \{w_{jk}\}
\]

End For

End For

---

**Figure 4.3:** Algorithm: Inflectional Set Generator for Verb.

1. For each candidate paradigm in \( CP \), calculate the Suffix Evidence Set \( SES \) and Suffix Evidence Value \( SEV \)

2. Select that paradigm for which the \( SEV \) is maximum as relevant paradigm for the input lemma \( l_i \).

The algorithm for the Paradigm Selector module is illustrated in Figure 4.4

### 4.4.2.6 Algorithm: Verb Morphological Lexicon Generation

This Automatic Paradigm Selection (AutoParSe) algorithm is used to generate a Verb Morphological Lexicon. The algorithm for Verb Morphological Lexicon Generation is illustrated in Figure 4.5

The input to the algorithm is a Verb Lexicon \( LX_V \) consisting of verb lemmas. For each verb lemma in \( LX_V \) the relevant Morphological Paradigm is selected using the following two steps:
Algorithm: Paradigm Selector for Verb

**Input:** Verb lemma \(l_i\), Candidate Paradigm set \(CP\), set of unique corpus words \(W_C\)

**Output:** Relevant paradigm identifier \(R_{p_i}\).

/* Select Relevant Paradigm*/

maxSupport=0  // Initialize support to zero

\(R_{p_i} = \emptyset\)

For each \(p_i \in CP\)

\[W_{p_i,l_i} = \text{generateInflectionalSet}(l_i,p_i)\]

\[SES_{p_i,l_i} = W_{p_i,l_i} \cap W_C\]

\[SEV_{p_i,l_i} = |SES_{p_i,l_i}|\]

support = \(SEV_{p_i,l_i}\)

If support > maxSupport

\(R_{p_i} = p_i\)

maxSupport = support

End If

End For

**Figure 4.4:** Algorithm: Paradigm Selector for Verb.

Algorithm: Verb Morphological Lexicon Generation

**Input:** Set of Verb Lemmas \((LX_V)\), Set of unique corpus words \((W_C)\), Set of Verb Suffix Group \((VSG)\) and Set of Verb Morphological Paradigms \(P_LV\).

**Output:** Verb Morphological Lexicon set \(LX_{VM}\).

\(LX_{VM} = \emptyset\)

For each lemma \(l_i \in LX_V\)

/* Compute compatible candidate paradigms */

\(CP = \text{generateCandidateParadigm}(l_i, P_LV, VSG)\)

/* Select Relevant Paradigm */

\(R_{p_i} = \text{paradigmSelector}(l_i, CP, W_C)\)

\(LX_{VM} = LX_{VM} \cup (l_i, R_{p_i})\)

End For

**Figure 4.5:** Algorithm: Verb Morphological Lexicon Generation.

1. First compute the Candidate Paradigm List \((CP)\) using the Candidate Paradigm Generator module for the input lemma \(l_i\). This module selects relevant candidate paradigms from amongst a list of paradigms by pruning those paradigms which are not compatible with the input lemma \(l_i\).

2. Next, the relevant paradigm from amongst the candidate paradigm is selected using the Paradigm Selector module. It involves the following steps:
(a) It makes use of the Inflectional Set Generator module in order to generate the Inflectional Set.

(b) It generates Suffix Evidence Set $SES_{p_i,l_j}$ for each verb lemma $l_j$ corresponding to paradigm $p_i$ using Inflectional Set generated in previous step.

(c) It selects that paradigm which maximizes the cardinality of $SES_{p_i,l_j}$ as relevant paradigm.

(d) It adds the (lemma, relevant paradigm identifier) pair to the Verb Morphological Lexicon ($LX_{VM}$).

### 4.4.3 Noun Morphological Lexicon Generation using AutoParse

The Framework that was used for Verb Morphological Lexicon Generation could not be directly applied to nouns because paradigm selection for Konkani nouns differs from paradigm selection for verbs. The differences are as listed below:

- Unlike verb lemmas, a noun lemma in Konkani can map to more than one Morphological Paradigm.

**Example:** The noun lemma मान (maana) has two senses in konkani which could mean either *respect* or *neck* and in each of the two senses the lemma maps to different paradigms.

- Noun paradigms can differ from each other either at the surface level or at the lexical level.

**Example:** The noun lemmas फातर (phaatara) (*stone*) and पान (paana) (*leaf*) map to paradigms which are same at the surface level but differ at lexical level i.e. the word form $l_i + \text{a}$ is generated for both paradigms, when $l_i = \text{फातर (phaatara)}$ the word form $l_i + \text{a}=\text{फातरा (phaataraa)}$ has corresponding grammatical feature as *singular oblique form* however when $l_i = \text{पान (paana)}$ the word form $l_i + \text{a}=\text{पाना (paanaa)}$ has corresponding grammatical feature as *singular oblique form or singular direct plural form*. 
As a result of the above differences the paradigm selection algorithm AutoParSe had to be enhanced to apply for nouns. The proposed enhancements to AutoParSe were as follows:

- We proposed the concept of paradigm differentiating measure to be able to map noun lemmas to more than one paradigm.
- We used a two staged paradigm selection approach wherein in the first stage the noun lemma was mapped to the Surface Level Morphological Paradigm making use of proposed concept paradigm differentiating measure. In the second stage, the lemmas were mapped to appropriate Lexical Level Morphological Paradigm using Logistic Regression

The details of the enhancements to AutoParSe and its application to noun are presented in this section.

**Problem Statement:** Given a set of noun lemmas in $LX_N = \{l_i : i = 1 \text{ to } n\}$, where $n$ is number of lemmas in Lexicon $LX_N$; a set of unique corpus words $W_C = \{(w_i,f_{w_i}) : i = 1 \text{ to } m\}$, where $m$ is number of unique words in Corpus $C$; a set of Noun Suffix Group $NSG = \{(g_i,S_i) : g_i \text{ is the group identifier, } i = 1 \text{ to } p\}$, where $p$ is number of Suffix Groups, $S_i$ is the suffix set corresponding to the $i^{th}$ Suffix Group; a set of Surface Noun Paradigm List $P_{LN_S} = \{(p_i, \{(BFR_j, s_l, g_l, pdm_l)\}) : p_i \text{ is the paradigm identifier, } BFR_j \text{ is the Base Formation Rule, } s_l \text{ is the stem formative suffix corresponding to the } l^{th} \text{ suffix group, } g_l \text{ is the group identifier corresponding to the } l^{th} \text{ suffix group, } pdm_l \text{ is the paradigm differentiating measure flag corresponding to the } l^{th} \text{ stem formative suffix, } i = 1 \text{ to } q, \text{ where } q \text{ is number of noun paradigms in } L, j = 1 \text{ to } r, \text{ where } r \text{ is number of Bases corresponding to the } i^{th} \text{ noun paradigm and } l = 1 \text{ to } s, \text{ where } s \text{ is number of Noun Suffix Groups corresponding to the } j^{th} \text{ Base of } i_{th} \text{ noun paradigms}\}$ and a set of Lexical Noun Paradigm List $P_{LN_L} = \{(p_j, p_i) : p_j \text{ is the lexical level paradigm identifier and } p_i \text{ is the corresponding paradigm identifier in } P_{LN_S}\}$ generate Noun Morphological Lexicon set $LX_{NM} = \{(l_i,p_j) : l_i \in LX_N, \text{ and } p_j \in P_{LN_S} \cup P_{LN_L}\}$
4.4.3.1 Konkani Noun Inflections

In general noun lemmas in Konkani appear in the singular direct nominative form in lexical resources such as WordNet, dictionary or thesaurus.

To generate an Inflectional Set corresponding to an input noun lemma we need to follow the following step

1. Obtain noun Base from noun lemma using appropriate BFR
2. Attach appropriate formative suffix to the noun Base to obtained stem
3. Attach appropriate suffixes\[12\] to the noun stem obtained in step 2

Example: If input noun lemma = पान (paana), appropriate BFR = ("ADD", "END", "∅") noun Base = पान (paana) Formative Suffix = ऑ(a) appropriate suffixes = क, कू, ंत, ...(ka, kucha, .Nta, ...) Inflectional Set = {पानाक, पानाकू, पानांत, ...(paanaaka, paanaakucha, paanaa.Nta, ...) }

4.4.3.2 Paradigm Selection for Konkani Nouns

A Konkani noun lemma can be mapped to more than one Morphological Paradigm. The noun Morphological Paradigms are such that they all differ from each other either at the surface level or at the lexical level. It is not possible to implement a Rule Based System to map noun lemmas to Morphological Paradigms due to ambiguity in paradigm selection presented next.

Ambiguity in Paradigm Selection for Konkani Nouns

Ambiguity in Paradigm Selection for Konkani Nouns exists due to the following reasons

\[12\]Case Marker Suffixes
• **Formative Suffix attachment:** There is no known linguistic rule\(^{13}\) to decide which Formative Suffix is to be attached to the Base to obtain the Inflectional Set. This gives rise to ambiguity in choosing the appropriate paradigm.

**Example:** When noun lemma does not end with a vowel as in case of the noun lemma पाल (paala) (*lizard*); then three possible formative suffixes could be attached which gives rise to three possible Stems namely पाला, पाली, पाले (paalaa, paalI, paale). Amongst these three possible Stems only पाली (paalI) is the correct choice. However no linguistic rule can be used to arrive at the correct stem thus causing an ambiguity in choosing a correct paradigm for the input noun lemma.

• **Multiple paradigm for single noun lemma:** A single noun lemma could be mapped to more than one noun paradigm. This gives rise to another ambiguity is paradigm selection.

**Example:** For noun lemma मराठी (maraThI) (*marathi language or marathi speaking person*); the same lemma will map to two different paradigms for the two different senses namely marathi language and marathi speaking person. In such a case simply computing Suffix Evidence Value SEV is not enough to resolve ambiguity.

• **Lexical level differences in paradigms:** Some paradigm differ only at lexical level and generate the same Inflectional Set at surface level. This is another ambiguity challenge faced for paradigm selection.

**Example:** For noun lemma पान (paana) (*leaf*); the same lemma will map to two different paradigms which are same at the surface level. This is because a single form in such paradigm have two different grammatical features as in case of पाना (paanaa) which could be singular oblique form or direct plural form which is a type of ambiguity.

---

\(^{13}\) Linguistic rule based on noun lemma ending characters alone in absence of knowledge of nouns grammatical gender
4.4.3.3 Framework for Noun Morphological Lexicon Generation

The Framework used for Noun Morphological Lexicon Generation is illustrated in Figure 4.6. It takes as input a Noun Lexicon ($LX_N$) resource, uses six resources namely Noun Suffix Group ($NSG$), Surface Noun Paradigm List ($PLN_S$), Lexical Noun Paradigm List ($PLNL$), Lexical Training Data Set ($TDS$), Derivational cum Oblique Suffix List ($DOS$) and Corpus ($C$) and generate an output resource namely Noun Morphological Lexicon ($LX_{NM}$). It consists of five modules namely Candidate Surface Paradigm Generator, Inflectional Set Generator, Surface Paradigm Relevance Checker, Paradigm Pruner and Lexical Paradigm Relevance Checker. The details of resources used and modules are described next.
4.4.3.4 Resources in Noun Morphological Lexicon Generation

The resources used in Noun Morphological Lexicon Generation Framework are as follows:

- **Noun Lexicon** ($LX_N$): Noun Lexicon resource comprises of noun lemmas. This resource has been extracted from the Konkani WordNet. An entry in the ($LX_N$) resource is of the form $l_i$. The following example illustrates an entry in this resource.

  Example: Noun Lexicon entry: पान (paana) (leaf)

- **Noun Suffix Group** ($NSG$): Noun Suffix Group resource comprises of Suffix Groups namely Singular Oblique Case Marker suffixes and Plural Oblique Case Marker suffixes. Suffix groups are created such that if one suffix of the group can be attached to the noun Stem all other members of the same group can also be attached to the same Noun Stem.

  An entry in the $NSG$ is of the form $(g_i, S_{g_i})$ where

  - $g_i$ is a group identifier,
  - $S_{g_i}$ is the set of noun suffixes belonging to the group.

  The following example illustrates an entry in this resource.

  Example: Noun Suffix Group entry:

  $(g_i, S_{g_i}) = (G-11, \{क(ka), न(na), अंत(.Nta), \ldots \})$ where

  - G-11 is the group identifier,
  - all suffixes in $S_{g_i}$ are noun singular oblique case marker suffixes.

- **Surface Noun Paradigm List** ($PL_N$): Surface Noun Paradigm List resource comprises of Noun Morphological Paradigms which differ from each other at the surface level. In cases where two Morphological Paradigms differ only at the
lexical level, the two paradigms are represented by a single paradigm in this resource. This resource makes use of the \((NSG)\) resource.

An entry in the \((PLNS)\) is of the form \((pi, \{BFRj, sl, gl, pdm_l\})\) where

- \(pi\) is the paradigm identifier, \(i = 1\) to \(q\), where \(q\) is number of noun paradigms in Language \(L\),
- \(BFRj\) is the Base Formation Rule, \(j = 1\) to \(r\), where \(r\) is number of Bases corresponding to the \(i^{th}\) noun paradigm,
- \(sl\) is the stem formative suffix corresponding to the \(l^{th}\) suffix group,
- \(gl\) is the group identifier corresponding to the \(l^{th}\) suffix group, \(l = 1\) to \(s\), where \(s\) is number of Noun Suffix Groups corresponding to the \(j^{th}\) Base of \(i^{th}\) noun paradigms and
- \(pdm_l\) is the paradigm differentiating measure boolean flag corresponding to the \(l^{th}\) stem formative suffix, which is set to 1 if the corresponding suffix is a part of paradigm differentiating measure which is used to distinguish the paradigm.

**Example:** Surface Noun Paradigm List entry:

\((pi, \{BFRj, sl, gl, pdm_l\}) = (P-11, \{("DELETE", "END", "\emptyset"), \odot I(a), G-11,1), ("DELETE", "END", "\emptyset"), \odot I(a.N), G-12,0), ...\})\) where

- \(pi=\)P-1 is the paradigm identifier,
- \(BFRj= ("DELETE", "END", "\emptyset")\) when \(j=1\),
- \(sl= \odot I(a)\) when \(l=1\) and \(sl= \odot I(a.N)\) when \(l=2\),
- \(gl= G-11\) when \(l=1\) and \(gl= G-12\) when \(l=2\),
- \(pdm_l= 1\) when \(l=1\) and \(pdm_l= 0\) when \(l=2\).

- **Lexical Noun Paradigm List \((PLNL)\):** Lexical Noun Paradigm List resource comprises of Noun Morphological Paradigms corresponding to those paradigms in \(PLNS\) which can be split into paradigms at the lexical level.
An entry in the \((P_LN_L)\) is of the form \((p_i, p_j)\) where

- \(p_i\) is the paradigm identifier at lexical level, \(i = 1\) to \(q\), where \(q\) is number of lexical noun paradigms corresponding to the surface level paradigm,
- \(p_j\) is the corresponding surface level paradigm identifier

**Example:** Lexical Noun Paradigm List entry:

\((p_i, p_j) = (P-11a, P-11)\) where

- \(p_i=P-11a\) is the paradigm identifier of the lexical level paradigm,
- \(p_j=P-11\) is the paradigm identifier of the surface level paradigm,

- **Lexical Training Data Set \((TDS)\):** Lexical Training Data Set resource comprises of features\(^{14}\) used as a training set to select Lexical Noun Paradigm. For each Lexical Noun Paradigm \(TDS\) has 100 entries.

- **Derivational cum Oblique Suffix List \((DOS)\):** Derivational cum Oblique Suffix List resource comprises of Oblique Formative Suffixes which can also act as a derviational suffix or ending characters in parallel word forms.

  **Example:** Derivational cum Oblique Suffix List for Konkani: \(\{ \text{◌ी}(I), \text{◌े}(e) \} \)

- **Corpus \(C\):** Corpus resource comprises of words found in a Corpus of the language. An entry in the Corpus resource is of the form \((w_i, f_{w_i})\). The following example illustrate an entry in the resource.

  **Example:** Corpus entry: (फातर\(^{15}\), 1164)

The output resource generated by Noun Morphological Lexicon Generation Framework is as follows:

\(^{14}\)Features used are listed in 4.1
\(^{15}\)फातर\((phaatara)\) \((stone)\)
• **Noun Morphological Lexicon** ($L_{NM}$): Noun Morphological Lexicon resource comprises of noun lemmas $l_i$ along with the corresponding paradigm identifier $p_j$. An entry in the $L_{NM}$ resource is of the form $(l_i, p_j)$. The following example illustrates an entry in this resource.

**Example:** Noun Morphological Lexicon entry: (पान, P-11a)

### 4.4.3.5 Modules in Noun Morphological Lexicon Generation

The modules in Noun Morphological Lexicon Generation Framework are as follows:

**Candidate Surface Paradigm Generator**

This module is used to select Surface Noun Paradigm compatible with the input noun lemma, from amongst those in $P_LN_S$ to generate an intermediate resource namely Candidate Paradigm List ($CP$). The algorithm for the Candidate Surface Paradigm Generator is illustrated in Figure 4.7.

**Algorithm: Candidate Surface Paradigm Generator for Noun**

**Input:** Noun Lemma $l_i$, Set of Surface Noun Morphological Paradigms $P_LN_S$, Set of Noun Suffix Group ($NSG$).

**Output:** Candidate Paradigm List $CP$.

/* Compute compatible candidate paradigms */

$CP = \emptyset$ // Candidate Paradigm set initialized to NULL

For each $p_i$ in $P_LN_S$

    BFR = getBaseFormationRule($p_i$)

    /* Compatible with Base Formation Rule */

    If isBFRCompatible($l_i$, BFR)

        $b_i$ = applyBFR($l_i$, BFR) // get corresponding Base

        /* Does not cause base end vowel and suffix begin vowel condition */

        If isBaseCompatible($b_i$, $p_i$)

            $CP = CP \cup \{p_i\}$

        End If

    End If

End For

**Figure 4.7:** Algorithm: Candidate Surface Paradigm Generator for Noun.
A paradigm is said to be compatible with an input lemma if it satisfies the following conditions:

- The input lemma $l_i$ is compatible with all the BFR in the corresponding paradigm i.e. For a Rule in BFR, when $\alpha =$ "DELETE", $\beta =$ "END" and $\gamma \neq \emptyset$, the Rule is compatible with input lemma only when ending characters in lemma match string specified in $\gamma$.

**Example:** If the input lemma $l_i = घोड़ो (ghoDo)(horse)$, a paradigm whose BFR has Rule ("DELETE", "END", "◌ी (I)") will not be compatible with input lemma $l_i$ as its ending characters do not match $\gamma$ which is "◌ी (I)"

- The paradigm does not cause two vowels to be adjacent when attaching a Suffix to the corresponding Base i.e. If the BFR of a paradigm does not cause a change to the input lemma and $l_i = b_i$ for the paradigm and base ends with vowel and suffix begins with vowel.

**Example:** If the input lemma $l_i = िहरो (hiro)(hero)$, $l_i$ will be compatible with those paradigms where execution of BFR causes no change to $l_i$ i.e. $l_i = b_i$, however if the corresponding formative suffix in the paradigm begins with a vowel as in िहरोा (hiroa) which is invalid, such paradigms which cause this condition are pruned.

**Inflectional Set Generator**

This module is used to generate the Inflectional Set intermediate resource for a given input lemma $l_i$ corresponding to a Surface Paradigm with paradigm identifier $p_i$. The algorithm for the Candidate Paradigm Generator is illustrated in Figure 4.8

The inflectional Set generated will consist all Inflectional forms when the paradigm differentiating measure (pdm) boolean flag passed to the module is set to 0. When the paradigm differentiating measure boolean flag is set to 1, it generates only those word forms which differentiate the current paradigm from other paradigms at the surface.
Algorithm: Inflectional Set Generator for Noun

Input: Paradigm identifier $p_i$, Verb Lemma $l_j$, Set of Surface Noun Morphological Paradigms $P_{N} \cup S$, Set of Noun Suffix Group ($NSG$) and a boolean flag $pdm$.

Output: Inflectional Set $W_{p_i,l_j}$.

/* Compute Inflectional Set for lemma $l_j$ corresponding to paradigm $p_i$ */
$W_{p_i,l_j} = \emptyset$ // Inflectional Set is initialized to NULL
BFR-List= getBaseFormationRules($p_i$)
For each $BFR_j$ in BFR-List
  /* $b_j$ is the $j^{th}$ Base corresponding to the paradigm */
  $b_j = $ applyBFR($l_i,BFR_j$)
  /* get Formative Suffix corresponding to the $j^{th}$ Base */
  $s_l = $ getFormativeSuffix($p_i,BFR_j$)
  /* get suffix set corresponding to the $j^{th}$ Base */
  $S_j = $ getSuffixSet($p_i,BFR_j$, $s_l$)
  $pdm_l = $ getPardigmDifferentiatingMeasure($p_i,BFR_j$, $s_l$) // get the paradigm differentiating measure value corresponding to the $j^{th}$ Base and Formative Suffix.
  For each $s_{jk}$ in $S_j$ // for each suffix in the suffix set
    /* Generate Inflectional Set only for suffixes with $pdm = 1$ */
    If $pdm=1$ then
      If $pdm_l=1$ then
        $w_{jk} = b_j.s_l.s_{jk}$
        $W_{p_i,l_j} = W_{p_i,l_j} \cup \{w_{jk}\}$
      End If
    End If
    /* Generate Inflectional Set for all suffixes */
    Else
      $w_{jk} = b_j.s_l.s_{jk}$
      $W_{p_i,l_j} = W_{p_i,l_j} \cup \{w_{jk}\}$
    End If
  End For
End For

Figure 4.8: Algorithm: Inflectional Set Generator for Noun.

level i.e. those words with $pdm$ set to 1. It makes use of a Finite State Machine based sequencing of suffixes to generate the different possible suffixes. The suffixes generated are attached to the noun Stem to give inflectional word forms that form the Inflectional Set. The noun Stem is obtained by attaching the corresponding Formative Suffix to the noun Base.

Surface Paradigm Relevance Checker

This module is used to select the relevant Surface Level Morphological Paradigm for
Chapter 4. AutoParSe for Generation of Morphological Lexicon

The input lemma $l_i$ from amongst the Candidate Paradigms in $CP$. The algorithm for the Surface Paradigm Relevance Checker module is illustrated in Figure 4.9.

**Algorithm: Surface Paradigm Relevance Checker for Noun**

**Input:** Noun lemma $l_i$, Candidate Paradigm set $CP$, set of unique corpus words $W_C$

**Output:** Relevant paradigm set $R_P$.

/* Select Relevant Surface Level Paradigm*/

threshold=2 // Initialize support to zero.
pdm=1 // Use paradigm differentiating measure to calculate Inflectional Set.
$R_P = \emptyset$

For each $p_i \in CP$

support = 0 //Initialize support to Zero

$W_{p_i, l_i} = \text{generateInflectionalSet}(l_i, p_i, \text{pdm})$

$SES_{p_i,l_i} = W_{p_i, l_i} \cap W_C$

support $= SEV_{p_i,l_i}$

If support $> \text{threshold}$

$R_P = R_P \cup p_i$

End If

End For

**Figure 4.9:** Algorithm: Surface Paradigm Relevance Checker for Noun.

The relevant Surface Level Noun Paradigm is selected for the input lemma $l_i$ using the following steps:

1. For each candidate paradigm in $CP$, it calculates the Suffix Evidence Set $SES$ and Suffix Evidence Value $SEV$ such that only those words which distinguish the paradigm are computed for $SES$ i.e. the $pdm$ is set to 1 for computing $SES$ and $SEV$.

2. Choose all those paradigms in $CP$ that have $SEV > \text{threshold}$ as relevant paradigms for the input lemma $l_i$.

The threshold has been set to two as more than two inflectional word forms unique to a particular paradigm are good indicators that the input lemma follows that paradigm.
Paradigm Pruner

This module is used to prune paradigms that are incorrectly assigned to the lemma and is done to increase the precision of paradigm selection.

Case wherein incorrect paradigms get assigned to nouns is as follows:

- **Derived word / Parallel word inflections in Corpus:** When lemma $l_i$ has a derived form or parallel form $l_j$ and $l_i$ follows paradigm $p_i$ and $l_j$ follows $p_j$ it has been observed that another paradigm $p_k$ generates the same set of inflectional words for $l_i$ as $p_j$ for $l_j$ which are found in corpus, thus causing incorrect paradigm to be assigned. This happens because certain oblique formative suffixes also act as derivational suffixes or parallel word ending characters in Konkani.

**Example 1:** For the noun lemma $l_i = नोकर (nokara) (servant)$, two paradigms were being assigned namely $p_i$ and $p_k$. The incorrect paradigm $p_k$ assigned was because the word forms generated by $p_k$ for $l_i$ were present in the corpus as they corresponded to the word forms of the derived word lemma $l_j = नोकरी (nokarI) (job) (naukari, job)$.

**Example 2:** For the noun lemma $l_i = घोडो (ghoDo) (male horse)$, two paradigms were being assigned namely $p_i$ and $p_k$. The incorrect paradigm $p_k$ assigned was because the word forms generated by $p_k$ for $l_i$ were present in the corpus as they corresponded to the word forms of the parallel word lemma $l_j = घोडी (ghoDI) (female horse)$.

Such incorrectly assigned paradigms are pruned in this module by performing the following steps

1. Generate all possible derived forms and parallel forms which have overlap with the oblique formative suffixes.

2. For derived forms and parallel forms present in the noun Lexicon $LX_N$ generate $SES$ with $pdm$ set to 1.
3. For each paradigm in Relevant Paradigms recompute SES with \(pdm\) set to 1

4. Prune paradigms for which intersection of SES computed in step 2 and 3 is not \(\emptyset\)

The algorithm for the Paradigm Pruner module is illustrated in Figure 4.10.

---

**Algorithm: Paradigm Pruner for Noun**

**Input:** Noun lemma \(l_i\), Relevant Paradigm Set \(R_P\), set of unique corpus words \(W_C\), Derivational Cum Oblique suffix List \(DOS\), Noun Lexicon \(LX_N\)

**Output:** Pruned Relevant paradigm set \(R_P\).

/* Prune Relevant Surface Level Paradigm*/

For each \(w_i \in WordList\)

- If \(w_i \in LX_N\)
  
  - \(CP = \text{getCandidateParadigm}(w_i)\)
  
  - For each \(p_i \in CP\)
     
    - \(SES_{p_i} = \text{getSES}(p_i, w_i)\)
    
     - For each \(p_j \in R_P\)
        
         - \(SES_{p_j} = \text{getSES}(p_j, l_i)\)
         
         - If \(SES_{p_i} \cap SES_{p_j} \neq \emptyset\)
            
             - Remove \(p_j\) from \(R_P\)

End For
End For
End If
End For
End For

---

**Figure 4.10:** Algorithm: Paradigm Pruner for Noun.

---

**Lexical Paradigm Relevance Checker**

This module is used to select an appropriate Lexical Level Paradigm for those Surface Level Paradigms which can be split to Lexical Level Paradigms. A training data set is prepared for each Lexical Level Paradigm. The features used in the data set are listed in Table 4.1. These features were chosen after observing that, in Konkani Lexical Level Paradigms, for one paradigm, the Direct Singular Form (DSF) and Direct Plural Form (DPF) are the same while for the other paradigm, Direct Plural Form (DPF) and Plural Oblique Form (POF) were the same. In general, these features correspond to those word forms that have multiple grammatical roles i.e. those word forms which cause
### Table 4.1: Data Set Features for Lexical Level Paradigm.

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Feature Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID</td>
<td>The paradigm identifier.</td>
</tr>
<tr>
<td>FreqDSF</td>
<td>Number of times the direct singular form of the noun occurs in the corpus.</td>
</tr>
<tr>
<td>FreqSOF</td>
<td>Number of times the oblique singular form of the noun occurs in the corpus.</td>
</tr>
<tr>
<td>FreqPOF</td>
<td>Number of times the oblique plural form of the noun occurs in the corpus.</td>
</tr>
<tr>
<td>TotSOV</td>
<td>Total number of oblique singular forms of the noun occurring in the corpus.</td>
</tr>
<tr>
<td>SOVWin</td>
<td>Distinct number of types amongst oblique singular forms of the noun occurring in the corpus.</td>
</tr>
<tr>
<td>TotPOV</td>
<td>Total number of oblique plural forms of the noun occurring in the corpus.</td>
</tr>
<tr>
<td>POVWin</td>
<td>Distinct number of types amongst oblique plural forms of the noun occurring in the corpus.</td>
</tr>
</tbody>
</table>

ambiguity. The intuition behind choosing these features was that, if in one paradigm a particular word form has multiple grammatical roles, than its corresponding relative frequency should differ from the other paradigm where it has a single grammatical role.

**Example:** Let $p_i$ and $p_j$ be two paradigms which are same at surface level but differ at lexical level. Let $l_i$ be the input lemma. In paradigm $p_i$, let the word form $w_i$ have two grammatical roles as in case of Konkani word फातार (phaatara) (stone) which is both Direct Singular Form (DSF) and Direct Plural Form (DPF). In paradigm $p_j$, let the same word form $w_i$ have only one grammatical role which is Direct Singular Form (DSF) and has a different form $w_j$ for Direct Plural Form (DPF) which is also Plural Oblique Form (POF). Thus in the data set for paradigm $p_i$, frequency of DSF and POF will follow a different pattern when compared to frequency of DSF and POF in $p_j$.

We picked Logistic Regression as our learning model as it works well on numeric data,
is simple and performed better than other machine learning classifiers as illustrated in Table 4.3. We created a training data set with 356 noun lemmas and assigned the paradigm identifier manually. This was used as a training model to pick lexical level paradigm for the input lemma. The algorithm for the Lexical Paradigm Relevance Checker module is illustrated in Figure 4.11.

Algorithm: Lexical Paradigm Relevance Checker for Noun

Input: Noun lemma $l_i$, Lexical Training Data Set $TDS$, set of unique corpus words $W_C$, Lexical Noun Paradigm List ($P_LN_L$), Pruned Relevant paradigm set $R_P$, Surface Noun Paradigm List ($P_LN_S$)

Output: Relevant paradigm set with lexical paradigms $R_P$.

/* Select appropriate Lexical Level Paradigm */
For each $p_i \in R_P$
  If $p_i \in P_LN_L$
    /* Compute corresponding Feature Set $FS$ for Lexical Level Paradigm*/
    $FS = \text{computeFeatureSet}(l_i, W_C, p_i, P_LN_S)$
    $R_{p_i} = \text{applyLogisticRegression}(TDS,FS)$
    Replace $p_i$ with $R_{p_i}$ in $R_P$
  End If
End For

Figure 4.11: Algorithm: Lexical Paradigm Relevance Checker for Noun.

4.4.3.6 Algorithm: Noun Morphological Lexicon Generation

This Automatic Paradigm Selection (AutoParSe) algorithm is used to generate a Noun Morphological Lexicon. The algorithm for Noun Morphological Lexicon Generation is illustrated in Figure 4.12

The input to the algorithm is a Noun Lexicon $LX_N$ consisting of noun lemmas. For each noun lemma in $LX_N$ the relevant Morphological Paradigms are selected using the following two steps:

1. Compute the Candidate Surface Level Paradigm List ($CP$) using the Candidate Paradigm Generator module for the input lemma $l_i$. This module selects relevant
Algorithm: Noun Morphological Lexicon Generation

**Input:** Set of Noun Lemmas ($LX_N$), Set of unique corpus words ($W_C$), Set of Noun Suffix Group ($NSG$), Surface Noun Paradigm List $PLNS$, Lexical Noun Paradigms List $PLNL$, Lexical Training Data Set $TDS$, Derivational Cum Oblique suffix List $DOS$.

**Output:** Noun Morphological Lexicon set $LX_{NM}$.

$LX_{NM} = \emptyset$

For each lemma $l_i \in LX_N$

/* Compute compatible candidate paradigms */
$CP = \text{generateCandidateSurfaceParadigm}(l_i, PLNS, NSG)$

/* Compute Relevant Surface Paradigm Set */
$RP = \text{SurfaceParadigmRelevanceChecker}(l_i, CP, W_C)$

/* Prune Relevant Surface Paradigm Set */
$RP = \text{ParadigmPruner}(l_i, RP, W_C, DOS, LX_N)$

/* Compute Relevant Lexical Paradigm */
$RP = \text{LexicalParadigmRelevanceChecker}(l_i, TDS, W_C, PLNL, RP, PLNL)$

/* Make entry in Noun Morphological Lexicon $LX_{NM}$ */

For each $p_i$ in $RP$

$LX_{NM} = LX_{NM} \cup (l_i, p_i)$

End For

End For

**Figure 4.12:** Algorithm: Noun Morphological Lexicon Generation.

Candidate Surface Level Paradigms from amongst a list of Surface Level Noun Paradigms by pruning those paradigms which are not compatible with the input lemma $l_i$.

2. Relevant Surface Level Paradigms from amongst the Candidate Surface Level Paradigms are selected using the Surface Paradigm Relevance Checker module. It involves the following steps:

(a) It makes use of the Inflectional Set Generator module in order to generate the Inflectional Set corresponding to suffixes whose $pdm$ is 1.

(b) It generates Suffix Evidence Set $SES_{p_i,l_j}$ and Suffix Evidence Value $SEV_{p_i,l_j}$ for each noun lemma $l_j$ corresponding to paradigm $p_i$ using Inflectional Set generated in previous step.

(c) It selects those paradigms with $SEV_{p_i,l_j} > $ threshold as relevant paradigm and adds them to relevant paradigm set $RP$.  


3. Incorrect Paradigms assigned to \( R_P \) are pruned using Paradigm Pruner module to give pruned \( R_P \).

4. Lexical Level Paradigms are assigned to paradigms in \( R_P \) by replacing Surface Level Paradigms with Lexical Level Paradigms.

5. It adds the (lemma, relevant paradigm set) \( (l_i, R_P) \) pair to the Noun Morphological Lexicon \( (LX_{NM}) \)

### 4.5 Experimental Results and Evaluation

The implementation of AutoParSe for Generation of Morphological Lexicon was done in Java. The framework was used to develop noun and verb Morphological Lexicon for Konkani.

#### 4.5.1 Data Sets

The following resources were used to create the data sets used in AutoParSe.

- **Konkani Verb Lexicon** \( (LX_V) \): Konkani Verb Lexicon resource with 1,226\(^{16}\) entries was extracted from the Konkani WordNet, which had 3,844 unique verb entries at the time of the study.

- **Konkani Noun Lexicon** \( (LX_N) \): Konkani Noun Lexicon resource with 20,369 entries was extracted from the Konkani WordNet which had 26,278 unique noun entries at the time of the study.

- **Konkani Corpus** \( (C) \): Konkani Corpus resource with 2,66,511 unique word forms was extracted from Konkani Corpus created by Asmitai Pratishthan with approximately 11,000,00 total words.

\(^{16}\)Data preprocessing caused reduction in number of entries considered for the study
Chapter 4. AutoParSe for Generation of Morphological Lexicon

- **Noun/ Verb Suffix Groups** ($NSG / VSG$): Verb Suffix List, Noun Case Marker Suffix Lists, Clitic Suffix List were prepared after examining the Suffix List generated by SAM based Morphology Learning and enhancing the same. Complex suffixes\(^{17}\) were generated automatically using foma, a finite state compiler. Suffixes were grouped manually to form suffix groups constrained by the morphotactics in the language.

- **Noun/ Verb Paradigm List** ($PLV / PLNS / PLL$): Paradigm Lists were prepared using the partial Paradigm List generated by SAM based Morphology Learning by enhancing the same by referring to Konkani grammar book (Borkar, 1992), Konkani linguistic dissertation studies (Almeida, 1989, Sardessai, 1986) and discussions with linguists.

### 4.5.2 Data Preprocessing

The following preprocessing steps were used to prepare the data for the experimental study.

- **Multi-word removal**: Words in the synset\(^{18}\) of the Konkani WordNet represented by multiword expression were pruned from the input while extracting noun and verb Lexicons. Such multiword expressions inflect the same as its component last word, hence were pruned.

  **Example 1**: Multi-word noun entries such as ल्हान_दोंगर ($lhana_do.Ngara$) (hillock) were pruned as ल्हान_दोंगर ($lhana_do.ngara$) inflects the same way as दोंगर ($do.ngara$) (hill).

  **Example 2**: Multi-word verb entries such as चिमटो_काडप ($chimaTo_kaaDapa$) (to pinch) were pruned as चिमटो_काडप ($chimaTo_kaaDapa$) inflects the same way as काडप ($kaaDapa$) (remove).

\(^{17}\)Suffixes which are formed by placing more than one suffix next to each other

\(^{18}\)A synset stands for synonymous set, consists of a group of synonymous words which can be used interchangeably. Synset are used to represent a concept in the language.
Chapter 4. *AutoParSe for Generation of Morphological Lexicon*

- **Unique word forms:** Since in a WordNet, a word lemma can appear in more than one synsets, we removed the duplicates and retained only unique word lemmas in our input lexicons.

### 4.5.3 Experimental Results

A gold standard data was manually created to test the performance of AutoParSe. The lemmas for the gold standard were selected such that all the paradigms are equally represented.

Two sets of experiments were conducted namely

- AutoParSe for Generating Verb Morphological Lexicon.
- AutoParSe for Generating Noun Morphological Lexicon.

Entries corresponding to lemmas for which no paradigm was assigned because their inflectional forms were not found in the corpus were considered as False Negatives. Entries corresponding to lemmas are considered as incorrect if a relevant paradigm is incorrect. The precision and recall were computed using the formulas given below.

\[
Precision = \frac{TruePositives}{(TruePositives + FalsePositives)}
\]

i.e. \( Precision = \frac{\#CorrectlyAssignedEntries}{(\#CorrectlyAssignedEntries + \#IncorrectlyAssignedEntries)} \)

\[
Recall = \frac{TruePositives}{(TruePositives + FalseNegatives)}
\]

i.e. \( Recall = \frac{\#CorrectlyAssignedEntries}{(\#CorrectlyAssignedEntries + \#UnAssignedEntries)} \)

### 4.5.3.1 Results - AutoPaSe for Verb Morphological Lexicon Generation

To evaluate the performance of AutoParSe for Konkani verbs, paradigms were manually assigned to each of the 1,226 verbs in the input verb lexicon. The results obtained for Verb Morphological Lexicon Generation are tabulated in Table 4.2.
Table 4.2: Results for Verb Morphological Lexicon Generation.

<table>
<thead>
<tr>
<th>Category</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of verbs used for the study</td>
<td>1,226</td>
</tr>
<tr>
<td>Total number of verbs for which correct paradigms were selected (true positives)</td>
<td>791</td>
</tr>
<tr>
<td>Total number of verbs for which incorrect paradigms were selected (false positives)</td>
<td>53</td>
</tr>
<tr>
<td>Total number of verbs for which paradigms were not selected (false negatives)</td>
<td>382</td>
</tr>
</tbody>
</table>

Precision = 0.937  Recall = 0.674  F-Score = 0.784

4.5.3.2 Results - Model Selection for Lexical Level Paradigm Assignment

To choose the model for lexical level paradigm assignment, we ran various classification algorithms on our development data sets created with features listed in Table 4.1 using 10 fold cross validation to determine the best training model. The performance of machine learning classifiers on our data set are tabulated in Table 4.3. Here Precision, Recall and F-score are the weighted average values generated.

Table 4.3: Model Selection for Lexical Level Paradigm Selection.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bayesian Classifiers</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>0.796</td>
<td>0.815</td>
<td>0.785</td>
</tr>
<tr>
<td>Bayes Net</td>
<td>0.787</td>
<td>0.806</td>
<td>0.79</td>
</tr>
<tr>
<td>Function Classifiers</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Continued on next page
### Table 4.3 – Continued from previous page

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistic</td>
<td>0.94</td>
<td>0.941</td>
<td>0.94</td>
</tr>
<tr>
<td>MultilayerPerceptron</td>
<td>0.821</td>
<td>0.834</td>
<td>0.822</td>
</tr>
<tr>
<td>RBFNetwork</td>
<td>0.806</td>
<td>0.82</td>
<td>0.79</td>
</tr>
<tr>
<td>SimpleLogistic</td>
<td>0.958</td>
<td>0.958</td>
<td>0.957</td>
</tr>
<tr>
<td>SMO</td>
<td>0.839</td>
<td>0.798</td>
<td>0.723</td>
</tr>
</tbody>
</table>

**Instance-Based Classifiers**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>IB1</td>
<td>0.84</td>
<td>0.846</td>
<td>0.842</td>
</tr>
<tr>
<td>KStar</td>
<td>0.828</td>
<td>0.834</td>
<td>0.807</td>
</tr>
</tbody>
</table>

**Ensemble Classifiers**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdaBoost</td>
<td>0.915</td>
<td>0.916</td>
<td>0.912</td>
</tr>
<tr>
<td>Bagging</td>
<td>0.937</td>
<td>0.938</td>
<td>0.938</td>
</tr>
<tr>
<td>Random Sub Space</td>
<td>0.898</td>
<td>0.896</td>
<td>0.887</td>
</tr>
<tr>
<td>Decorate</td>
<td>0.952</td>
<td>0.952</td>
<td>0.951</td>
</tr>
<tr>
<td>Logit Boost</td>
<td>0.932</td>
<td>0.933</td>
<td>0.93</td>
</tr>
</tbody>
</table>

**Rule-Based Classifiers**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>PART Decision List</td>
<td>0.94</td>
<td>0.941</td>
<td>0.94</td>
</tr>
<tr>
<td>Ridor</td>
<td>0.94</td>
<td>0.941</td>
<td>0.94</td>
</tr>
<tr>
<td>ZeroR</td>
<td>0.61</td>
<td>0.781</td>
<td>0.685</td>
</tr>
</tbody>
</table>

**Decision Tree Classifiers**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Forest</td>
<td>0.928</td>
<td>0.93</td>
<td>0.928</td>
</tr>
<tr>
<td>Logistic Model Tree</td>
<td>0.977</td>
<td>0.978</td>
<td>0.977</td>
</tr>
<tr>
<td>REPTree</td>
<td>0.936</td>
<td>0.935</td>
<td>0.936</td>
</tr>
</tbody>
</table>

Analyzing the performance of the various classifiers from Table 4.3, Logistic Regression was chosen as a training model to select relevant paradigms from amongst those which
differ at lexical level only.

4.5.3.3 Results - AutoPaSe for Noun Morphological Lexicon Generation

To evaluate the performance of AutoParSe for Konkani nouns, a gold standard data was prepared consisting of 2,000 noun lemmas selected from the input noun lexicon, which were assigned paradigms manually. The results obtained for Noun Morphological Lexicon Generation are tabulated in Table 4.4

<table>
<thead>
<tr>
<th>Category</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of nouns used for the study</td>
<td>2,000</td>
</tr>
<tr>
<td>Total number of nouns for which correct paradigms were selected (true positives)</td>
<td>1,685</td>
</tr>
<tr>
<td>Total number of nouns for which incorrect paradigms were selected (false positives)</td>
<td>67</td>
</tr>
<tr>
<td>Total number of nouns for which paradigms were not selected (false negatives)</td>
<td>248</td>
</tr>
</tbody>
</table>

**Table 4.4:** Results for Noun Morphological Lexicon Generation.

Precision = 0.961    Recall = 0.871    F-Score = 0.914

4.5.4 Analysis of Experimental Results

We analyzed the results from two perspectives namely

1. Precision of Paradigm Selection

2. Recall for Paradigm Selection
Precision of Paradigm Selection

We observed a high precision of paradigm selection for both nouns and verbs. However paradigm selection failed in cases where the inflected form of the lemma obtained using an incorrect paradigm was not actually an inflection, but another word present in the corpus, which happened to have a word ending which match the suffixes.

**Example:** For the input noun lemma कात (kaata) (skin), a particular paradigm generates an inflected form कातार (kaataar) (Qatar country name) found in the corpus hence this incorrect paradigm gets selected for the input lemma. The correct inflection for lemma कात (kaata) (skin) is कातीर (kaatIra) (on the skin).

We prune derivational or parallel word paradigms incorrectly assigned in Noun Lexicon generation. This pruning increases precision of AutoParSe for nouns. Table 4.5 illustrates the effect of pruning on precision.

<table>
<thead>
<tr>
<th>Method</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>AutoParSe without pruning</td>
<td>0.932</td>
</tr>
<tr>
<td>AutoParSe with pruning</td>
<td>0.961</td>
</tr>
</tbody>
</table>

Recall for Paradigm Selection

When we analyzed the results to check if we could further improve the recall, we realized that it was not absolutely necessary to do so. The factors which caused the recall to reduce was unassigned lemmas\(^{19}\). We studied these unassigned lemmas to find possible reasons why paradigms were not selected for them. We observed that word forms of such nouns were not present in the corpus used. Table 4.6 shows the categories in unassigned lemmas.

We observed that these words were not natural to Konkani language. A natural question that arises is how do they appear in the input if they are not natural to the

\(^{19}\)We refer to those lemmas for which paradigms were not selected as unassigned lemmas
Table 4.6: Categories in Unassigned Lemmas

<table>
<thead>
<tr>
<th>Category Type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Named Entities</td>
<td>अंगोला (a.ngolaa) (Angola -name of a country)</td>
</tr>
<tr>
<td>Foreign words or borrowed</td>
<td>अका￷डयन (akaarDiyana) (Accordian) इंफ्रा–</td>
</tr>
<tr>
<td>words</td>
<td>स्ट्र क्चर (i.nphraasTrkcharu) (Infrastructure)</td>
</tr>
<tr>
<td></td>
<td>अटॅक (aT.cka) (Attack)</td>
</tr>
<tr>
<td>Coined words</td>
<td>अक्षरगणीत (axaragaNIta) (Aksharganit algebra)</td>
</tr>
<tr>
<td>Rare usage words</td>
<td>अकृ त्य (akrutya) (an action not to be performed)</td>
</tr>
</tbody>
</table>

language. The answer lies in the resource used, namely, Konkani WordNet, to create the lexicon. In a WordNet, a word is a part of a synset which follows the principal of minimality, coverage and replacebility (Bhattacharyya, 2010). As a result, to comply with the principal of coverage, rarely used synonyms of the word are also present in the WordNet. The fall in recall is due such rare words, which would have to be manually assigned paradigms.

4.6 Conclusion

AutoParSe algorithms designed and implemented helped to generate a Morphological Lexicon automatically. This helps in reducing the human effort required to create Morphological Lexicon. The salient feature of AutoParSe is its ability to assign multiple paradigms to a single lemma. For example the Konkani word मान (maana) (respect/neck) gets assigned to two different paradigms for each of the senses neck and respect. Overlap in inflectional and derivational suffixes in Konkani tend to reduce the precision and need to be handled appropriately. By assigning pdm to paradigm differentiating suffixes, we get a new way to correctly map multiple paradigms to a noun lemma, which reflects the different senses in which the noun could occur.
To the best of our belief, most automatic paradigm selectors tend to produce a single mapping which does not capture multiple senses of words.

Sometimes, linguists assigning paradigms to words could miss a word sense which is better captured by our automatic paradigm selector. Multiple paradigms also suggest multiple acceptable inflectional form uses for a word. For example, न्हेयर (nha.nyara) and न्हेयेर (nha.nyera) both mean on the river are both present in the corpus and should be captured by a good system. A corpus is bound to have spelling dialect variations of word form. AutoParSe is flexible enough to accommodate such inflectional form variations which a linguist may disagree with and stick to standard inflectional rules of the language. Thus, our AutoParSe method will be able cater better to a true real life corpus morphological analysis requirement.