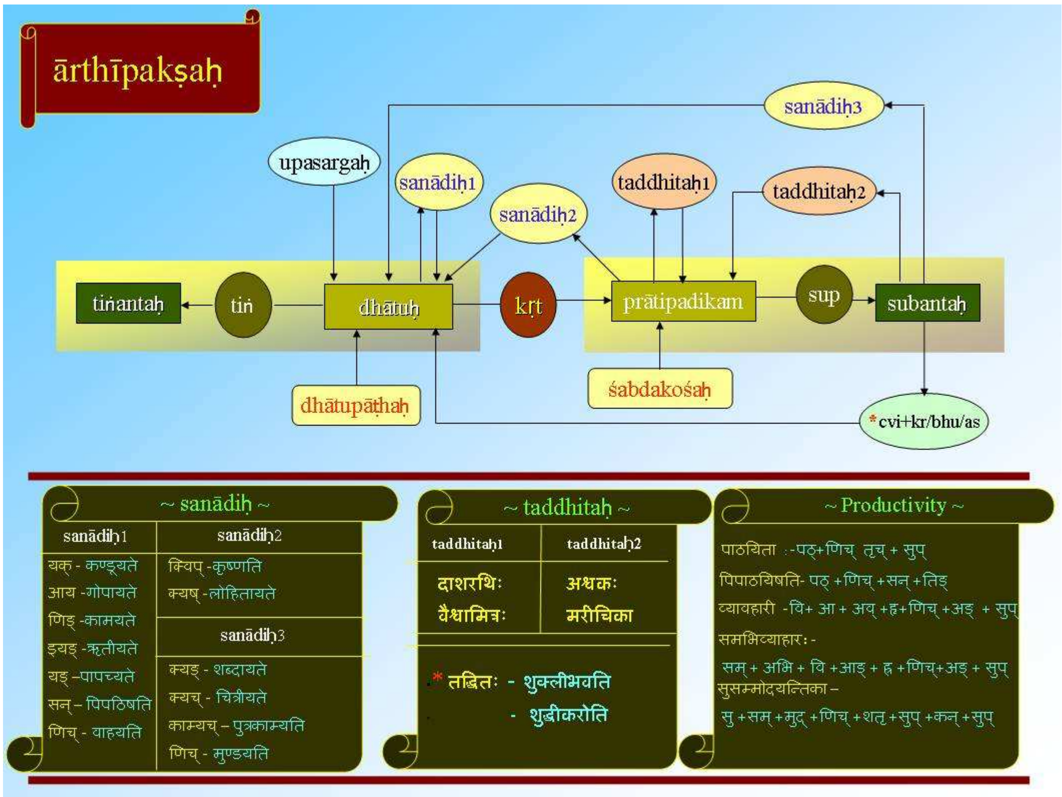
Sanskrit Morphological Analyser

Amba Kulkarni
Department of Sanskrit Studies
University of Hyderabad
Hyderabad, India
apksh@uohyd.ernet.in

February 3, 2009

1 Word Formation in Sanskrit:

The finite state automaton in figure 1 describes the word formation possibilities in Sanskrit. As can be seen from the figure 1, theoretically it is possible to generate not only infinite number of forms but also in size (in terms of morphemes in a word) from a given word. However, the ability of human mind to process a complex string puts a limit on these potentially infinite productions to a finite number and is supported by the actual data. Words with more than three suffixes are typically rare compared to the words with single suffix. Further, for all practical purposes, the derived word forms can always be kept in the lexicon as headwords. These may be analyzed if there is a need.



~ sanādiḥ ~

sanādiḥ1	sanādiḥ2
यक् - कण्डूयते	क्विप् -कृष्णति
आय -गोपायते	क्यष् -लोहितायते
णिङ् -कामयते	sanādiḥ3
इयङ् -ऋतीयते	क्यङ् - शब्दायते
यङ् –पापच्यते	क्यच् - चित्रीयते
सन् – पिपठिषति	काम्यच् – पुत्रकाम्यति
णिच् - वाहयति	णिच् - मुण्डयति

~ taddhitaḥ ~

taddhitaḥ1	taddhitaḥ2
दाशरथिः	अश्वकः
वैश्वामित्रः	मरीचिका

* तद्धितः - शुक्लीभवति
- शुद्धीकरोति

~ Productivity ~

पाठयिता :-पठ्+णिच् तृच् + सुप्
पिपाठयिषति- पठ् +णिच् +सन् +तिङ्
व्यावहारी -वि+ आ + अव् +हृ+णिच् +अङ् + सुप्
समभिव्याहारः -
सम् + अभि + वि +आङ् + हृ +णिच्+अङ् + सुप्
सुसम्मोदयन्तिका –
सु +सम् +मुद् +णिच् +शतृ +सुप् +कन् +सुप्

The complexity in Sanskrit morphology as shown above, is further aggravated by two factors: sandhi formation, and productive samāsa (compound) formation.

Table 1: Legends

dhaatu(verbal root)	dhaatupatha(list of verbal roots)	krut(nonfinite verbal suffix)
samaasa(compound)	sanaadi(derivational suffixes)	shabdakosha(lexicon)
subant(noun)	sup(nominal suffix)	taddhita(derivational suffix)
ting(finite verbal suffix)	tinganta(finite verb form)	upasarga(verbal prefix)

1.1 Sandhi formation

Like many South Indian languages, Sanskrit uses sandhi extensively. Sandhi is of two kinds external sandhi (rules which govern the sandhi between two words) and internal sandhi (rules which govern the sandhi within a word involving two or more morphological segments). The internal sandhi rules are used at the morphological level. But the external sandhi needs to be handled differently. Sanskrit has 48 phonemes or contrastive segments (A), leading to more than two thousand possible combinations of any two of them at a time (A x A). Out of these more than 60sandhi formation. Further, the mapping from AxA to A is a many-one mapping leading to multiple splittings corresponding to a single phoneme (segment). For example, there are 4 possibilities into which a segment 'ā' can be analysed:
ā -> a + a
ā -> a + ā
ā -> ā + a
ā-> ā + ā

Hence the word 'rāmālaya' can be analysed as composed of two words in eight possible ways (by taking into account the above 4 rules only) :
a) ra + amālaya
b) ra + āmālaya
c) rā + amālaya
d) rā + āmālaya
e) rāma + alaya
f) rāma + ālaya
g) rāmā + alaya
h) rāmā + ālaya
apart from the default spliitings as
i) r + āmālaya
j) rā + mālaya
k) rām + ālaya
l) rāmā + laya
m) rāmāl + aya
n) rāmāla + ya
o) rāmālay + a

A good coverage morphological analyzer can rule out the 10 of them, since the words amālaya, āmālaya, r, ra, rā, rām, rāmāl, and rāmāla, rāmālay are not validated through the lexicon. The complexity increases further as Sanskrit literary tradition being largely oral, there is a tendency to join the consecutive words through the process of sandhi. Therefore, if one or both the components are not validated through the lexicon, one needs to split these unrecognised components further recursively to ensure that no possibilities are missed out. Thus we see that a good coverage morphological analyzer needs a sandhi splitter and a sandhi splitter requires a good coverage morphological analyzer leading to an apparent double bind. It is possible to design the modules in such a way that core morphological analyser is built first which handles simple words and then there is a module for sandhi which in turn calls the core morphological analyser to validate the split suggested by the sandhi splitter.

1.2 Samāsa (compound) formation

Another feature that increases the complexity of Sanskrit word formation is productive samāsa formation. Since the samāsas involve semantic component, they are to be handled at two levels – syntactic and semantic. At syntactic level, since sandhi is involved, we need a tight coupling of core morphological analyser, sandhi splitter and a samaasa splitter. At semantic level, once we get the constituent elements of a samāsa, we need a module to decide its meaning. Figure 2 describes the syntactic and semantic aspects of the samāsa.

Six different samāsa combinations[1] in Sanskrit are possible as shown below.
Subanta (noun) + Subanta (noun) Example: rāj-puruṣaḥ
Subanta (noun) + Tiṅanta (verb) Example: parya-bhūṣayat
Subanta (noun) + Prātipadika (nominal root) Example: kumbha-kāraḥ
Subanta (noun) + Dhātu (verbal root) Example: kata-prū
Tiṅanta (verb) + Subanta (noun) Example: kṛnta-vicakṣaṇā
Tiṅanta (verb) + Tiṅanta (verb) Example: khādata-modatā .

Unlike English, where the words in a compound are written typically as distinct words (separated by spaces or by hyphen), in Sanskrit the components of samāsa's are joined together to form a single word. Naturally the components undergo a process of sandhi formation.

Two texts were selected for analysis: one prose (pancatantra) and one poetic (the first kānda of rāmāyayaṇa). It was found that around 20% to 25% of the words in these texts were compounds. The compound formation process in Sanskrit (for at least some types of compounds) is very productive. Therefore,

[1] supāṃ supā tiṅā nāmnā dhātunātha tiṇā tiṇā
subantena-iti vijñeyaḥ samāsaḥ ṣaḍvidhau budhaiḥ

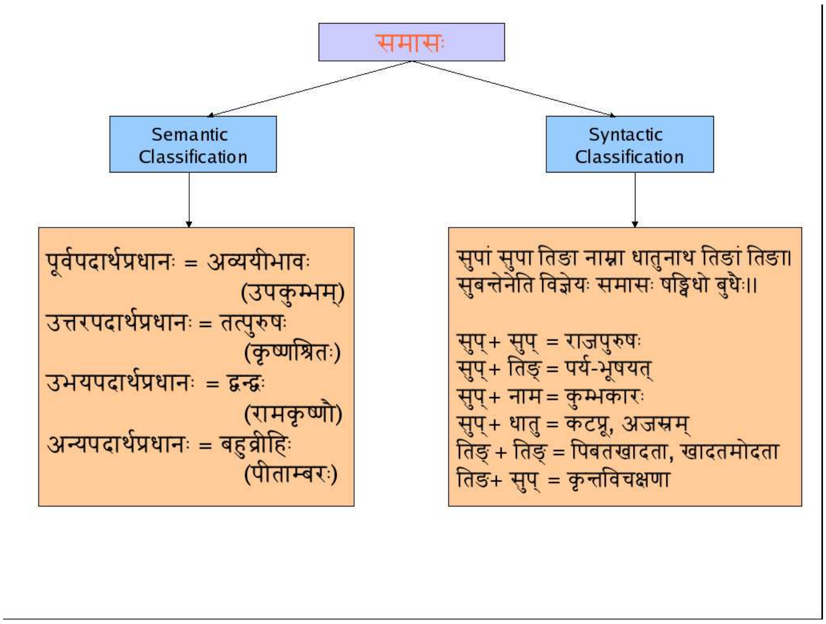

समासः
Semantic Classification
Syntactic Classification
पूर्वपदार्थप्रधानः = अव्ययीभावः (उपकुम्भम्)
उत्तरपदार्थप्रधानः = तत्पुरुषः (कृष्णश्रितः)
उभयपदार्थप्रधानः = द्वन्द्वः (रामकृष्णौ)
अन्यपदार्थप्रधानः = बहुव्रीहिः (पीताम्बरः)
सुपां सुपा तिङा नाम्ना धातुनाथ तिङां तिङा।
सुबन्तेनेति विज्ञेयः समासः षड्विधो बुधैः।।
सुप्+ सुप् = राजपुरुषः
सुप्+ तिङ् = पर्य-भूषयत्
सुप्+ नाम = कुम्भकारः
सुप्+ धातु = कटप्रू, अजस्रम्
तिङ् + तिङ् = पिबतखादता, खादतमोदता
तिङ+ सुप् = कृन्तविचक्षणा

Figure 1: Samāsa classification in Sanskrit

it is not possible to include all the compounds in the lexicon.

It should be clear, thus, that to have a good coverage morphological analyser, one needs a compound analyser. Compound formation involves sandhi. Therefore a compound analyser needs a sandhi splitter. And a sandhi splitter in turn requires a morphological analyser leading to circularity!

To avoid this circularity, we propose to build a morphological analyser that handles simple words. We call this analyser a ‘core morphological analyser’. A sandhi splitter calls this core morphological analyser. For a samaasa splitter also we require a sandhi splitter. But unlike a sandhi splitter, which confirms that all the cnstituents produced by the sandhi splitter must be ’pada’s, in case of samāsa we ensure that all the previous words are samāsa pūrva padas and the final one is a pada. So the interaction among these three modules may be summarised as in figure 3.

2 Core Morphological Analyser

The core morphological analyser assumes that the words are sandhi split and also the constituents of samaasas are split. The analyser should analyse both the inflectional as well as the derivational morphology. We first discuss the inflectional morphology.

2.1 Inflectional Morphology

Pāṇini defines a word as 'sup tiṅantam padam'. Anything that ends in a sup or tiṅ suffix is a pada. Thus there are only two categories. A category of bases called pratipadika (nominal base) take a sup suffix and a subant or a noun is formed. A verbal base called a dhātu takes a tiṅ suffix and a verbal form is generated.

Pāṇini has listed all the verbal bases in the dhātupāṭha. However, since there are variations in the dhātupāṭha and also few more dhātus have been added since Pāṇini's time, we started with the common dhātus along with the new dhātus which are missing in the original dhātupāṭha. Where there are variations in the base form, the most common form has been chosen.

The verbal bases given by Pāṇini are with the 'it'(markers). But these markers are not typically used in the dictionary. For example the verbal base 'kṛ' has the markers 'du' and 'ñ', thus the full form of the verb in Pāṇini's dhātupāṭha is 'dukṛñ'. But the dictionary entry for the verb is 'kṛ'. So the question was, in the analysis, which one should be produced as a root – the verbal base without the markers or with the markers. It was found that, if there are more than one verbs with the same verbal base, and are typically distinguished by their markers. So we can not consider the verbal base without the markers as a root. If we consider the verbal base with markers as the root, we found several verbs which have same verbal base with markers, but they belong to different gaṇas (classes). Hence only verbal base with the markers also can not give justice. The pair verbal-base with markers and gaṇas also are ambiguous between more than one readings. Finally it was found that the triplet verbal base without markers, gana and the verbal base with markers act as an unique identifier, and hence this triplet was selected as an identifier for the verb form analysis.

After the dhātupāṭha was selected, the next task was to decide the number of features the form depends on. Typically any dhātupāṭha keeps the following information of each of the verbs: a) dhatu with a marker b) dhatu without a marker c) gana d) set/anit/vet e) padI: AtmanepadI/parasmaipadI/ubhayapadI f) sakarmaka/akarmaka g) meaning of the verb

We have seen above that the triplet a), b) and c) uniquely determine the root. Then what is the role of other features? The feature set/anit/vet is useful in the generation. sakarmaka/akarmaka information is necessary for kaaraka analysis and determination of meaning. padI information is needed to decide the meaning of the verb form again. Now the question is, which of these are part of the morph analysis, and which of these are part of the lexicon. These questions then will decide which features should go in the morph analysis output and which should go in the dictionary.

The case of noun analysis was the easiest. The siddhānta kaumudī has adopted the paradigm model, and gives a list of paradigms for different endings

and different genders of the prātipadikas.
In case of nouns again, we require further sub categories as:

- nāma (noun)
- sarvanāma (pronoun)
- samkhyā (number)
- samkhyeya (cardinal)

One may argue that the information that whether a word is a noun or a ponoun lies in the lexicon iteself, then why should we have a separate category called sarvanāma? The answer is, there are words which when used as a pronoun may have different meaning and form from its usage as a noun. One such word is 'sva'. When it is used as a noun, its meaning is 'money' and its 7^{th} case singular form will be 'sve' form. But when it is used as a pronoun, its meaning is 'of_self' and its 7^{th} case singular form is 'svasmin'.

Similarly we require the cardinal and number as two different categories. Because, in Sanskrit, each of the numbers have a specific gender. But when they are used as a cardinal number, they are viśeṣaṇas, and they assume the same gender as their viśeṣyas. When the gender is changed, naturally, their form also changes. Thus from the form itself, many a times we know whether the word is a noun or an adjective. For example the word 'ekA' refers to a singular feminine nominal form, and is an adjective. If the word is used to denote a pure number, it will be in neuter gender. To account for this information, we have two categories for the numbers: pure numbers as they occur in mathematics and adjectives as they occur in everyday usage. Finally we require one more category to denote the ordinal numbers, which are always viśeṣaṇas.

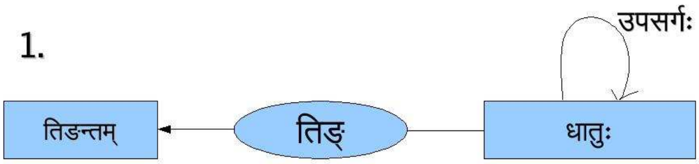
2.2 Derivational Morphology

Unlike modern Indian languages, Sanskrit has a very rich derivational component of morphological analysis. There are 6 primary ways to derive new bases – nominal as well as verbal. This derivational process being recursive, one may derive new bases, again by adding suffixes to the derived bases.

- No change in the Base category
  - verbal base + sanādi suffix => verbal base
  - upsarga + verbal base => verbal base
  - nominal base + taddhita suffix => nominal base
- Change in the Base category
  - verbal base + krudanta suffix => nominal base
  - nominal base + sanādi suffix => verbal base

Derivational Morphology

1.



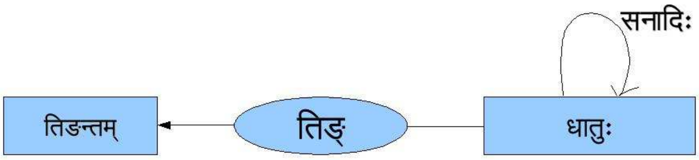
उपसर्गः

तिङन्तम् ← तिङ् — धातुः

उदा०- आ-गम्

1

2.



सनादिः

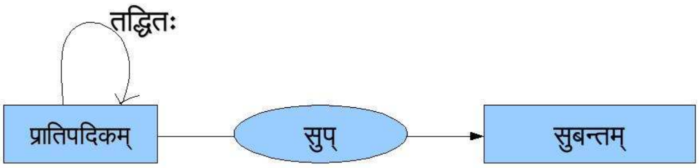
तिङन्तम् ← तिङ् — धातुः

सनादिप्रत्यया:- क्विप्, सन्, णिच्, यङ्, यक्, आय, इयङ्

उदा०- गम्_णिच्

2

3.



तद्धितः

प्रातिपदिकम्

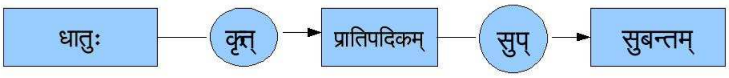
सुप्

सुबन्तम्

तद्धितप्रत्ययाः- त्व/ता, तर/तम, मय/मयी

उदा०- रामत्वम्

4.



धातुः

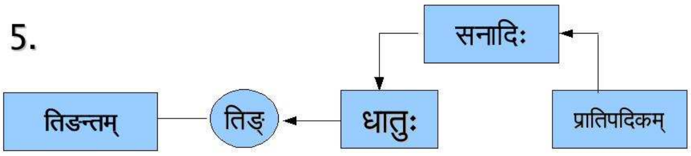
कृत्

प्रातिपदिकम्

सुप्

सुबन्तम्

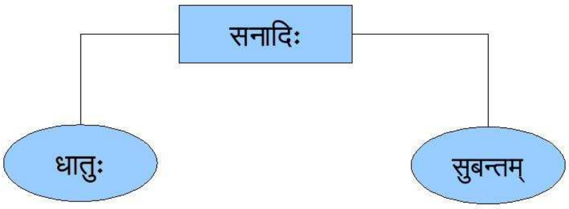
5.



प्रत्ययाः- क्विप्, णिच्

5

6.



नामधातुप्रत्ययाः-क्यच्, क्यङ्, काम्यच्, क्यष्, णिङ्

6

- noun + sanādi suffix => verbal base

As we see from the above, there are 3 cases, where the new base belongs to the same category as the original one, and there are 3 cases where the new base has different category than the original one.

We give here examples from each of these derivational categories to show how the analysis is represented in each of the cases.

- verbal base + sanādi suffix => verbal base
  pāṭhayati => root=paṭ1_ṇic, lakāra = laṭ, puruṣa = pra, vacana = 1, pada = parasmaipada, gaṇa = 1,dhatu_with_it = paṭha
  Note that the sanādi suffix has been added to the root.

- upsarga + verbal base => verbal base
  āgacchati => root = ā-gam1, lakāra = laṭ, puruṣa = pra, vacana = 1, pada = parasmaipada, gaṇa = 1,dhatu_with_it = gamlṛ
  Note that the upsarg and the root are joined by a '-' sign. this is to distinguish between a verbal base, upsarga from the sanādi suffix.

- nominal base + taddhita suffix => nominal base
  rāmatvena => root = rāmatva, liṅga = napuṁ, vanaca = 1, vibhakti = 3
  rāmata => root = rāma, taddhita_suff = tva, liṅga = napuṁ Note here that unlike in the above examples, here the root does not show the taddhita suffix. The reason is, in most of the cases this is not productive. Further, the meaning may also not be compositional, as in the case of other suffixes. So one will have to search these entries preferably in the dictionary, and if not available, then look at the derivation and produce the compositional meaning.

- verbal base + krudanta suffix => nominal base
  dattena => root = datta, liṅga = puṁ, vanaca = 1, vibhakti = 3
  datta => root = dā1, kṛt_pratyaya = kta, liṅga = puṁ

- nominal base + sanādi suffix => verbal base

- noun + sanādi suffix => verbal base

Here we have given examples with only one level of derivational morphology. In fact the Finite state tranducers for each of these modules is separate, and may be called when necessary. Since many of the times, even the derived nouns are available in the dictionary, we need not call the derivational morph. But if one desires so, there is a provision to get the detailed analysis further.
Thus, for example, suppose the input word is 'dattena'. The morph analyser will first give the inflectional morphology and produce an output with 'datta' as the root word. If the reader desires so, the analysis of the word 'datta' may

again be shown in a pop up window. The input and output specifications for the morphological analyser following the SSF format are given below.

3 Software Considerations:

In what follows we discuss the input and output specifications, and also the evaluation parameters and methodology. These will serve as standards to test the performance of different morphological analysers and also the same analyser at different stages of development.

3.1 Input Specification:

The input to the morphological analyser consists of two fields separated by TAB:
ADDR (Address)
TKN (Token)

3.2 Output Specifications:

The output of the morphological analyser consists of following three fields separated by TAB.
ADDR (Address)
TKN (Token) (in unicode)
FS (Feature structure)

Feature Structure is an attribute/value pair. Multiple feature structures are separated by '—'.

3.2.1 Inflectional Morphology

According to Pāṇini there are only two basic categories at the level of inflectional morphology. However, we for the sake of computational purpose, also consider avyaya as one of the categories. Further, when we consider the Vedic Sanskrit, we may require one more category, upasarga.

The basic categories for morphological analysis of Sanskrit, therefore, are

- नामपदम् (nāmapada) (Noun)
- क्रियापदम् (kriyāpada) (verb)
- अव्ययम् (avyaya) (indeclinable)
- उपसर्गः (upasarga) (pre-position?)

abbreviated feature name	in Sanskrit	feature name
root	प्रातिपदिकम् (prātipadika)	(Root of the word)
lcat	पद - विशेष (pada-viśeṣa)	(Lexical category of the word)
gend	लिङ्गम् (liṅgam)	(Gender of the word)
numb	वचनम् (vacanam)	(number corresponding to the word)
pers	पुरुषः (puruṣaḥ)	(person corresponding to the word)
vibh	विभक्तिः (vibhaktiḥ)	(case marker)

Table 2: Table 1

1. नामपदम् nāmapada (Noun):

   The feature structure for the noun is described in the table 1.

   The values of each of these features for Sanskrit is given below.

   - lcat(pada-viśeṣa)
     - n ना (nā) (If it is a nāmapada)
     - P सर्व (sarva) (If it is a sarvanāma)
     - num सं (saṁ) (If it is a cardial number)
     - ord पूरण (pūraṇa) (If it is an ordinal number)
   - gend (liñgam)
     - पुं puṅ (puñlliṅga)
     - स्त्री (strīliñga)
     - नपुं (napuñsakaliṅga)
   - numb (vacanam)
     - 1 (ekavacanam)
     - 2 (dvivacanam)
     - 3 (bahuvacanam)
   - pers (puruṣaḥ)
     - उ (uttama)
     - म (madhyama)
     - प्र (prathama)
   - vibh (vibhaktiḥ)
     - 1 (prathamā)
     - 2 (dvitīyā)
     - 3 (tṛtiyā)
     - 4 (caturthī)
     - 5 (pañcamī)

  - 6 (ṣaṣṭhī)
  - 7 (saptamī)
  - 8 (sambodhana)

2. kriyāpadam (Verb):
   The feature structure for verb is shown in the table 2.

abbreviated feature name	in Sanskrit	feature name
root	धातु (dhātu)	(Root of the word)
lcat	क्रि kri	(Lexical category of the word)
prayoga	प्रयोग (prayoga)	(voice)
lakaara	लकारः (lakāra)	(tense-aspect-modality marker)
pers	पुरुषः (purūṣa)	(person corresponding to the word)
numb	वचनम् (vacanam)	(number corresponding to the word)
pada	पदी (padī)	(pada-type)
dhatu_with_it	इत्_धातु (it_sahita_dhātu)	(root with a marker)
gana	गणः (gaṇa)	(verb-class)

Table 3: Table 2

The values of numb and pers are as above.

The values of other features for Sanskrit are given below.

- lakaara (lakāra)
  - लट् (laṭ)
  - लिट् (liṭ)
  - लुट् (luṭ)
  - लृट् (lṛṭ)
  - लोट् (loṭ)
  - लङ् (laṅ)
  - विधिलिङ् (vidhiliṅ)
  - आशीर्लिङ् (āśīrliṅ)
  - लुङ् (luṅ)
  - लृङ् (lṛṅ)
- pada (pada)
  - आत्मनेपदी (ātmanepadī)
  - परस्मैपदी (parasmaipadī)
- prayoga (prayoga)
  - कर्तरि (kartari)

  - कर्मणि (karmaṇi)
  - भावे (bhāve)
- gaṇa (gaṇa)
  - 1 (bhvādiḥ)
  - 2 (adādiḥ)
  - 3 (juhotyādiḥ)
  - 4 (divādiḥ)
  - 5 (svādiḥ)
  - 6 (tudādiḥ)
  - 7 (rudhādiḥ)
  - 8 (tanādiḥ)
  - 9 (kryādiḥ)
  - 10 (curādiḥ)

List of dhatu_with_it will be given in the appendix.

3. avyaya
The analysis of an avyaya will consist of the following features:
root (prātipadika) (Root of the word)
lcat ”avy” (Lexical category of the word)

The lcat will have a value “avy”

4. upasarga (pre-position)
(Note: This category is required only for Vedic Sanskrit literature.)
The analysis of an upasarga will consist of the following features:
root (prātipadika) (Root of the word)
lcat “upasarga” (Lexical category of the word)

3.2.2 Examples

We give below some example outputs.
Input:

1 वनम्

Output:

1 वनम् < fs root=वन, lcat=ना, gend=नपुं, numb=1, pers=प्र, vibh=1 > || < fs root=वन, lcat=ना, gend=नपुं, numb=1, pers=प्र, vibh=2 >

Input:

2 रामः

Output:

2 रामः < fs root=राम, lcat=ना, gend=पुं, numb=1, pers=प्र, vibh=1 > || < fs root=रा1, lcat=क्रिया, prayoga:कर्तरि, lakAra:लट्, pers:उ, numb:3, pada:परस्मैपदी, dhatu_with_it:रा, gana:2 >

3.2.3 Derivational Morphology

As we have seen above, there are two cases of derivational morphology – one where there is change in the category and the other one where there is no change inthe category. Accordingly the features also vary in case of the outputs. We show below the output in all the 6 cases discussed above.

- No change in the Base category
  - verbal base + sanādi suffix => verbal base
    Here the sanādi suffix will be added to the verbal base, joined by _, in the output.
    Example: पठ_णिच्
  - upsarga + verbal base => verbal base Here the upasarga will be added to the verbal base, joined by -, in the output.
    Example: आ-गम्
  - nominal base + taddhita suffix => nominal base Here the thaddhita suffix will be added to the nominal base, joined by _, in the output.
    Example: राम_त्व
- Change in the Base category
  Here we show the output in two layers. In the first layer we just show the inflectional analysis with the derived root as the base, and if necessary (when the derived word is not there in the dictionary), we show the derivational analysis as well.
  - verbal base + krudanta suffix => nominal base
    Example: गच्छति: < fs
    root=गच्छत्, lcat=ना, gend=पुं, numb=1, pers=प्र, vibh=7
    root=गम्, lcat=क्रि, k.rt_pratyaya=शतृ > ||
    < fs
    root=गच्छत्, lcat=ना gend=नपुं, numb=1, pers=प्र, vibh=7
    root=गम्, lcat=क्रि, k.rt_pratyaya=शतृ
    >

- nominal base + sanādi suffix => verbal base Examples to be added
- noun + sanādi suffix => verbal base Examples to be added

4 Evaluation Parameters

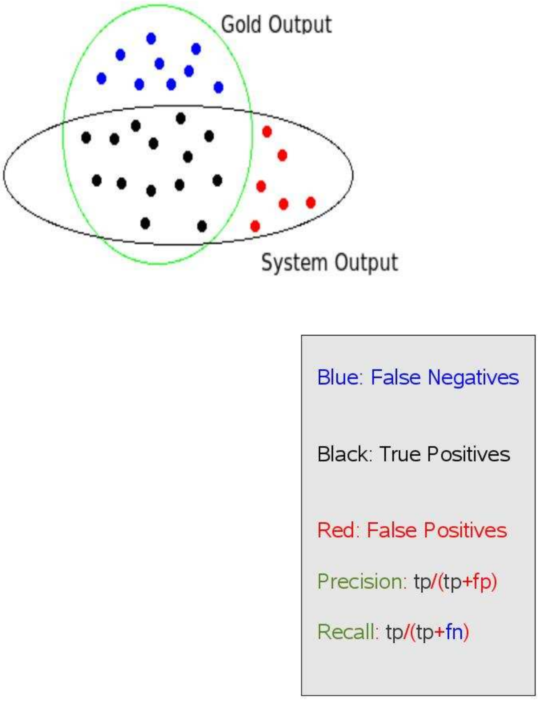
Typically the evaluation metrics of any tool involves two parameters – precision and recall. The precision tells you the confidence which you can have on the performance of the tool, and the recall gives you the coverage. Since the morphological analyser gives more than one answers, only these two parameters for evaluation of morphological analyser are not enough. We need a more sophisticated measure which tells us which of the answers produced by the morphological analyser are wrong, and also how many answers are missing. Further, though more than one ansers are possible, in many cases, some of the ansers are more probable and some of them are less.
Especially in case of Sanskrit, we have to address two kinds of users – serious researchers who want to understand the scientific texts in Sanskrit, and others who are interested in reading the literary texts in Sanskrit. The requirement criterion and evaluation of morphological analyser differ in both these cases. The serious reader would like to see all the possibilities, while a casual reader will be interested only in the more probable answers. Thus we need two seperate measures to cater to the needs of these two types of readers.

4.1 Evaluation metric for a serious reader

Let total number of words be N. Let A_i denote number of possible analysis for a word W_i. Thus total number of possible answers for all the words together is T = $\sum_{i=1}^{n} A_i$. Let B_i denote the answers produced by the morphological analyser for a word W_i. Let C_i answers denote the correct answers(true positives) and D_i denote wrong answers(false positives). Thus $B_i = C_i + D_i$.
Sum of all the correct answers is C = $\sum_{i=1}^{n} C_i$.
Sum of all the wrong answers is D = $\sum_{i=1}^{n} D_i$.
Precision $P = C/(C + W)$
Recall $R = C/T$

4.2 Evaluation metric for a casual reader

Let total number of words be N. Let A_i denote number of possible analysis for a word W_i.
Let B_i denote the answers produced by the morphological analyser for a word W_i.
Let C_i answers denote the correct answers and D_i denote wrong answers. Thus $B_i = C_i + D_i$.


Gold Output
System Output
Blue: False Negatives
Black: True Positives
Red: False Positives
Precision: tp/(tp+fp)
Recall: tp/(tp+fn)

Casual reader, unlike a serious reader would not be interested in the rare analysis. So we give weightage to the analysis based on its frequency of usage.

Following is the suggested weightage:
Highly probable analysis: 0.6
Less frequent but not rare: 0.3
Rare analysis: 0.1

It is necessary to carry out proper studies to come up with some criterion to decide the distribution of weightages.

Let wt_{ij} denote the weight of the j^{th} analysis of the i^{th} word.
With this weightage then, the total weight of all possible analysis is

$T = \sum_{i=1}^{n} \sum_{j=1}^{A_i} wt_{ij}$,
where $\sum_{j=1}^{A_i} wt_{ij} = 1$
Or simply, T = n, where n is the total number of words.

Total weight of all the correct analysis produced by morph analyser will be $C = \sum_{i=1}^{n} \sum_{k=1}^{C_i} wt_{ik}$, where wt_{ik} stands for the weight of the k^{th} correct analysis.

Assuming penalty for each wrong answer to be 1, total penalty for the wrong analysis produced by the morphological analyser will be
$W = \sum_{i=1}^{n} D_i$

Then the Precision and Recall will be
Precision $P = C/(C + W)$
Recall $R = C/T$

5 Evaluation: 6 point evaluation - an alternative to Precision-Recall

Since the morphological analyser produces more than one answers, it will be appropriate to carry out more detailed evaluation of the morpholoical analyser than just evaluating the precision and recall values.

Table 4 describes various possibilities:

The frquency count of each of these 6 cases will help the developers of morphological analysers to decide which aspect of morphology needs the further attention for improvement. It is the 4^{th} case which should have maximum frequency and the remaining should have less frequency.

As long as the wrong answered are pruned out by other modules, or the

sr no	Correct (True positives)	Missing (False Negatives)	Wrong (False Positives)	remark
1	1	1	1	ativyApti as well as avyApti
2	1	1	0	avyApti
3	1	0	1	ativyApti
4	1	0	0	Ideal Case
5	0	1	1	avyApti as well as ativyApti (sp case o
6	0	1	0	Unrecognised words avyApti (sp case o
The other two possibilities will never arise				

missing answers are less frequent, it is not harmful. So though it is desirable to reduce the false negatives, if the false negatives are less frequent, they do not affect the performance of the system much from a casual reader's point of view. But for a serious reader, false negatives also do matter.

5.1 Evaluation Methodology

The evaluation is to be carried out based on two GOLD standard data – one for a serious reader and the other from the point of view of a casual reader.

5.1.1 Serious Reader

Here we prepare a GOLD standard data of around 1000 words, chosen carefully to cover various apsects of morphology – paradigms, inflectional morphology and derivational morphology. This data is not disclosed to the groups who are developing the morphological analysers.
The outputs of the morphological analysers by various groups are compared with the manually created data for the 6 point scale as well as precision-recall.

5.1.2 Casual Reader

Here we prepare a GOLD standard data for the analysis of words in context. Many such databases are already available. For example, the Department of Sanskrit Studies, University of Hyderabad, has analysis of 100 shlokas of Sanks̀epa rāmāyaṇa. Peter Scharf at Brown University has analysis of complete rāmopākhyāna (a story of Rāma from Mahābhārata). Many Geeta Press books have analysis of various texts such as mahabharat, ramayana, geeta, upanishadas etc. So a combination of these texts may be used to evaluate the morph analysers to ensure that the corerct analysis are not missed out, and these answers are produced with hish frequency.