Anusaaraka: An Accessor cum Machine Translator

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

Hindi: Official Language

English: Secondary Official Language

Law, Constitution: in English!

22 official languages at State Level

Only 10% population can 'understand' English

Akshar Bharati Group:
Hindi-Telugu MT: 1985-1990
Kannada-Hindi Anusaaraka: 1990-1994
Telugu, Marathi, Punjabi, Bengali-Hindi Anusaaraka : 1995-1998
English-Hindi Anusaaraka: 1998-

Sanskrit-Hindi Anusaaraka: 2006-

Several Groups in India working on MT
Around 12 prime Institutes

5 Consortia mode projects on NLP:
IL-IL MT
English-IL MT
Sanskrit-Hindi MT
CLIR
OCR

Around 400 researchers working in NLP

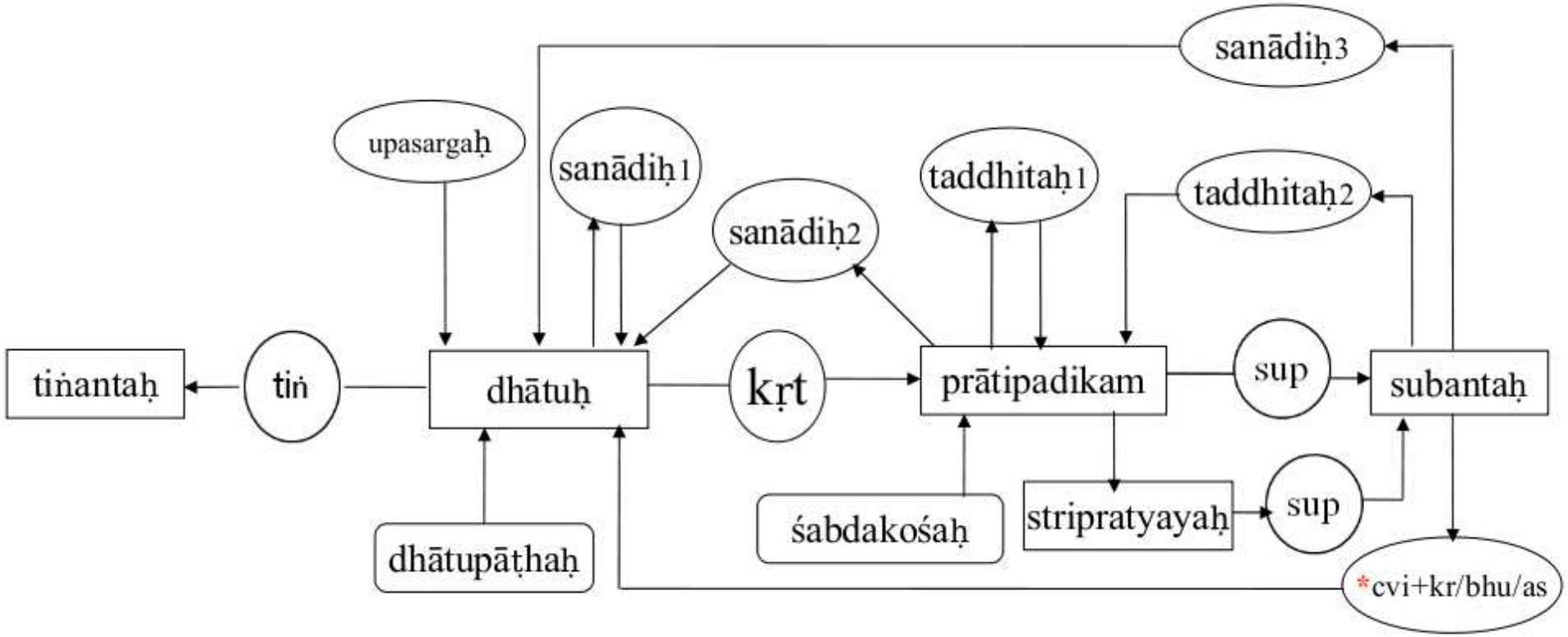
Major event every year: ICON (December)

Association with Apertium group: Since 2007

Mainly concentrating on Morph Analysers

Of late, we have decided to use Apertium pipeline in English-Hindi Anusaaraka for MT.

Word Formation in Sanskrit: A glimpse of complexity



Anusaaraka: An Accessor CUM Machine Translation

1 What is an accessor?

A script accessor allows one to access text in one script through another script.

GIST terminals available in India, is an example of script accessor.

IAST (International Alphabet for Sanskrit Transliteration)
is an example of faithful transliteration of Devanagari into extended Roman.

कृष्ण = क् ऋ ष् ण् अ

k ṛ ṣ ṇ a (IAST)

k q R N a (WX notation)

Salient Features

- Faithfull representation
- Reversibility
- No loss of information
- Text in other script is accessible with little extra training

What is a Language accessor?

Language Accessor tries to generalise and apply this philosophy to the problem of language conversion which is several order more complex than that of script conversion.

Urdu-Hindi Language Accessor

Urdu-Hindi: Same language written in two different scripts.
The divergence is at Script level, and not at the language level!
Urdu does not normally represent short vowels. Hindi requires it.
Thus a Urdu Consonant may correspond to any one of the following Hindi patterns:

C / Ca / Ci / Cu

Further if this C is a semi-vowel, it may correspond in addition to a stand-alone vowel (o/au in case of 'vaav'; e/ai in case of 'ye')

ہوا	ہ	و	ا	हवा/हुवा
	ह	व	अ	missing vowel
اور	ا	و	ر	और/ओर
	अ	व	र	semi vowel as a vowel
کیا	ک	ی	ا	क्या/किया
	क	य	अ	Halanta or a missing vowel

Learning of a Urdu Script involves:

- Learning of alphabet, their shapes, conjugate shapes
- Disambiguating the consonant by providing appropriate vowels/ etc.

Anusaaraka reduces the burden.
Machine takes the load of alphabet: the shapes of letters, their conjugate shapes etc.
Human being shares the load of disambiguation in the context.

Example of Urdu-Hindi Language Accessor

Share the load between man and machine

machine is good at rote memory + logic
man is good at World Knowledge, Common sense, Cultural Knowledge, Domain Knowledge,...

However there is a tight coupling between the two loads.

Anusaaraka Guide lines for developing MT systems:

- Make complete information available to the user
  but, do not clutter the scene.
- Seperate the resources
  that can be made, in principle, reliable
  from those that are, inherently unreliable.
  mention the degree of reliability.
- provide alternative means to get the information
- Do not reinvent the wheel
- Use existing resources and tools

How do we achieve this?

- Make complete information available to the user but, do not clutter the scene.
  —— Ensure Substitutivity and Reversibility
  —— Hiding Mechanism/User Interface
- Seperate the resources that are, in principle, reliable from those that are, inherently unreliable. Mention the degree of reliability explicitly.
  —— Run various modules in parallel, and show the output at various levels in descending order of reliability.
- provide alternative means to get the information
  —— Provide online help,
  —— Develop Human readable algorithms first and then implement whatever is possible with today's technology and resources.

- Do not reinvent the wheel
  —— Resort to GPL
- Use existing resources and tools
  — Develop suitable interfaces for 'plug and test' of different tools,
  — In case several tools are available for a particular task, use voting to select the best output.

Software Architecture Implications:

- Substitute the "Word Formulae" first
- Develop a Graphical User Interface(GUI) to present the right amount of information at right time.
- Develop an interface to allow 'plug and test' different resources
- Develop a module to select the best of K outputs

Language Accessor: Fidelity

Languages code information only Partially
Languages vary in their conventions w.r.t. the information coding

This leads to incommensurability among languages

Divergence is at various levels:
Word Level, Sentence Level, Discourse Level, ...

How does Anusaaraka guarantees Fidelity in spite of incommensurability?

Anusaaraka resorts to "Word Formula"
A practical implementation of a concept from Indian Grammatical Tradition: **Pravṛtti - nimmita**:
(The reason behind the use of a word in a particular sense)

Word Level Incommensurity

Labeling and Packaging of concepts may vary

(All examples are for English-Hindi pair)

Lexical Gap	Technical_words	–
	ly: adj -> adv	–
	Determiners	–
one-one	I	*maiṁ*
many-one	he,she,it	*vaha*
one-many	uncle	*māmā, cācā, tāū, phūphā, ...*

Overlapping regions:

play: *bajānā, khelanā, abhinaya karanā*
light: *halakā, prakāśa, prakāśita karanā*

Anusaaraka Solution:

- Lexical Gap
  - Concept word
    Borrow the word till a Target language equivalent is coined.
  - Function word
    Use special symbol to indicate special function.

Learning component: Load on the user

- Many-one English: He/She/It/that
  Hindi : *vaha*

  He: *vaha*{masc.}
  She: *vaha*{fem.}
  It: *vaha*{neut.}
  that: *vaha*{adj.}
  Learning component: semantics of special notation

- One-Many
  uncle -> *māmā, cācā, tāū, phūphā*, ...

  uncle -> *māmā^cācā‘*
  Learning component: semantics of special symbol

- Ovelapping Regions
  English: He played well.
  Hindi: Depending on the context, translation changes.

  cricket -> *khelanā*
  violin -> *bajānā*
  in drama -> *abbhinaya karanā*

  Develop a śabdasutra (Word Formula)

He	played	well
vaha{*masc.*}	*khelanā/bajānā‘+*{*ed*}	*acchā*

What is a *śabdasūtra*?

- a formula (concise way of representation)

  This is used in the anusaaraka output

- a thread connecting various senses of the given word

  This forms the learning component of the user training.

Example of śabdasūtra:

Dictionary Entries for the word 'light':

"light","A", "1. *halakā*"
This suitcase is light and good.

"light","N","1. *prakāśa*"
I could see a light in the room.

"light","V","1. *jalānā*"
Mohan lighted the match-stick.

–"2. *sulagānā*"

Mohan lighted the cigarette.

–"3. *prakāśita karanā*"
The torch lighted the way for him.

–"4. *prasanna honā*"
His face lighted up when he heard the news.

Noun: prakāśa
activity: prakāśita karanā
Similarity with the activity: Mohan lighted the cigarette.
Similarity with the result: Mohan lighted his 4-celled torch before entering the corridor.
activity of the instrument: The torch lighted the way for him.

Metaphoric use: prasanna honā

underlying thread:
prakāśa -> prakāśita karanā

sūtra :

prakāśa[ke yogya karanā] / halakā

Word: daṇḍa (of Sanskrit)

1. Stick
2. Staff of a banner
3. Handle of a sauce pan
4. A churning stick
5. A pole as a measure of length
6. A form of millitary array
7. A line
8. A staff as a symbol of power
9. Punishment
10. Control over the speech etc.

Word: daṇḍa (of Sanskrit)

1. Stick (Core meaning)
2. Staff of a banner (Similarity with shape)
3. Handle of a sauce pan (Similarity with shape)
4. A churning stick (Similarity with shape)
5. A pole as a measure of length (property of the object)
6. A form of millitary array (Similarity with shape)
7. A line (Similarity with shape)
8. A staff as a symbol of power (Use of the basic object / associated action)
9. Punishment (result of the associated action)
10. Control over the speech etc. (Similarity with the result)

Word Formula for Sanskrit word 'daṇḍa' through English

daṇḍa = Stick -> Similarity in Shape

-> Property of an object (it has length)

-> Result of the action associated (Punishment)

Or concisely as

daṇḍa = Stick{shape/instrument to punish/punishment}

Similar concept in Westerm Linguistics: Monoism

"A monosemic approach, on the other hand, posits only one abstract sense for a word, and the various contextual meanings or readings that the word receives are predictable from features of the context in which the word occurs" (Saxton: A Monosemic Semantics of Just: 1994)

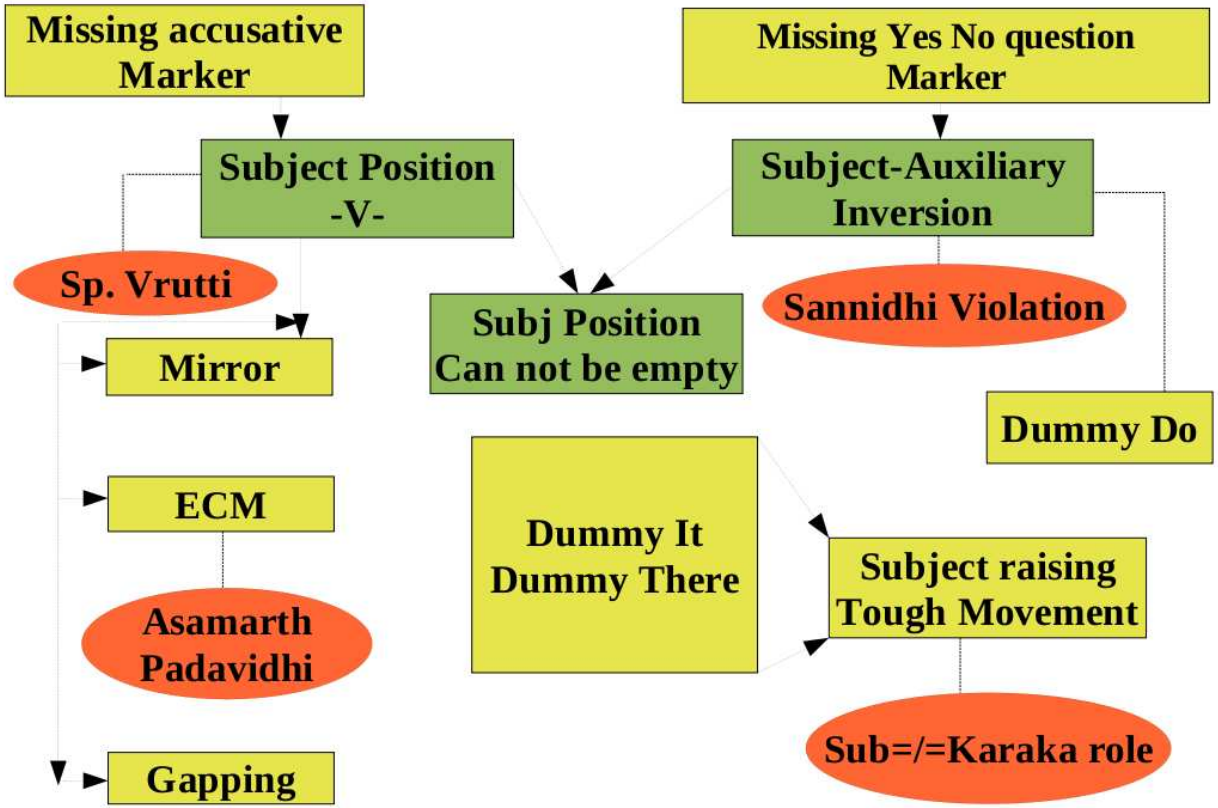
For example, Saxton gives various senses of *just* as below.

- I live *just* around the corner. (in the sense of immediacy)
- I *just* got home from school. (in the sense of recently)
- Tina looks *just* like her mother. (in the sense of exactness)
- Your beet salad is *just* delicious. (in the sense of very)
- May be it is *just* hormonal. (in the sense of only)
- I can *just* squeeze through that fence. (in the sense of barely)

Saxton, by postulating the "core" meaning of *just* as *without deviating*, extracts the appropriate meanings from the features in the context.

English-Hindi Learning Component

- Śabdasūtra Notation,
- Śabdasūtras for highly polysemous words
- Few hundred function words
- English-Hindi contrastive grammar


Missing accusative Marker
Missing Yes No question Marker
Subject Position -V-
Subject-Auxiliary Inversion
Sp. Vrutti
Sannidhi Violation
Subj Position Can not be empty
Mirror
Dummy Do
ECM
Dummy It Dummy There
Subject raising Tough Movement
Asamarth Padavidhi
Sub=/=Karaka role
Gapping

Reducing the load further:

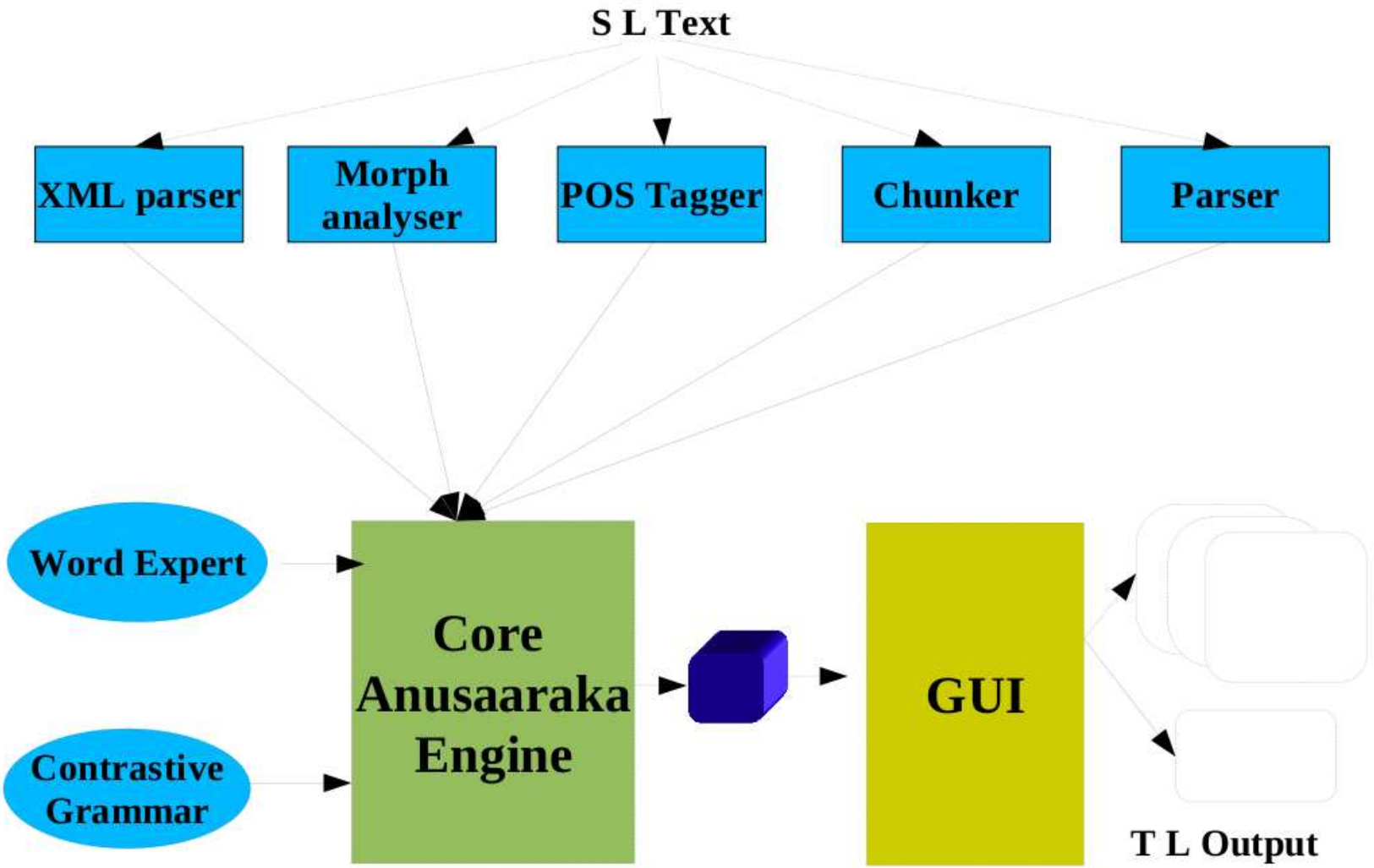
Towards MT ...

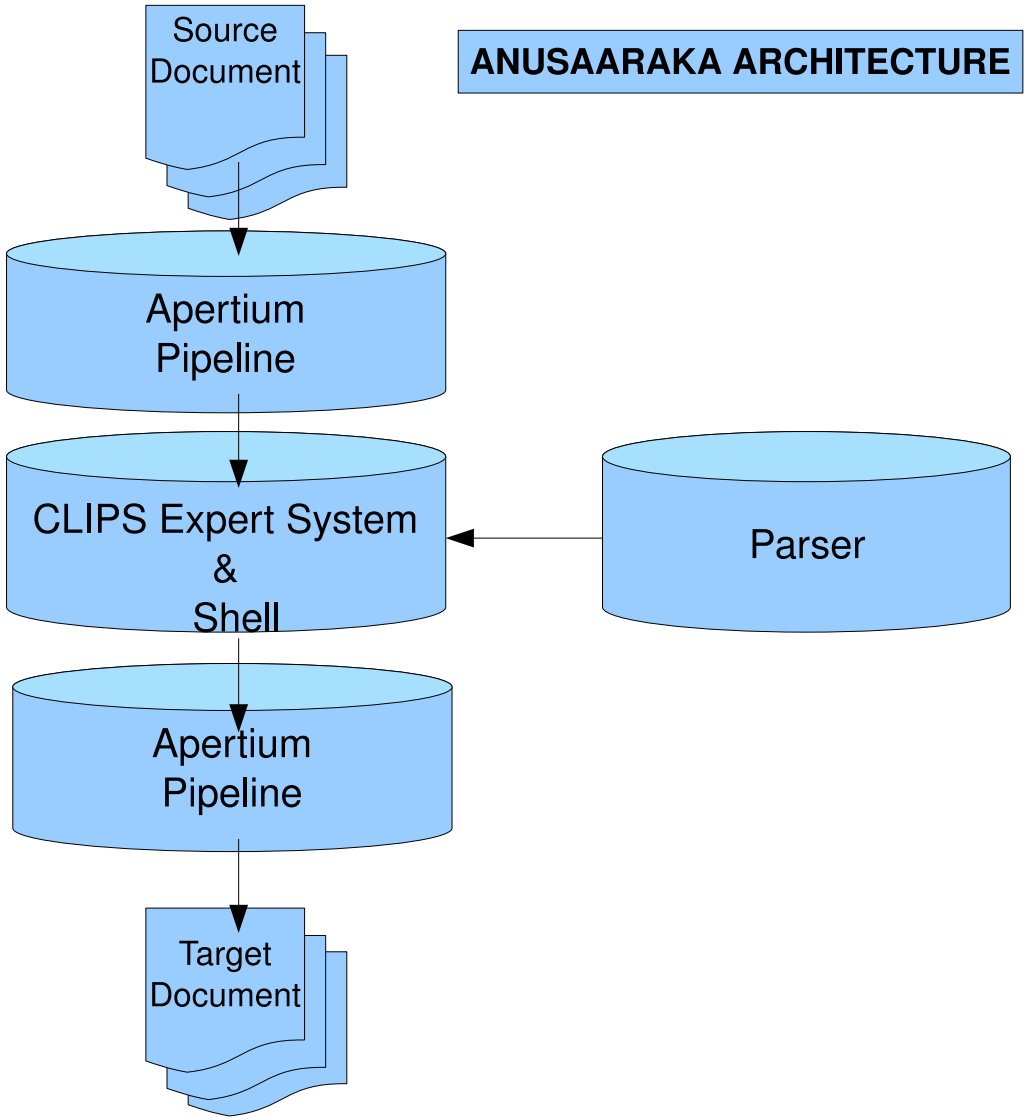
If the languages are close enough, then load on user is very less.

However, if the incompatibility increases, the load on user also increases.

Can we further reduce the load on user?

Incommensurability	Solution
2D Drawing of a 3D structure	Provide different views of the 3-D structure Plan (top view), Elevation (front view), Side view
Among Languages	Provide glosses from different views Word Sense Disambiguation, movement of function words, Descrambling


S L Text
XML parser
Morph analyser
POS Tagger
Chunker
Parser
Word Expert
Contrastive Grammar
Core Anusaaraka Engine
GUI
T L Output


Source Document
ANUSAARAKA ARCHITECTURE
Apertium Pipeline
CLIPS Expert System & Shell
Parser
Apertium Pipeline
Target Document

CLIPS(C Language Integrated Production System)
Developed by NASA

The basic elements of CLIPS, as of any expert system, are

- Fact-list: global memory for data.
- knowledge-base:consists of user specified rules
- Inference engine:controls overall execution.

Provides full freedom to a language expert/developer to specify the rules

No restriction on the format

TASKS:

- Pāṇinian interface to the Parser
- WSD rules
- Deciding WORD(quasi-compound) boundary
- Deciding the TL WORD order
- Generating WORD in the TL

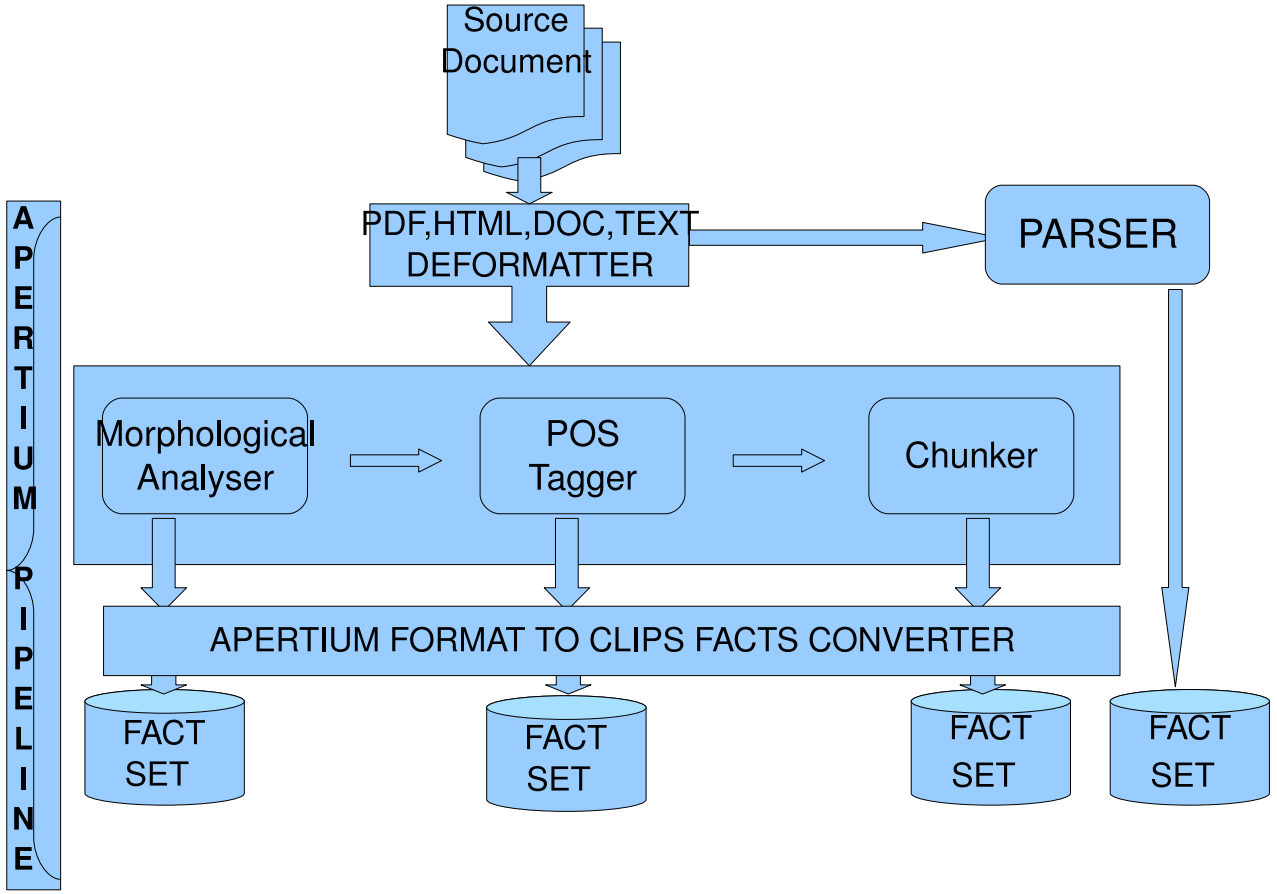
Language expert needs Freedom and full power of the programming language to express his algorithm

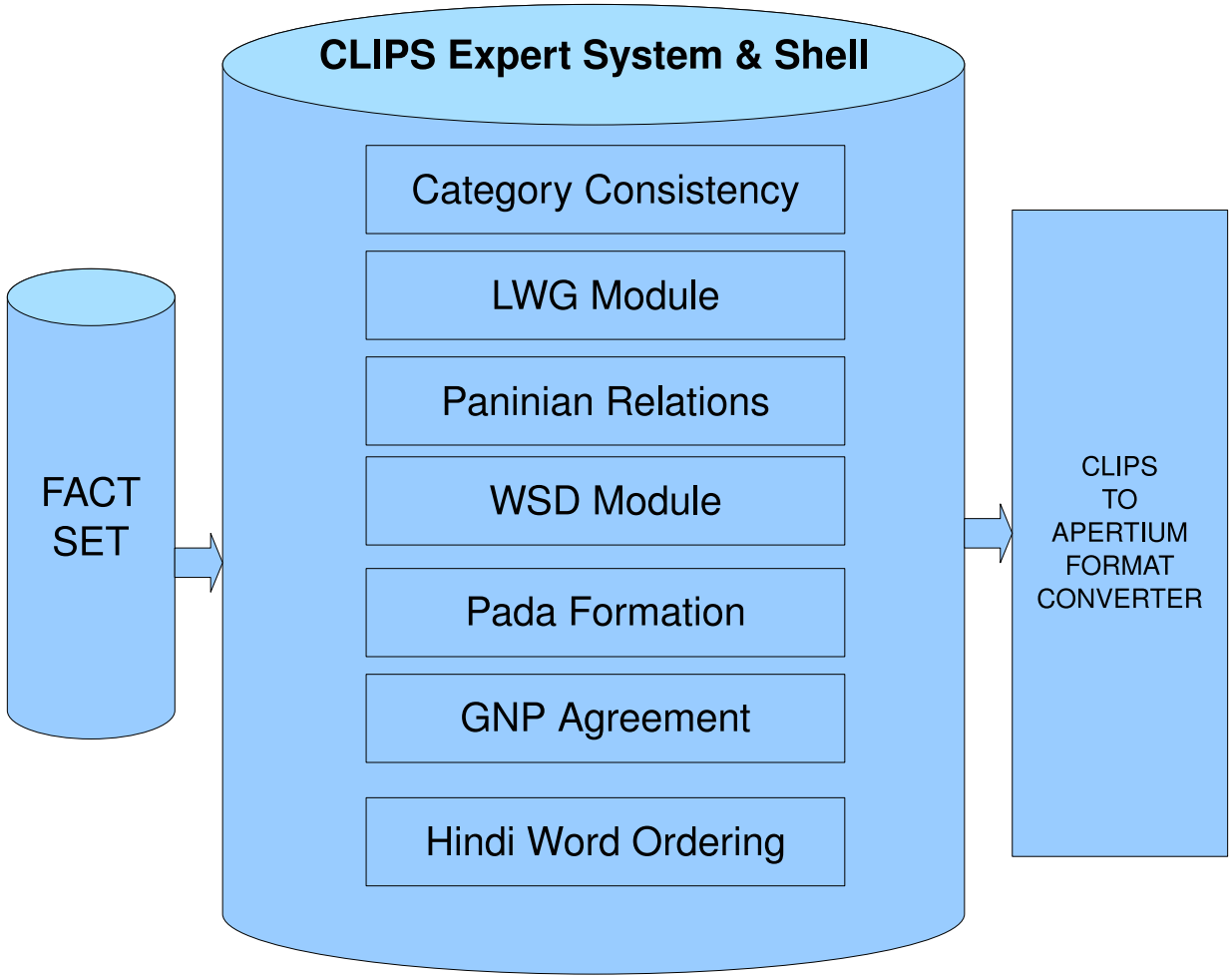
Well tested and stable algorithms then can be taken out of the CLIPS environment from efficiency point of view.

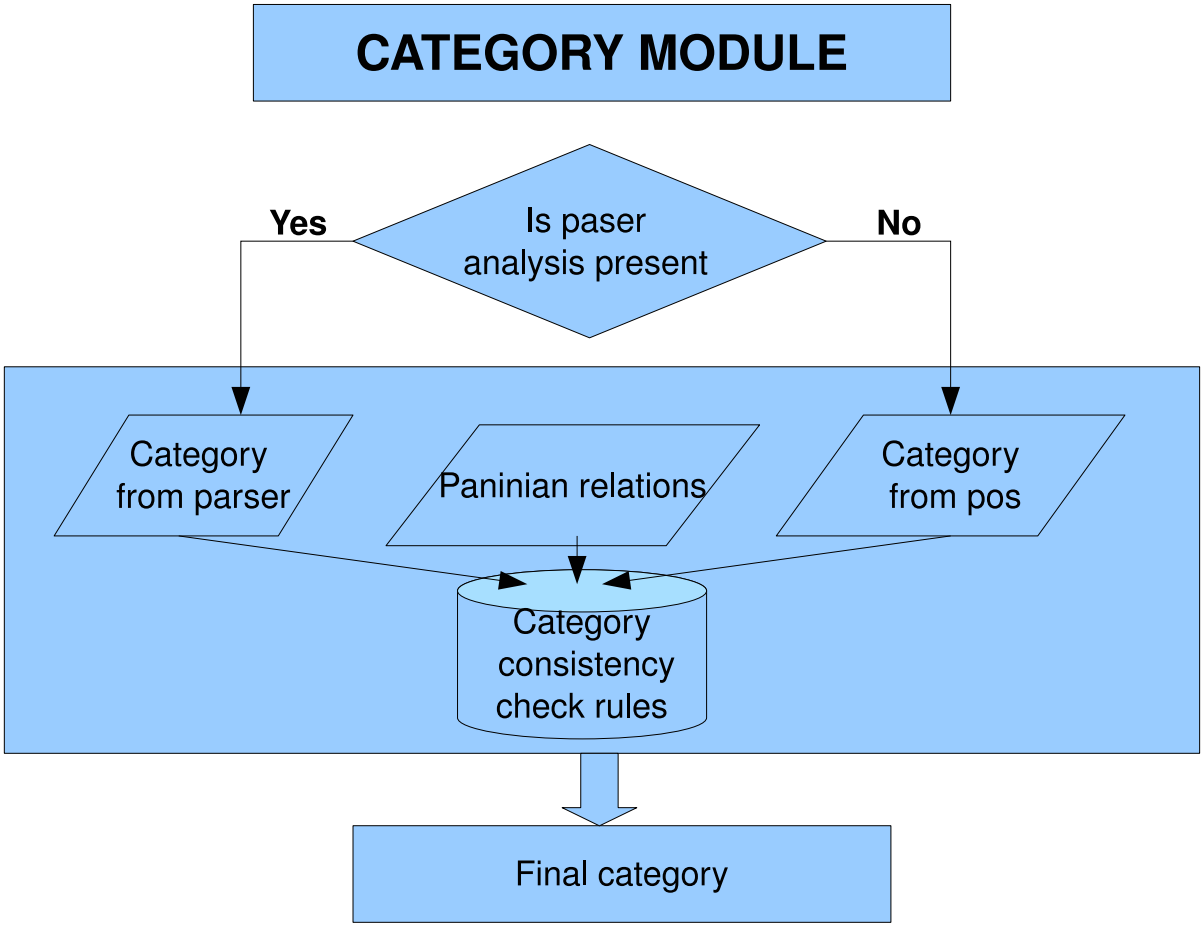
WSD: Major and Voluminous task

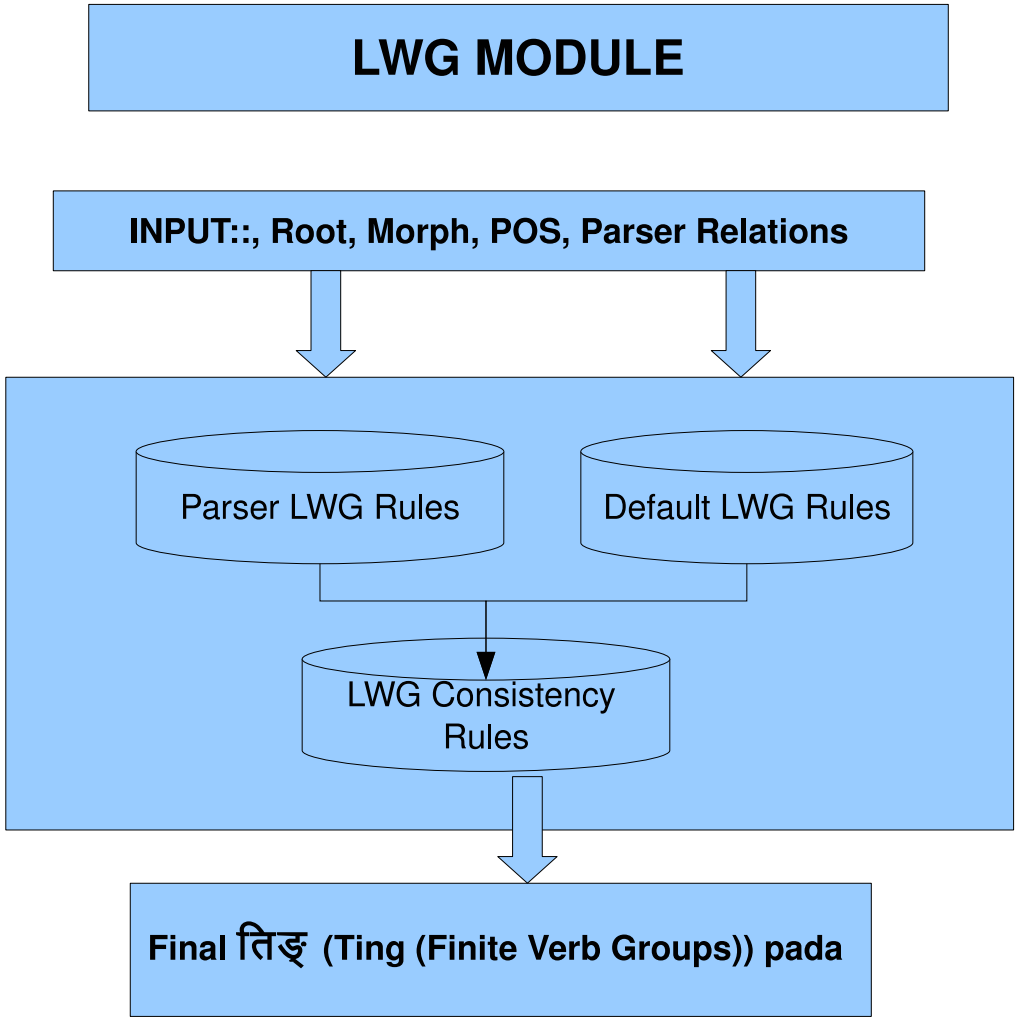
Need for Mass involvement
(Oxford dictionary, Wikipedia)

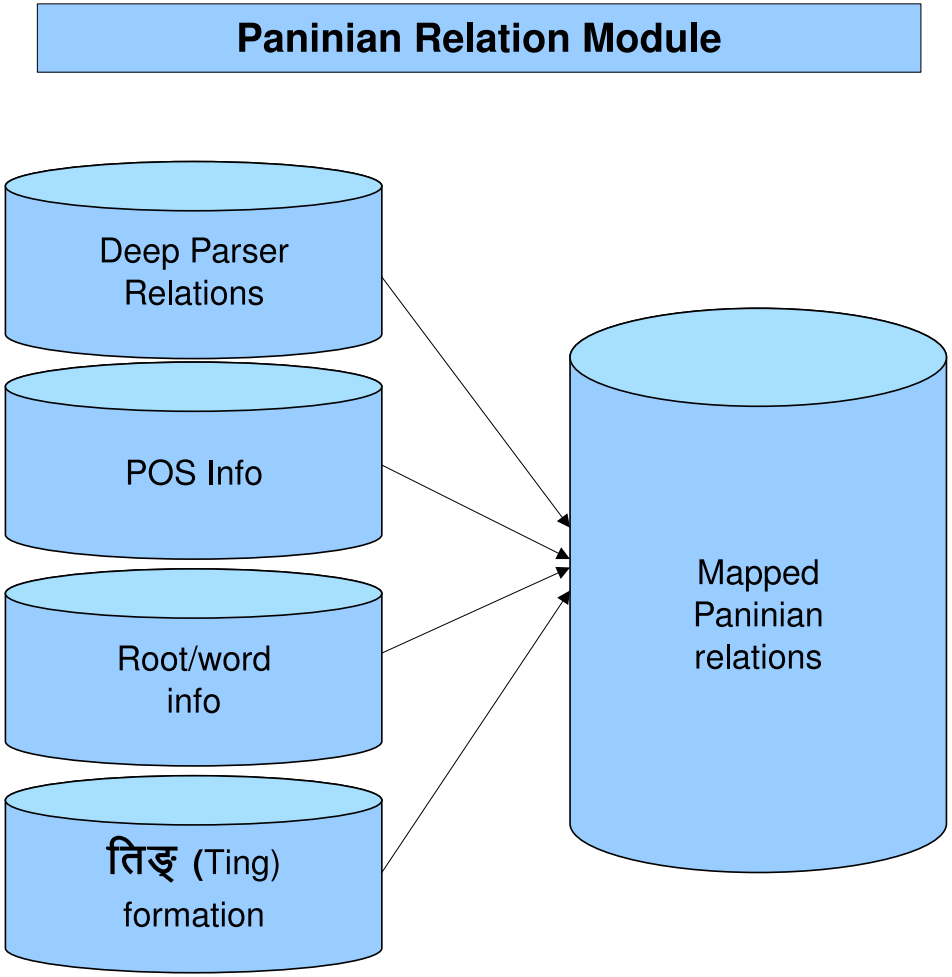
1 word per day == > 200 man years for 70,000 words

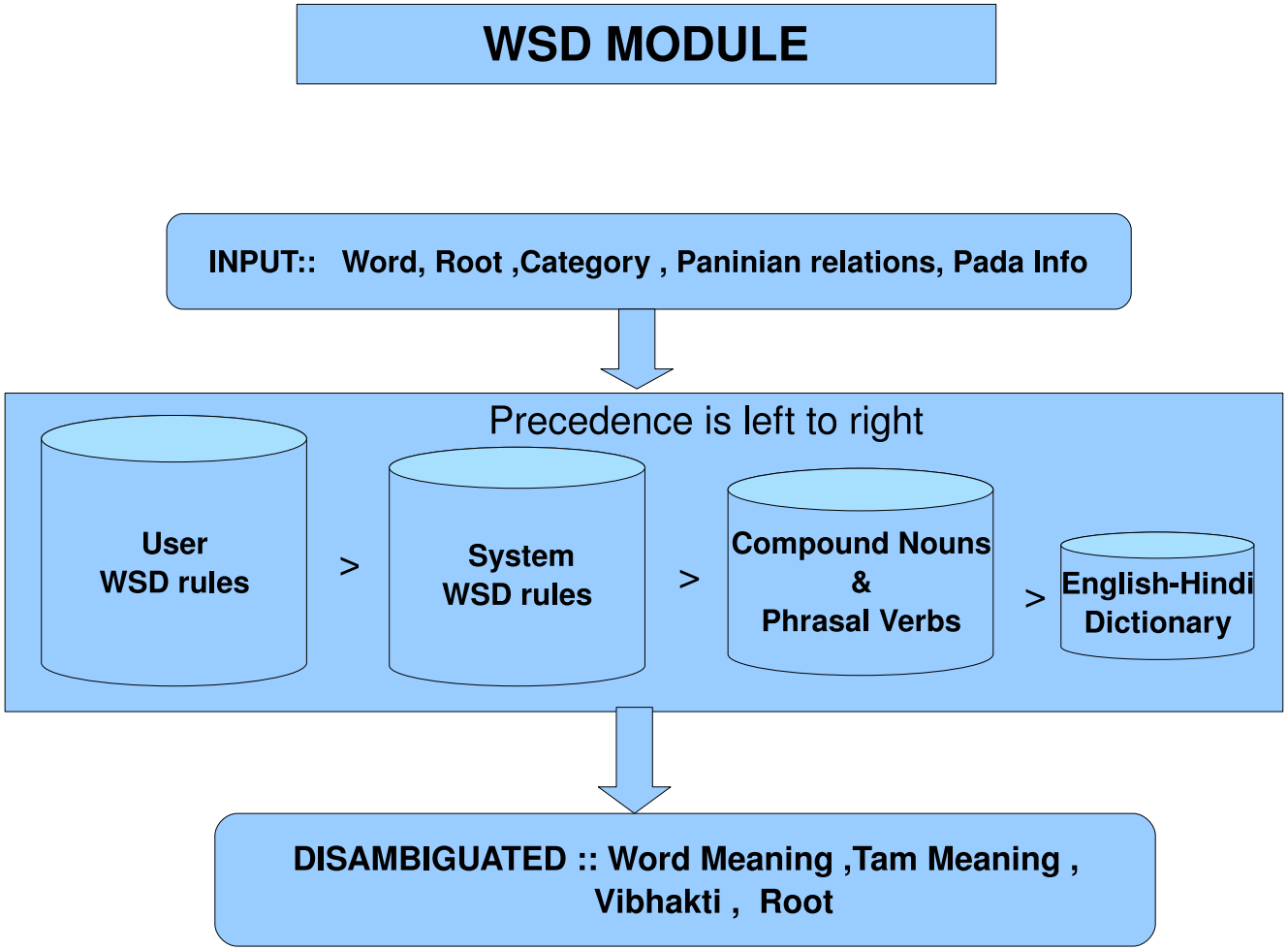

Source Document
PDF,HTML,DOC,TEXT DEFORMATTER
PARSER
APERTIUM PIPELINE
Morphological Analyser
POS Tagger
Chunker
APERTIUM FORMAT TO CLIPS FACTS CONVERTER
FACT SET
FACT SET
FACT SET
FACT SET

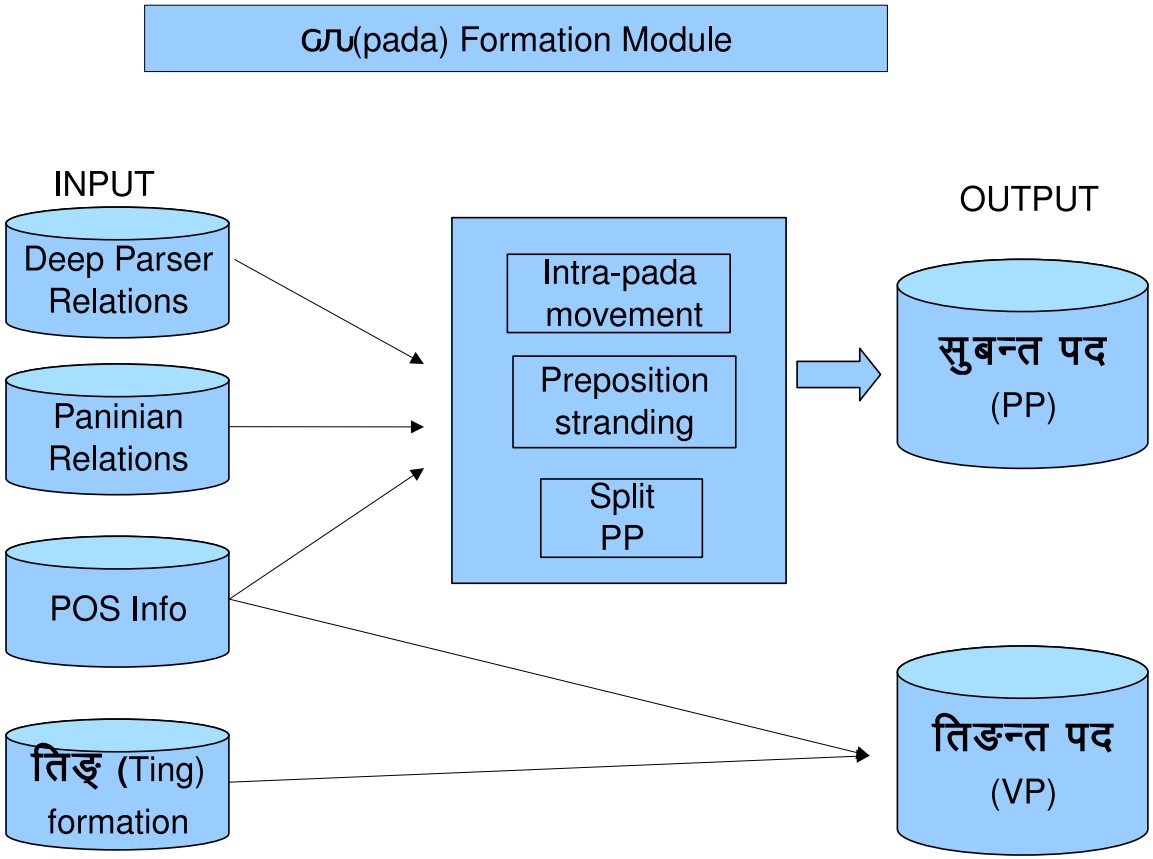

CLIPS Expert System & Shell
Category Consistency
LWG Module
Paninian Relations
WSD Module
Pada Formation
GNP Agreement
Hindi Word Ordering
FACT SET
CLIPS TO APERTIUM FORMAT CONVERTER

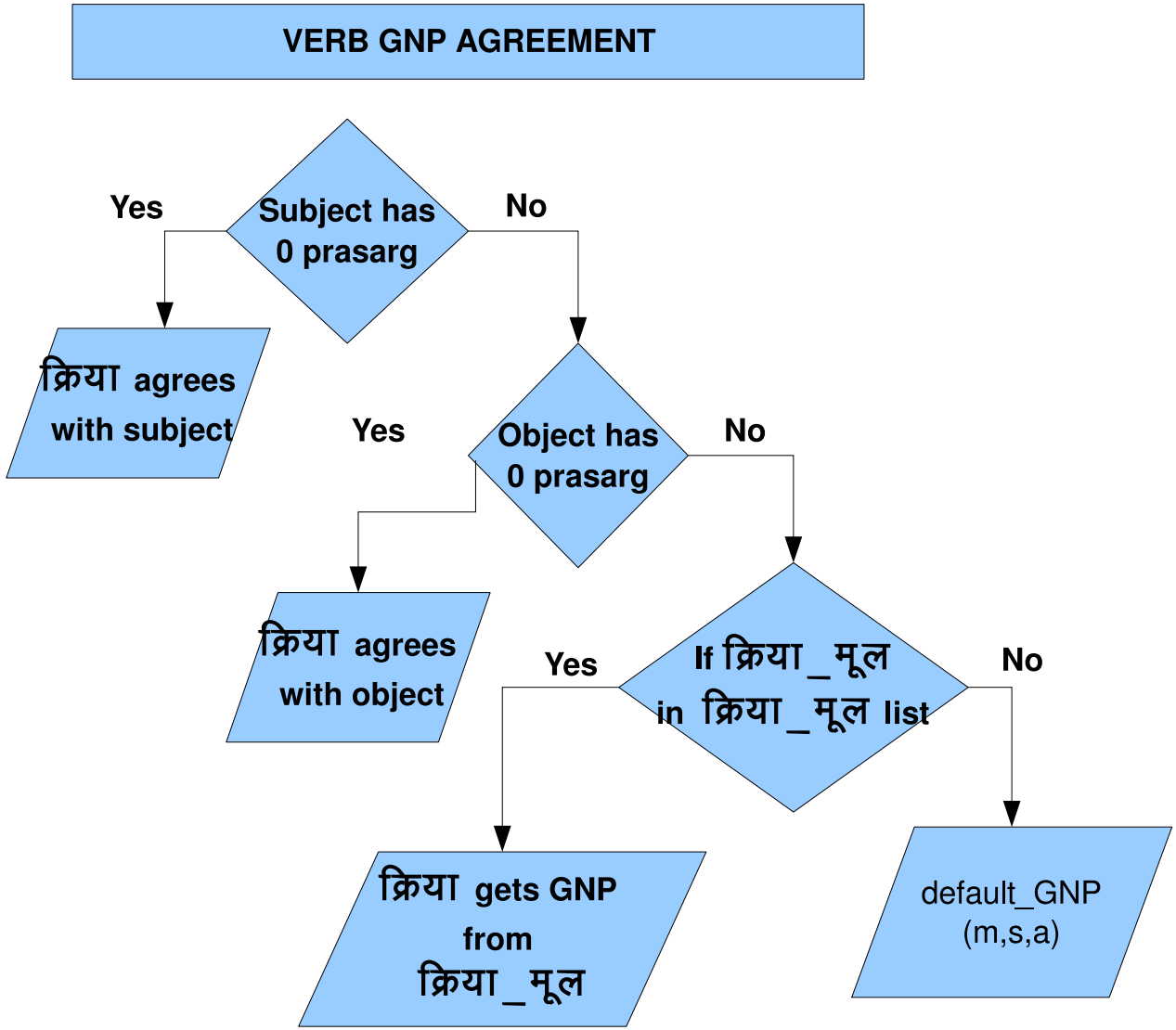

CATEGORY MODULE
Yes
Is paser analysis present
No
Category from parser
Paninian relations
Category from pos
Category consistency check rules
Final category

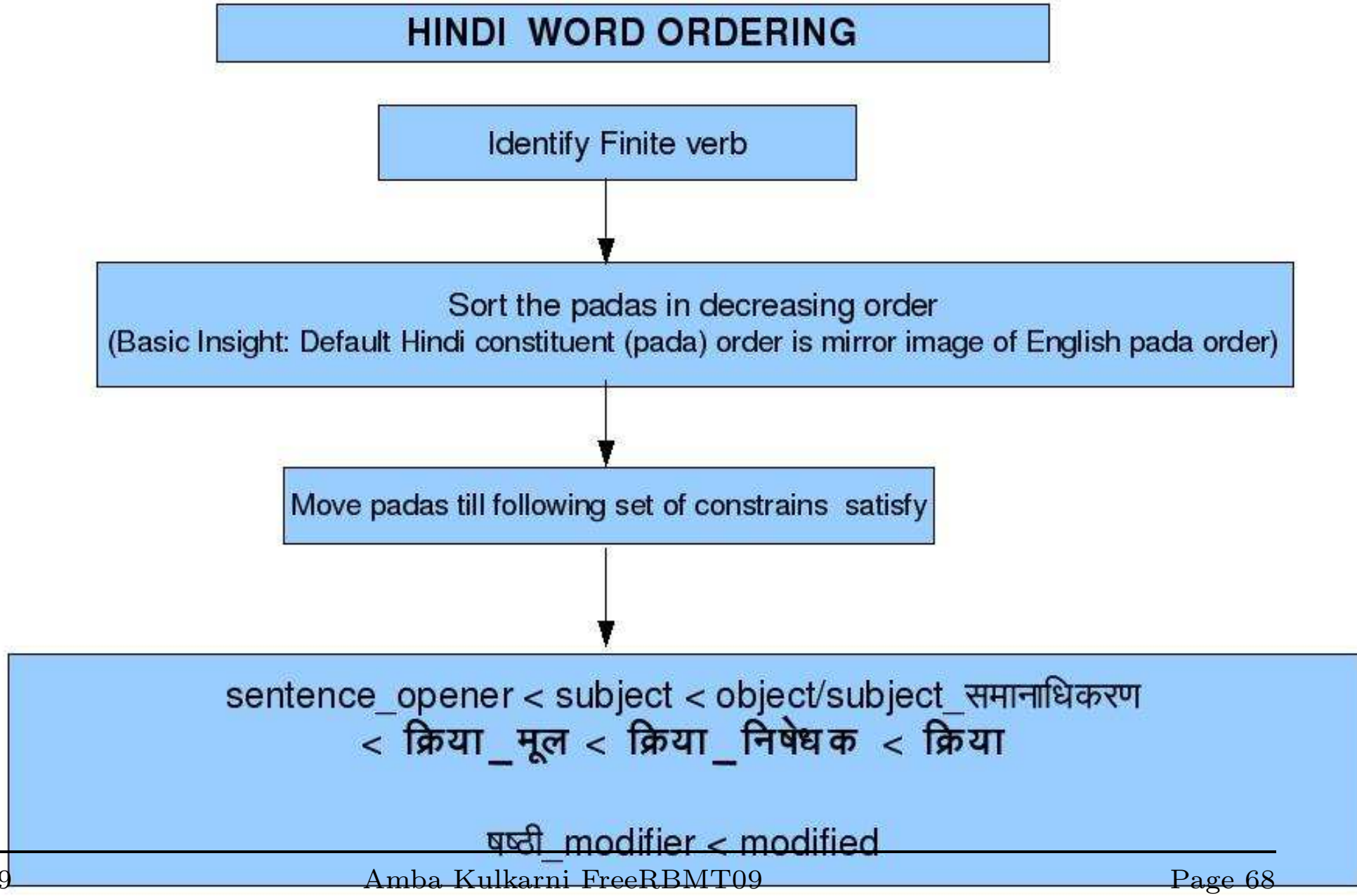

LWG MODULE
INPUT::, Root, Morph, POS, Parser Relations
Parser LWG Rules
Default LWG Rules
LWG Consistency Rules
Final तिङ् (Ting (Finite Verb Groups)) pada

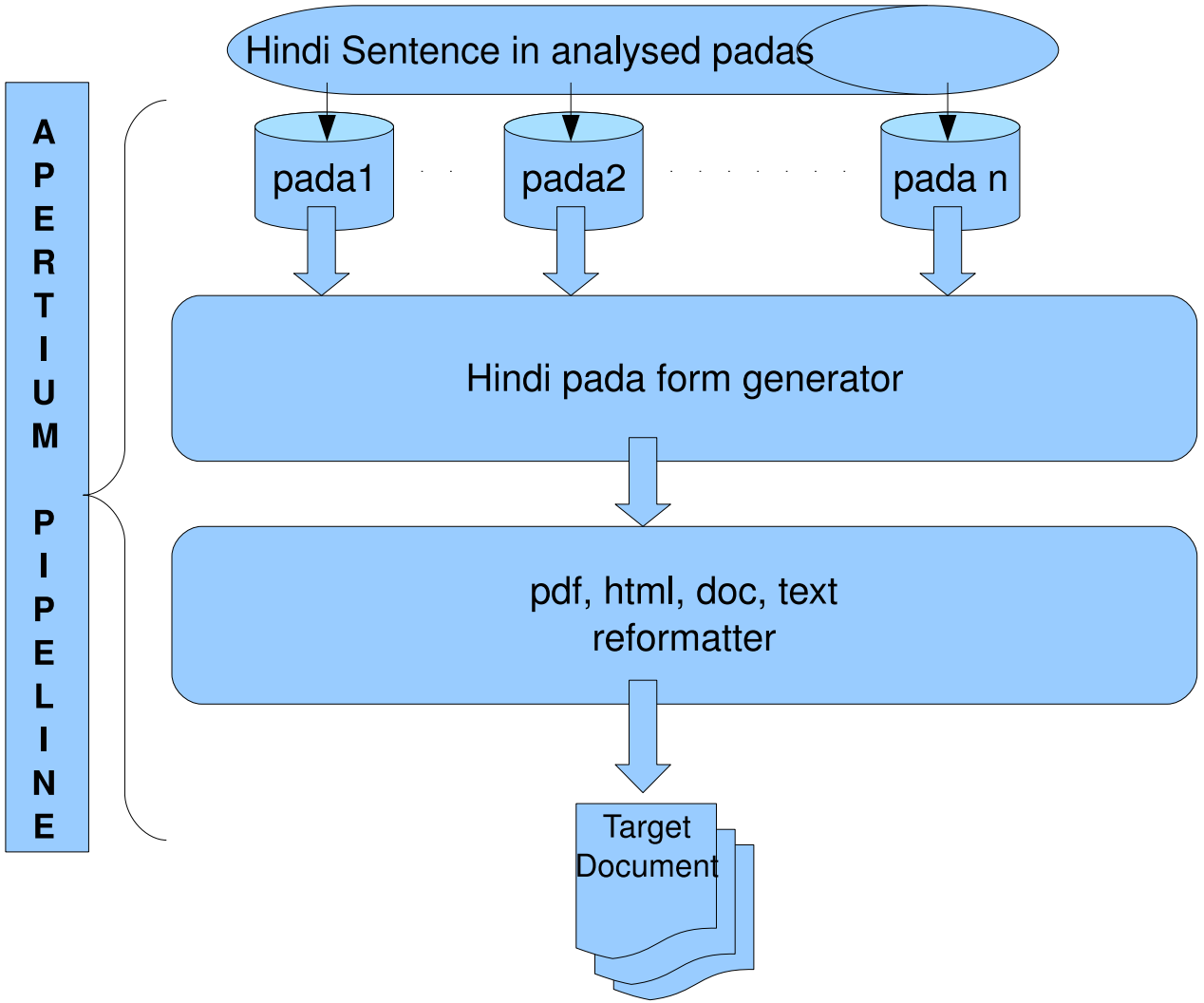

Paninian Relation Module
Deep Parser Relations
POS Info
Root/word info
तिङ् (Ting) formation
Mapped Paninian relations


WSD MODULE
INPUT:: Word, Root ,Category , Paninian relations, Pada Info
Precedence is left to right
User
WSD rules
>
System
WSD rules
>
Compound Nouns
&
Phrasal Verbs
>
English-Hindi
Dictionary
DISAMBIGUATED :: Word Meaning ,Tam Meaning ,
Vibhakti , Root


पद(pada) Formation Module
INPUT
Deep Parser Relations
Paninian Relations
POS Info
तिङ् (Ting) formation
Intra-pada movement
Preposition stranding
Split PP
OUTPUT
सुबन्त पद
(PP)
तिङन्त पद
(VP)


VERB GNP AGREEMENT
Subject has 0 prasarg
Yes
No
क्रिया agrees with subject
Object has 0 prasarg
Yes
No
क्रिया agrees with object
If क्रिया_मूल in क्रिया_मूल list
Yes
No
क्रिया gets GNP from क्रिया_मूल
default_GNP (m,s,a)


HINDI WORD ORDERING
Identify Finite verb
Sort the padas in decreasing order
(Basic Insight: Default Hindi constituent (pada) order is mirror image of English pada order)
Move padas till following set of constrains satisfy
sentence_opener < subject < object/subject_समानाधिकरण
< क्रिया_मूल < क्रिया_निषेधक < क्रिया
षष्ठी_modifier < modified


APERTIUM PIPELINE
Hindi Sentence in analysed padas
pada1
pada2
pada n
Hindi pada form generator
pdf, html, doc, text
reformatter
Target
Document

Why Better?

- Anusaaraka is an Incremental Robust Machine Translation System
- Anusaaraka makes the whole process of MT 'completely transparent' to the user.

Anusaaraka is

- An Incremental Machine Translation
  - layered output
  - Successive layers more and more close to MT.
- Robust
  - Clear cut seperation of the resources that are in principle reliable from those that involve probabilistic component.
  - Graceful degradation In case of failures it produces a 'rough' translation.

    It is not 'rough[a]' in the sense that is not accurate or precise, but in the sense that it requires some human effort to

[a]compare with 'rough journey' where you are taken to the destination, but the journey is not comfortable.

understand the text.

Anusaaraka: User Transparency

The whole process of Machine Translation is transparent to even a layman.

Anybody with an aptitude for 'language analysis' can contribute to the development of a Machine Translation system even without any exposure to the formal linguistic training.

Feature	Typical MT System	Anusaaraka
Goal	Aims to produce Natural translation. In case of failures produce 'rough' [a] translation.	To provide access to the Source Language text. The output may be 'rough' [b]
Unit of input	Currently, independent sentences	a complete XML document
System Components	Morph analysers, POS taggers, Parsers, Sense disambiguation modules, Generator	Same as in MT Plus Anusaaraka User Interface

[a] what is 'rough' is not well defined.

[b] Here rough is as in the sense of 'rough journey' where you are taken to the destination, but the journey is not comfortable.

Sequence of Operations	Outputs are cascaded. So errors too get cascaded	The basic tasks are processed independently. Price paid: Duplication of effort
Transparency	Processing is not transparent to the end-user	Processing is transparent to the end-user
Access	User has access only to the final output	User has access to the output at each level
Guidelines for Linguists	No specific guidelines	First write an algorithm for 'Human beings' and not necessarily for 'computers'!
Principle	Ad-hoc	"Information Dynamics"

Approaches	1. EBMT 2. Rule based 3. Statistical 4. Hybrid	Eclectic: Choose the best of each of these approaches. Use best of the available resources under GPL.

Consequences	1. Later modules are affected by the errors of the previous modules 2. Rough is not well defined. Hence users may get mislead. 3. User can not participate in the development process 4. Linguists end up in reinventing the wheel again and again.	1. Parallel processing ensures that different modules do not interfere. 2. Well defined 'Roughness'. Theoretically no chances of user getting mislead. 3. User can participate in the development activity. 4. Linguist prepares data only once.