Information Coding in a language: Some insights from Pāṇinian Grammar

Akshar Bharati and Amba Kulkarni

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

Abstract. The knowledge of how a language codes information, how much information it codes and where it codes is very crucial for a computational linguist working in the area of Natural Language Processing and in particular Machine Translation.

Pāṇini has given utmost importance to the information coding in a language string. This is evident from the use of the same marker *Ṇ* twice making the pratyāhāras apparently ambiguous.

In support of our claim that Pāṇini had information coding at the back of his mind while writing the grammar for Sanskrit in the form of *Aṣṭādhyāyī*, we discuss as representatives the 3 sūtras: *anabhihite* (3.1.1), *svatantraḥ kartā* (1.4.54) and *samānakartṛkayoḥ pūrvakāle* (3.4.21). These 3 sutras precisely point out where the information is coded, how much information is coded and the manner in which the information is coded in Sanskrit.

Key Words: Information coding in Pāṇini, pratyāhāra, *kartā*, agent, implicit coding

1 Introduction

India has around 2500 years of rich heritage in linguistic studies. Out of the six *vedāṅga*s (fields of studies necessary to study the vedas) viz. *śikṣā, vyākaraṇa, chandas, nirukta, jyotiṣa* and *kalpa*, the first four are concerned with language studies. *Śikṣā* deals with pronunciation, *vyākaraṇa* with grammatical aspects, *chandas* with prosody and *nirukta* with etymology. Among all these the importance of *vyākaraṇa* is long recognised and is evident from the enormous literature on vyākaraṇa. It has a major role to play in understanding how a language communicates thoughts from one human being to the other.

Pāṇini consolidated all the earlier grammars for Sanskrit and presented a concise and almost exhaustive descriptive coverage of the then prevalant Sanskrit language. This grammar is in the form of aphorisms – around 4000 divided

into 8 chapters of 4 sections each. "Pāṇini's grammar is universally admired for its insightful analysis of Sanskrit"(Kiparsky, 2009). In spite of being a grammar basically written for Sanskrit, it provides many ingenious concepts for language analysis, which are universal in nature.

"The goal of Pāṇinian enterprise is to construct a theory of human communication using natural language" (Bharati,1995; p 59). No doubt, Pāṇinian Grammar (PG), as any other grammar formalism would give, gives a very good theory to identify the relations among words in a sentence but importance of PG lies in the minute observations of Pāṇini regarding the information coding in a language.

In the next section we establish our claim that Pāṇini was aware of the various means a language engages to code an information. This is evident from the way he analysed Sanskrit and also from the way he framed the sūtras. We cite an example from the *Māheśvarasūtra*s. The third section discusses three sūtras from the *Aṣṭādhyāyī* and show how they answer important questions related to information coding.

2 Various means a language engages to encode information: An illustration from Māheśvarasūtras

The *Māheśvarasūtras* form an integral part of the *Aṣṭādhyāyī*. It consists of 14 sūtras. Each sūtra has one or more phonological segments terminated by a marker (*anubandha* or *it*). Pāṇini has used around 42 different subsets of phonemes in the *Aṣṭādhyāyī*. The *Māhesèvarasūtra*s are a linear arrangement of these 42 partially ordered sets (known as *pratyāhāra*s) with markers placed in between (at the end of each sūtra) indicating different set boundaries. The linear arrangement with markers helps one to obtain the 42 sets by a mechanical procedure thereby facilitating an easy memorisation of these sets. Kiparsky (1991) and Petersen (2004) have given respectively linguistic insight and mathematical proof of the optimality of the *Māhesèvarasūtras* with respect to the placement of the markers as well as the number of markers. Petersen has elegantly shown why the repetition of h in the sūtras is necessary and that the choice of h is optimal.

Pāṇini used the same consonant $\dot{N}$ as an anubandha at two different places in the *Māhesèvarasūtra*s. There has not been any satisfactory explanation of the reasons behind the repetition of Ṇ. Patañjali asks was there a dearth of phonemes that Pāṇini used same consonant twice introducing an un-necessary ambiguity? The commentaries by Patañjali and Bhartṛhari provided an important insight into the problem from information coding point of view. Let us look at the case in detail and see what the commentators have to say. Here are the first 6 *Māhesèvarsūtra*s with repeated $\dot{N}$.

a i u $\dot{N}$

$$\begin{array}{c} ṛ\ ḷ\ K \\ e\ o\ Ṅ \\ ai\ au\ C \\ h\ y\ v\ r\ T \\ l\ Ṇ \end{array}$$

This makes the pratyāhāra *aṆ* and *iṆ* ambiguous since the pratyāhāra *aṆ* may refer to $\{a\ i\ u\}$ or $\{a\ i\ u\ ṛ\ ḷ\ e\ o\ ai\ au\ h\ y\ v\ r\ l\}$, and the *iṆ* may refer to $\{i\ u\}$ or $\{i\ u\ ṛ\ ḷ\ e\ o\ ai\ au\ h\ y\ v\ r\ l\}$. Patañjali examines all the sūtras that use *aṆ* and *iṆ* and finally concludes that in each of these cases one can resolve the ambiguity. Bhartṛhari's commentary on the *Mahābhāṣya*, the *Dīpikā*, is worth mentioning. Bhartṛhari(Abhyankar, p.90) observes that *sāmarthya* (ability to convey a specific meaning), *prasiddhi* (frequency of usage), *liṅga* (indicator) and *lāghava* (economy) are the deciding factors for resolving the ambiguity arising because of the repetition of *Ṇ*.[1]

The *Aṣṭādhyayī* has five sūtras that use the pratyāhāra *aṆ*. They are

ḍhralope pūrvasya dīrgho'ṇaḥ (6.3.110)[2]
ke'ṇaḥ (*aṅgasya hrasvaḥ*) (7.4.13)[3]
aṇo'pragṛhyasyānunāsikaḥ (*vā*) (8.4.56)
uraṇ raparaḥ (1.1.50)
aṇudit savarṇasya cāpratyayaḥ (1.1.68)

In what follows we show how in each of these cases ambiguity can be resolved.

2.1 *Sāmarthya* (ability to convey proper meaning)

The first 3 cases viz.

ḍhralope pūrvasya dīrgho'ṇaḥ (6.3.110)
ke'ṇaḥ (*aṅgasya hrasvaḥ*) (7.4.13)
and
aṇo'pragṛhyasyānunāsikaḥ (*vā*) (8.4.56)

contain the words *hrasva*, *dīrgha* and *pragṛhya*. These terms refer only to vowels. In other words, there are no cases where they qualify any of the phonemes from the set $\{h\ y\ v\ r\ l\}$. Therefore Patañjali argues that if in these three sūtras *Ṇ* were to refer to the 2^{nd} *Ṇ* in the pratyāhāra sūtras, it would have been sufficient to use the pratyāhāra *aC* which refers to the set of vowels. Since the pratyāhāra

[1] *ayam ṇakāro dvir anubadhyate. atra prakaraṇe ṣaṭprakārāḥ upakṣiptaḥ – āsattiḥ vyāptiḥ sāmarthyam prasiddhir liṅgam lāghavam iti.*

[2] In references to the *Aṣṭādhyayī* numbers separated by periods are to chapter (textitadhyāya), part (*pāda*), and sūtra, respectively, for example, 6.3.110 indicates the 110^{th} sūtra in the 3^{rd} pāda of the 6^{th} adhyāya.

[3] Words in brackets are understood to recur from earlier sūtras. This recurrence is termed as *anuvṛtti*

aC is already in use and hence does not lead to the introduction of an additional pratyāhāra, economy (*lāghava*) would be achieved. In fact, additional economy would be achieved at sūtra level, he argues, because even *aC* would not have to be mentioned, being the default case. The fact that Pāṇini has mentioned *aṆ*, therefore implies that he meant the first *aṆ* and the pratyāhāra referring to the smaller set {*a i u*} and not the bigger one. Thus it is the words *hrasva*, *dīrgha*, and *pragṛhya* in the context which determine that the word *aṆ* conveys one meaning over the other. Bhartṛhari terms this as *sāmarthya*, the ability of a particular meaning to express itself (in a particular context).

2.2 *Prasiddhi* (frequency of usage)

In the next sūtra *ur aṇ raparaḥ* (1.1.50), the possibility of the 2nd *Ṇ* is ruled out on the basis of unavailibility of any example which involves bigger set {*a i u ṛ ḷ e o ai au h y v r l*}. Patañjali discusses two examples in his commentary and he points out that either the effect of the rule is nullified by another sūtra, or the application of this sūtra leads to redundancy in some other sūtra, which is undesirable. Hence he concludes that if Pāṇini meant the 2nd Ṇ, he could have used the smaller pratyāhāra *aC*. Since Pāṇini used *aṆ*, in the absence of any other clue for decision, Patañjali concludes that *Ṇ* in this sūtra is the 1st one and not the 2nd one (because in all the previous sūtras involving *aṆ*, it is the 1st *aṆ* that is being used). According to Bhartṛhari, it is the *prasiddhi* (frequency of usage) which is the deciding factor in this sūtra.

2.3 *Liṅga* (marker)

The 5th sūtra that uses *aṆ* is

aṇudit savarṇasya cāpratyayaḥ (1.1.68)

From this sūtra alone it is not obvious which *aṆ* is meant. There is another sūtra *ur ṛt* (7.4.7) which says *ṛ* becomes *ṛt*. The *t* in ṛt makes ṛ *tapara* which means that the *ṛ* represents only those sounds of its class that are of the same time, in accordance with the sūtra *taparas tatkālasya* (1.1.69), in exception to 1.1.68 which allows a vowel to refer to all sounds of its class. If the *Ṇ* in the pratyāhāra *aṆ* in 1.1.68 were the first *Ṇ*, it would not have been necessary to mark *ṛ* as *ṛt* in 7.4.7. The very presence of the sūtra 7.4.7 therefore indicates that *ṛ* is a member of the *aṆ* in 1.1.68, and hence the *Ṇ* in 1.1.68 is the 2nd *Ṇ*.

2.4 *Lāghava* (economy)

Finally in case of *iṆ*, it is observed that when Pāṇini wanted to mention the 1st *Ṇ*, only two phonemes *i* and *u* being involved, he used *yvoḥ* instead of *iṆaḥ*. In fact, *yvoḥ* = *y v o ḥ* involves 3.5 (=0.5 + 0.5 + 2 + 0.5) morae (*mātrā*s, the time measure of utterence of a phoneme) whereas *iṇaḥ* = *i ṇ a ḥ* involves 3 (= 1 +

0.5 + 1 + 0.5) mātrās. Thus in spite of prolixity (*gaurava*) of 0.5 mātrā, Pāṇini prefers *yvoḥ* over *iṇaḥ*, so that one always understands 2^{nd} *Ṇ* in all other cases, achieving *lāghava* (economy) in other cases.

2.5 Why repetition?

Patañjali at the end of the discussion on this topic in the *Mahābhāṣya* raises a valid question: was there a dearth of consonants that Pāṇini used the same phoneme twice? In response he warns

vyākhyānato viśeṣapratipattiḥ na hi sandehād alakṣaṇam

(If one has got a doubt, one should not jump to the conclusion that the sūtra is defective. One should seek additional information from the commentaries.)

At the surface level, by repeating Ṇ, no doubt an ambiguity is introduced. Pāṇini's aṣṭādhyāyī as several other Sanskrit texts do, does not carry any introduction or preface to his work explaining the purpose of his work, the methodology he used, etc. In the absence of any explanation by Pāṇini, on the repeatition of Ṇ, we are forced to conclude that **Pāṇini must be fully aware of the ambiguities a natural language has and also different sources of information such as sāmarthya, liṇga, lāghava, etc. available for disambiguation, and therefore in this particular case, might have allowed the repetition of the consonant.**

3 Pāṇini's subtle observations regarding Information coding in Sanskrit

Though a substantial part of the *Aṣṭādhyāyī* deals with the rules related to morphology, phonology and sandhi, an important section of it deals with concepts important from the language analysis point of view. Two of the important sections are those related to *kāraka* and *samāsa*. It is the *kāraka - vibhakti* mapping that provides a bridge between semantics and syntax. In this section, we show with examples, the importance Pāṇini has given to information coding in a language string, while developing the theory of language analysis.

We produce three evidences from Pāṇini's *Aṣṭādhyāyī* where Pāṇini makes subtle observations about the information coding in a sentence.

3.1 Anabhihite

Consider the following pair of Sanskrit sentences:

San: *rāmaḥ grāmam gacchati.*
gloss: Rama{nom} village{acc} go{active_voice,pr_tense,3_person,sg}

San: *rāmeṇa grāmaḥ gamyate.*
gloss: Rama{instr} village{nom} go{passive_voice,pr_tense,3_person,sg}

and
San: *gacchāmi.* / gacchasi.
gloss: go{active_voice,pr_tense,1/2_person,sg}

A typical computational linguist would say,

- In case of an active voice, a kartṛ gets a nominative case and a karma gets an accusative case.
- In case of a passive voice, kartṛ gets an instrument case and karma (in case of transitive case) gets a nominative case.
- kartṛ(karma) and the verbal suffix agree in number and person in active(passive) voice.
- Sanskrit also allows first person and second person pronoun drop.

This is fairly a good attempt to describe various phenomena observed in the above sentences. However, just as for a vaiyākaraṇa brevity is important[4], for a computational linguist not only the solution but also its optimality and generality matter the most. Optimality ensures that it consumes optimal time and space, whereas generality ensures that the same code will work for other languages as well. Further for a computer scientist, who is looking at the dynamics of information coding in a natural language, it becomes important to know 'where' exactly is the information about the kāraka roles coded? This helps him in deciding the parsing strategies.

Pāṇini handled the four cases described above in a very compact and elegant way. He gave the following sūtras:

1. *laḥ karmaṇi ca bhāve ca akarmakebhyāḥ (kartari) 3.4.69*
2. *anabhihite 3.1.1*
3. *kartṛkaraṇayoḥ tṛtīyā 2.3.18*
4. *karmaṇī dvitīyā 2.3.2*
5. *prātipadikārthaliṅgaparimāṇavacanamātre prathamā 2.3.46*

3.4.69 says that it is the *lakāra* (tense-aspect-modality marker) which expresses the *kartā*, *karma* or *bhāva* (action).

Having said this, now Pāṇini starts a section on mapping the *kāraka* relations into *vibhakti*s with the sūtra *anabhihite*[5]. In case the relation has not been expressed by any of other means, then the rules from 2.3.2 to 2.3.73 come into effect and the unexpressed *kāraka* relations are expressed through the *vibhakti*s. Then naturally, one would ask what does then the nominative case signify? According to Pāṇini (2.3.46) the nominative case just indicates the gender, number

[4] ardhamātrā lāghavena putrotsvamanyante vaiyākaraṇāḥ

[5] *Kātyāyana* in his *vārtika* (if not already expressed). on this sūtra states that there are 4 ways by which the *kāraka* relations can be expressed – by means of *tiṅ* suffix, *kṛt* suffix, *taddhita* suffix (derivational suffix deriving a noun) and *samāsa* (compound).

etc. and not any *kāraka* relation.

We see that Pāṇini deviates from the 'normal' thinking in two ways.

- He does not give two different rules for active and passive. But handles both by a single rule(Kiparsky, 1982).
- This he achieves, by his minute observation: which information is redundant and which is not. It is natural to think of vibhaktis associated with nouns as marking the relations. But with his 'lateral thinking', he categorically denies any 'information content related to the marking of relations' in the suffix denoting 'prathamā vibhkati', and claims the presence of relation marking information in the verbal suffixes.

What we learn from the way Pāṇini framed the rules is to look for **where** the information is coded. The very fact that language allows absence of pronoun triggers that it is the verbal suffix which codes the *kāraka* relation and not the nominative case.

Many a times a language has redundant information. It is necessary to identify which part of it is redundant and which part of the coding is genuine. The question "where does a language code information?" thus helps us in ruling out the redundant information which helps one to build a NLP system that is more reliable and robust.

3.2 How much information is coded

In the previous section we saw that the *vibhakti*s (case markers) are determined by the *kāraka* role a noun has with respect to the verb and the *prayoga*(voice)

$$vibhakti = \text{f}\ (kāraka, prayoga).$$

Vibhkati (case marker) and the *prayoga* (voice) are the surface level realities. *Kāraka*s are the basic syntactico-semantic categories. These categories, "serve as intermediaries between grammatical expressions and their semantics" (Cardona,1978) providing a bridge between the surface form and its meaning.

We argue below that Pāṇini, **by way of introducing an intermediary level of analysis draws a line between what is coded in a language string and what is extra-linguistic.**

Look at the sentences

1. *rāmaḥ kuñcikayā tālam udghāṭayati.*
2. *kuñcikā tālam udghāṭayati.*
3. *tālaḥ udghāṭyate.*

Semantically speaking, in the above sentences, *rāma* is an agent, *kuñcikā* is an instrument and *tālaḥ* is a goal. However, according to Pāṇinian analysis all of them are kartṛ. It is obvious that by calling all these three *kartā*s, the actual

semantic roles are not captured and one needs one more mapping from these *kāraka* roles to the thematic roles to arrive at the semantics. Natural question is then why Pāṇini did not go for the semantic analysis? And why did he chose the *kāraka* level analysis? Pāṇini observes

svatantraḥ kartā (1.4.54).

An activity involves more than one participants. The underlying verb expresses the complex activity which consists of subactivities of each of the participants involved. For example, in case of opening of a lock, three subactivities are very clearly involved (Bharati,1995), viz.

1. the insertion of a key by an agent,
2. pressing of the levers of the lock by an instrument (key), and
3. moving of the latch and opening of the lock.

Though in practice, to a large extent all three subactivities 1 through 3 together constitute the activity 'opening a lock', sometimes the subactivities 2 and 3 together are also referred to as 'opening a lock' and the activity 3 alone is also referred to as 'opening a lock'. Different languages may or may not have different lexical items expressing these subactivities. When we say *rāma*, *kuñcikā* and *tālaḥ* are the *kartā* of opening of a lock, *rāma* is the *kartā* of the complex activity 1 through 3, *kuñcikā* that of 2 through 3 and *tālaḥ* that of 3 alone.

Patañjali, in mahābhāṣya, interprets *svatantraḥ kartā* as: In a complex activity consisting of subactivities a_1 through a_m, if the speaker does not intend to mention participants capable of performing activities a_1 through a_j ($j < k$), the participant initiating the subactivity a_k will be the *kartā*.[6]

Thus in the absence of an agent (*rāma*), by promoting an instrument (*kuñcikā*) to *kartā*, Pāṇini draws our attention to the fact that language does not code information completely. Information related to the semantic encoding is not coded in a language string. To arrive at the conclusion that *kuñcikā* is an instrument and *tālaḥ* is a goal, one has to appeal to the world knowledge. The greatness of Pāṇini lies in **"identifying exactly how much information is coded and then giving it a semantic interpretation"** (*sūtras* 1.4.23 - 1.4.55). This level of semantics is the one which is achievable / reachable through the grammar rules and the language string alone. This puts an upper bound for the analysis, making it very clear what is guaranteed and what is not. We can extract only that which is available in a language string 'without any requirement of additional knowledge'. To give an analogy, one can not use low quality energy to do the high quality work.

[6] Patañjali on kārake 1.4.23: evaṃ tarhi pradhānena samavāye sthālī paratantrā, vyavāye svatantrā|tadyathā amātyādīnām rājñā saha samavāye pāratantrya.m vyvāye svātantryaṃ ||(in the absence of a king, the senior most minister will enjoy the powers of king.)

3.3 How (manner) is the information coded?

The *sup* and *tiṅ* suffixes assign *kāraka* roles to the nouns. The principles governing the relations between these suffixes with the *kāraka* roles are as under (Kiparsky, 2002).

1. Every *kāraka* must be expressed by a morphological element.
2. No *kāraka* can be expressed by more than one morphological element.
3. Every morphological element must express something.

Now consider a sentence

San: *rāmaḥ dugdham pītvā śālām gacchati.*
gloss: Rama{nom} milk{acc} after_drink{gerund} school{acc} go{pr,active,3p,sg}

In this sentence, there are two verbs viz. *gam* and *pā*. Both of them have a mandatory expectancy of two *kāraka*s viz. *kartṛ* and *karma*. Further the relation between the subordinate verb and the main verb should also be marked. Thus there are 5 relations which need to be marked. In the above sentence, there are 5 words and hence only 4 relations can be expressed through the suffixes. Relations that are expressed by the suffixes are shown in Figure 3.3.
The *kartā* of the verb *pā* is not marked explicitly. A native speaker, however,

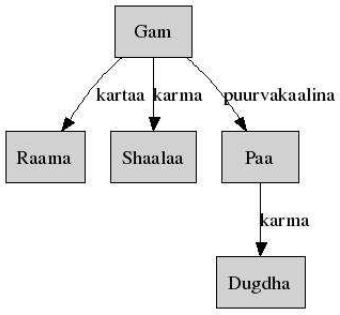

Fig. 1. modifier-modified relations

does not have any problem in answering the question 'who drank the milk?'. This indicates that it is the 'Language Convention' that tells: in case of *ktvā* suffix[7] the *kartā* of the subordinate verb is the same as that of the main verb. Pāṇini has postulated this in terms of a sūtra

[7] which indicates that the action corresponding to the verb with *ktvā* suffix takes place before the action indicated by the main verb.

samānakartṛkayoḥ pūrvakāle (3.4.21)

It is the **language convention which gives a license to not to code the information explicitly**. The implicit coding of the information may need extra processing for making such a knowledge explicit. It then becomes crucial for MT developers to know what is coded explicitly and what is coded implicitly. If the two languages have different language conventions, one needs to make implicit information explicit in other language. This may lead to unacceptable constructions, or even to a catastrophe, if not handled properly.

Consider
San:*vanāt grāmam adya upetya odanam āśvapatena apāci.* (Kiparsky, 2009)
gloss: forest{abl} village{acc} today after_reaching rice{nom} Asvapata{inst} cook{passive, past, 3pr,sg}

In Sanskrit, following the sūtra, *samānakartṛkayoḥ pūrvakāle*, it is clear that it is *āśvapata* who returned and it is he who cooked. But such constructions are not allowed in English. English needs passive absolutive, if the finite verb is in passive. Sanskrit uses same '*ktvā*' in both the active as well as passive form of the finite verb. Hence in MT they pose a problem, as they may lead to unacceptable / ungrammatical constructions.

4 Conclusion

With the emergence of Linguistics, linguists had started recognising the importance of Pāṇini's grammar. And now with the advent of computer technology, computational linguists have started recognising Pāṇini as an information scientist.

We conclude: The information coding and flow of information are at the center of the Pāṇinian analysis. The questions where does a language codes information, how much information does it code, and the manner in which it codes the information are the three dimensions or the parameters that are crucial in identifying the "true nature of the language". These three parameters may be used to determine the syntactic divergence between the languages. And hence we claim that any grammar which is developed with the three questions in mind: **where**, **how much** and **how** is the information coded, would be truly in Pāṇinian spirit.

Acknowledgement The author acknowledges the fruitful discussions she had with Vineet Chaitanya and K. V. Ramkrishnamacharyulu.

References

1. Abhyankar and Limaye, 1967. *Mahābhāṣya - Dīpikā of Bhartṛhari* Bhandarkar Oriental Research Institute, Pune, India
2. Bharati Akshar, Vineet Chaitanya, Rajeev Sangal, 1994. *NLP A Paninian Perspective,* Prentice Hall of India, Delhi, India
3. Cardona, George,1978. *Panini: A Survey of Research*, Mouton, Hague-Paris.
4. Cardona, George, 1988. *Panini: His Work and Its Tradition, (Vol. 1: Background and Introduction)*, Motilal Banarsidas, Delhi, India.
5. Jigyasu, Brahmadatt, 1979. *Ashtadhyayi (Bhashya) Prathamavrtti, three volumes*, Ramlal Kapoor Trust Bahalgadh, (Sonepat, Haryana, India)(In Hindi)
6. Joshi, S.D. (editor) 1968. *Patanjali's Vyakarana Mahabhashya*, (several volumes), Univ. of Poona, Pune, India.
7. Joshi, S.D. and Roodebergen J.A.F., 1998. *The Aṡṭādhyāyīof Pāṇini* (several volumes), Sahitya Akademi, Delhi, India.
8. Kiparsky, Paul, 2009. *On the Architecture of Panini's Grammar*, eds. Geŕard huet, Amba Kulkarni and Peter Scharf, Proceedings of the Sanskrit Computational Linguistics, Springer Verlag.
9. Kiparsky, Paul, 1982. *Some Theoretical Problems in Panini's Grammar*, Bhandarkar Oriental Research Institute, Poona, India.
10. Kiparsky, Paul, 1991. *Economy and the construction of the Śivasūtras*, in Pāṇinian Studies, ed. M. M. Deshpande and S. Bhate, Ann Arbor, Michigan.
11. Petersen, Wiebke, 2009. On the minimality of Pāṇini's Śivasūtras, eds. Amba Kulkarni and Geŕard Huet, In the proceedings of Third International Sanskrit Computational Linguistics Symposium, Springer Verlag.