Resolving Anaphors in Sanskrit

Madhav Gopal[1(✉)] and Girish Nath Jha[2]

[1] Centre for Linguistics, SLL & CS,
Jawaharlal Nehru University, New Delhi, India
mgopalt@gmail.com

[2] Special Centre for Sanskrit Studies, Jawaharlal Nehru University,
New Delhi, India
girishjha@gmail.com

Abstract. This paper is based on the M.Phil. dissertation of the first author. It talks about the automatic resolution of lexical anaphors in Sanskrit with the help of POS tagged data. The present approach exploits the grammatical features of Sanskrit language in determining the antecedents of anaphors and cataphors. A system called SARS has been developed and implemented. The input to the system is the POS tagged Sanskrit text in which various morphological features are attached to the words. The system is currently handling intra-sentential anaphors/cataphors and giving encouraging results.

Keywords: Anaphora resolution · Sanskrit · Pañcatantra · Reflexives · Reciprocals · Anaphors and cataphors · POS tagging

1 Introduction

Anaphora resolution has proven to be a formidable task for computers to perform. It plays a significant role in most of the NLP applications such as machine translation (MT), text summarization, question answering systems, information extractions etc. It is a very wide topic in NLP and generally, it includes lexical anaphors, third person pronouns, and other anaphoric items. In Sanskrit anaphor resolution is in initial state and has received very little computational research. Until now, only a few works (Jha et al. 2008, 2009, 2011) are available on this topic.

The present work is a small effort to help Sanskrit make translatable mechanically in other human languages so that it could be reachable to maximum number of people of the world sharing knowledge stored in it. Anaphora resolution systems are of crucial importance for correct translation. When translating into languages which mark the gender of pronoun, for example, it is essential to resolve the anaphoric relation in source language. In translation, the reference to a discourse entity encoded by a source language anaphor by the speaker (or writer) has not only to be identified by the hearer (translator or translation system) but also re-encoded in a co-referential/co-specificational expression of the target language. So, to do these things efficiently we need an anaphora resolution system for Sanskrit.

Z. Vetulani and J. Mariani (Eds.): LTC 2011, LNAI 8387, pp. 83–92, 2014.
DOI: 10.1007/978-3-319-08958-4_7

2 Early Works on Sanskrit Anaphora Resolution

Jha et al. (2008, 2009, 2011) have done a wider case study of anaphors in Sanskrit scanning the language from the epics to *śivarājavijayam*. They have discussed about the concept of anaphora and anaphora resolution techniques in Indian intellectual tradition including Vyākaraṇa, Nyāya, and Mīmāmsā. The papers have focused on pronominal anaphors and proposed a workable solution in terms of an algorithm. A Sanskrit Analysis System (SAS) containing several modules has also been developed that will supposedly assist the resolution. We have modified their generalisations and have presented the algorithm based on our study of the language in the text of Pañcatantra (PT). Thus, we have deviated from their proposal for reflexives and reciprocals.

Pralayankar and Devi (2010) also have presented an algorithm, which reportedly identifies different types of pronominals and their antecedents in Sanskrit. The computational grammar implemented there "uses very familiar concepts such as clause, subject, object etc., which are identified with the help of morphological information and concepts such as precede and follow". The method adopted for resolving the anaphors is by exploiting the morphological richness of the language. The system is reportedly giving encouraging results. This work is also not flawless. In their algorithms they have ignored the fact that Sanskrit reflexives are impersonal, i.e. they do not encode person feature and also that Sanskrit reflexives do not agree in gender with their antecedents. The algorithm does not consider reciprocals and cataphoric usage of reflexives and reciprocals. The paper does not define the term 'possessive' used in the algorithm.

3 Lexical Anaphors in Sanskrit

Sanskrit has a great number of lexical anaphors like other vocabulary items. It has only the nominal form of the anaphors: reflexives and reciprocals. The reflexive pronouns, like any nominal, inflect for case (barring *svayam*). The possessive reflexives, which could be more appropriately named as 'reflexive possessive adjectives', even inflect for number and gender also. The reciprocals are largely stagnant forms and do not inflect for any grammatical feature. However, very rarely casal inflections of some reciprocals are also available in Sanskrit texts. The reflexives and reciprocals are also used in compound constructions[1].

3.1 Reflexives

Sanskrit reflexives do not encode person feature; they are impersonal. The reflexive pronouns found in the PT could be classified in three categories:

[1] Our study of lexical anaphors described here is based on our study of the *Pañcatantra*. The examples are taken from different sections of this text and in transliteration we have followed IAST.

3.1.1 Proper Reflexives

In the text we have found *ātman* as the only proper reflexive pronoun. Like personal pronouns, it takes case suffixes, but it is invariant for gender and person features. In contrast to personal pronouns, referent *number* is not a coded feature of the reflexive *ātman*. However, its genitive forms behave like possessive reflexives and agree with the number of the antecedent, but unlike other possessive reflexives, they do not vary for gender agreement with the possessed item.

(1) *tataḥ* ***aham*** *pūrvam* *eva* ***ātmāna-m***
then 1sg.nom before emph self-acc
aratna-m *samarpya*
jewellery_less-acc surrender.grn
etān *muñcāmi.*
these.m.3pl.acc release.1sg.prs
'Then, I being without any jewellary, hereby, dedicate myself and get these people released.' <caurabrāhmṇa-kathā, mitrabheda>

3.1.2 Possessive Reflexives

sva, svīya, svaka, svakīya, ātmīya, and *nija* are possessive reflexives and are inflected for case, number and gender features. All of these possessive reflexives are semantically identical and mutually interchangeable. They are used more like an adjective and generally precede the modified element.

(2) *tvām* *dṛṣ-ṭvā* *dūrataḥ* *api*
2sg.acc see-grn distantly emph
caura-siṃha-ḥ *praviṣ-ṭaḥ* ***svam***
thief-lion-sg.nom enter-pspl self.m.sg.acc *durgam.*
fort.sg.acc
'Seeing you distantly, the thief lion entered in his fort.' <siṃha-śaśaka-kathā, mitrabheda>

3.1.3 Intensive Reflexives

The intensive reflexive pronouns *svayam* and *svayameva* are morphologically invariant and refer to the subject of the sentence. Intensive reflexives are like emphatic markers, and that is why they are called emphatic reflexives as well. The reflexive *svayam-eva* is semantically more emphatic; *eva* itself is an emphatic in the language. Otherwise *svayam* and *svayam-eva* are the same. Both of them do not occupy an argument position in a sentence. Referent number is not a coded feature of these. This type of reflexive too is used in compounds.

(3) *bhṛty-aiḥ* *vinā* ***svayam*** *rājā*
servant.pl.ins without self king
loka-anugraha-kāribhiḥ
people-welfare-doer.pl.ins
mayūkh-aiḥ *iva* *dīptāṃśuḥ* *tejasvī* *api*
ray-pl.ins like sun brilliant emph

na	*śobha-te*
not	suit-3sg.prs.pass

'As without rays the brilliant sun does not look beautiful, so is a king himself without servants who serve people.' <kīlotpāṭi-vānarakathā, mitrabheda >

3.2 Reciprocals

Reciprocal pronouns do not inflect for case, gender, number, or person; however, there is one instance of *parasparam* in the PT that has been used in genitive singular. All the reciprocals always need either a plural antecedent or a dual one for their interpretation. The reciprocals used in the PT include *anyōnyam*, *parasparam*, *itaretaram*, and *mit^{h}aḥ*.

(4)

tau	*ca*	***parasparam***
3du.m.nom	and	rec
mantra-yataḥ		
advise-3du.prs		

'And they (two) advise each other.' <prastāvanā-kathā, mitrabheda >

(5)

yathā	*chāyā-tap-au*	*nityam*		
as	shadow-sun-du	always		
su-sambaddh-au	***parasparam***			
well-affiliated-du.m	rec			
evam	*karma*	*ca*	*kartā*	*ca*
same_way	act.nom	and	doer.nom	and
saṃ-śliṣṭ-au	***itaretaram***			
well-stick.pspl-du	rec			

'As shadow and sun are always well connected to each other, so are the act and doer to each other.' <mandabhāgyasomi- laka-kathā, mitrasamprāptikam>

(6)

tataḥ	*ca*	*niśīth-e*	*yāvat*
afterwards	and	midnight.loc	rel
paśya-ti	*tāvat*	*tau*	*eva*
see-3sg.prs	then	3du.dst.m.nom	only
dvau	*puruṣ-au*	***mithaḥ***	*mantra-yataḥ.*
two	man-du.nom	rec	advise-3du.prs

'And afterwards, when in the midnight he sees, those two men advise each other.' <vṛṣabhānugaśṛgāla-kathā, mtsp>

3.3 Cataphoric Expressions

Expressions are cataphoric when they (pronoun etc.) precede the linguistic expression necessary for their interpretation in the given discourse. In PT this kind of expressions has been sparsely used. Some of such usages are illustrated through the following sentences:

(7) *tat śru-tva ga-ta-āyuṣa-m iva*
that listen-grn go-pspl-age-acc as
ātmān-am *manyamānaḥ* ***sanjīvaka-ḥ***
self-acc assuming Sanjeevaka-nom
param viṣāda-m agamat.
great sadness-acc go.3sg.pst
'After listening that, Sanjeevaka believing himself as dead, got to great sadness.' <śṛgāladundubhi-kathā, mitrabheda >

(8) *tataḥ svami-prasāda-rahit-au kṣut-*
then master-mercy-less-du.nom hunger-
kṣāma-kaṇṭh-au ***parasparam***
weak-throat-du.nom rec
karaṭaka-damanak-au *mantra-yete.*
Karaṭaka-Damanaka-du.nom advise-3du.prs
'Then, deprived of the mercy of their master, Karaṭaka and Damanaka advise each other.' <dantilagorambhayoḥ kathā, mitrabheda>

4 The Sanskrit Anaphora Resolution System (SARS)

The system developed is called Sanskrit Anaphora Resolution System (SARS). As input it takes POS tagged data of Sanskrit. It goes through five processes before it gets the final result. The input is pasted in the text area of the system which is there in the homepage of SARS, and then is sent for processing and resolution. The pre-processor of the system replaces the double *daṇḍas* (full stop marker in Sanskrit) with single *daṇḍas* (the double *daṇḍas* are used in the text at the end of a verse). Afterwards, the data is tokenized in single sentences delimited by a single *daṇḍa*. After sentence tokenization each word is tokenized. The system considers a single tokenised sentence at a time for resolving the anaphors. In each tokenized sentence the system first checks whether a reflexive or reciprocal is available in the sentence in question. If they are not there then it leaves that sentence and considers next sentence for the same. And if they are there, it searches for its antecedents as per our rules (see the algorithm) and resolves it. After resolving the anaphor in a sentence it moves on to the next sentence and operates likewise till the end of the input text. The candidates for being the antecedents of reflexives (PRF) and reciprocals (PRC) are common nouns (NC), proper nouns (NP), personal pronouns (PPR) and relative pronouns (PRL) as per our generalisations in the PT. All these categories are simply identified on the basis of their morphosyntactic tags. Thus POS tagging information is very crucial for anaphora resolution in this system.

4.1 Methods and Techniques Used

SARS is planned to develop in web architecture. The system tools are Java based. Front end is done in Java Server Pages (JSP) running on the web server called Apache Tomcat. The programming has been done in the Java environment. The system takes the input in text area, and then sends it for tokenisation and other steps described earlier.

4.2 The POS Tagging of the text

To conduct this research we have taken the text of PT which was POS tagged by the authors with Indic Language POS Tagset (IL-POST) developed by Microsoft Research India. The corpus was subsequently published by the LDC (http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2011T04). The IL-POST is a standard framework for tagging major Indian languages. This is a hierarchical tagset, based on guidelines similar to EAGLES. This tagset encodes information at three levels - Categories, Types, and Attributes. It is supposed to provide cross-linguistic compatibility, reusability, and interchange-ability. The tags used are extremely fine-grained, and incorporate a great deal of information about case, gender, number and so on. The IL-POST is able to accommodate all desired linguistic features of Sanskrit and the tagged corpus remains compatible with other languages tagged by a brethren tagset (Jha et al. 2009, 2011).

4.3 The SARS Algorithm

Currently the system considers only lexical anaphors (reflexives and reciprocals) and it identifies intrasentential antecedents of these anaphors in the input text. The algorithm is based on our study of lexical anaphors in the PT and it may work for the other similar texts as well.

4.3.1 Algorithm for Reflexive Anaphora

1. Tokenize each sentence (S) of the input text.
2. Pick up the S in which PRF tag occurs.
3. Check whether the S has NP, NC, PPR, or PRL.
4. Consider all the NP, NC, PPR, and PRL in the S that precede the PRF.
5. Check whether NP, NC, PPR, or PRL has/have .nom tag.
6. If one of the NP, NC, PPR, or PRL has .nom tag, then that word is identified as the antecedent of the PRF.
7. If more than one of these have .nom tag then the nearest to PRF would be its antecedent.
8. If the conditions 6 or 7 are not met then the NP, NC, PPR, or PRL having .ins tag would be considered the antecedent of the PRF.

4.3.2 Algorithm for Reflexive Cataphora

9. If the S does not have any preceding NP, NC, PPR, or PRL with.nom tag then consider the following NP, NC, PPR, or PRL containing.nom tag.
10. If one of the following NP, NC, PPR, or PRL has .nom tag, then that word is identified as the antecedent.

11. If more than one of these have.nom tag, then the nearest to the PRF would be its antecedent.
12. If the conditions 9 or 10 are not met then the NP, NC, PPR, or PRL having.ins tag would be considered the antecedent of the PRF.

4.3.3 Algorithm for Reciprocal Anaphora

1. Tokenize each sentence (S) of the input text.
2. Pick up the S in which the PRC tag occurs.
3. Check whether the S has NP, NC, PPR, or PRL.
4. Consider all the NP, NC, PPR, and PRL in the S that precede the PRC.
5. Check whether NP, NC, PPR, or PRL has/have .du.nom or .pl.nom tag.
6. If one of the NP, NC, PPR, or PRL has .du.nom or .pl.nom tag, then that word is identified as the antecedent of the PRC.
7. If more than one of these have .du.nom or .pl.nom tag then the nearest to the PRC would be its antecedent.
8. If all of them are containing .sg.nom tag then all of them would be antecedents of the PRC.
9. If the conditions 6, 7 or 8 are not met then the NP, NC, PPR, or PRL having .du.ins or. pl.ins tag would be considered the antecedent of the PRC.

4.3.4 Algorithm for Reciprocal Cataphora

10. If the S does not have any preceding NP, NC, PPR, or PRL then consider the following NP, NC, PPR, or PRL containing .du.nom or .pl.nom tag.
11. If one of the following NP, NC, PPR, or PRL has .du.nom or .pl.nom tag, then that word is identified as the antecedent of the PRC.
12. If more than one of the following NP, NC, PPR, or PRL have .du.nom or .pl.nom tag, then the nearest to the PRF would be its antecedent.
13. If all of the following NP, NC, PPR, or PRL have .sg.nom tag, then all of them would be collectively identified as the antecedent of the PRC.
14. If the conditions 11, 12 or 13 are not met then the NP, NC, PPR, or PRL having .du.ins or. pl.ins tag would be considered the antecedent of the PRC.

5 Testing of SARS

SARS has been tested on 3659 sentences from the POS tagged text of PT. These sentences are of various types that have been given to SARS for anaphora resolution. Before entering the data into the program for resolution we have noted the number of anaphor occurrences in the input text. In the input data 90 PRF (reflexives) and 25 PRC (reciprocals) were found (total 115 anaphor occurrences). Later on the number of anaphor antecedent pairs resolved by the system is noted. The intersentential and zero pronoun cases which remained unresolved have also been noted; such cases were 25 for reflexives and 1 for reciprocal has been found in such use. All these readings have been taken manually. The number of correctly and incorrectly found anaphor antecedent pairs is noted. Table 1 presents the summary of results we have obtained after testing SARS:

Table 1. Evaluation results of Sanskrit anaphors

Anaphors	Total	Correct	Wrong	Unresolved
Reflexives	90	50	15	25
Reciprocals	25	20	4	1
Total	115	70	19	26

5.1 Result Analysis and Limitation

Currently this system is giving results for intrasentential lexical anaphors. The results are encouraging. There were 89 intrasentential lexical anaphors and cataphors in the input text and out of which 70 were correctly resolved. However, an improvement is needed for better results. The system is not able to handle intersentential anaphors at present state. Also, if a sentence has both a reflexive and a reciprocal, it picks up one of them and gives the result. The cases of reflexive compounding and reciprocal compounding are not handled. We are reporting an ongoing work; in future, hopefully, we shall be reporting better results.

5.2 Limitations of the System

The system is able to identify only intrasentential lexical anaphors. There are a lot more anaphoric usages in the language that, yet, need to be handled automatically in order to facilitate the language processing tasks. The intersentential cases of reflexives and reciprocals are yet to be resolved. The system does not recognise clause boundaries and does not work for complex sentences validly. The compounding of all kinds of pronouns is a new type of problem in anaphora resolution research in Sanskrit. Such structures increase the load of the machine and reduce the efficiency of the system. The present system does not take care of compound reflexives and compound reciprocals. To resolve them it will require a separate module called compound processor. Also, the system does not consider pro drop, that is, zero pronoun cases.

6 Conclusion and Further Research

The applications of anaphora resolution are enormous. SARS program can prove to be a very useful tool for machine translation systems. If the anaphors and their antecedents are known to machine, the machine can take care of them when they are being translated in typologically different languages as there might be some issues of agreements. To fit to these agreements such resolution is unavoidable. As the system developed is highly scalable, it can be used for other Sanskrit texts with the same performance.

This is a small research on designing an anaphor resolution system for Sanskrit. The input to the system is POS tagged Sanskrit data. The linguistic analysis of the data for machine is based on the tagging information only. As the system is currently handling only intrasentential anaphors, it would have to be enhanced to cover wider areas of coreference resolution like intersentential anaphor handling and also to resolve third person pronouns which are often anaphoric. With the help of verbal inflections the cases of zero pronouns can also be handled, as the Sanskrit verb encodes information about its subject also. The use of anaphoric adjectives is a predominant feature of Sanskrit and their resolution is also warranted. To what extent morphological features will solve the problem and how the rest problems would be solved – are some of the relevant questions which require rigorous research in order to resolve satisfactorily.

To use this system professionally for other Sanskrit texts as well we would need the following additions:

- adding a sandhi processing module
- adding lexical resources
- adding compound processing module
- adding a comprehensive POS tagger

For the smooth running of the system the issues of sandhi splitting, compound processing, and punctuation marking need to be taken care of efficiently. In comparison to other Indian languages Sanskrit texts are a bit complex and therefore their pre-processing has to be done rigorously.

References

Gopal, M., Mishra, D., Singh, D.P.: Evaluating tagsets for Sanskrit. In: Jha, G.N. (ed.) Sanskrit Computational Linguistics. LNCS, vol. 6465, pp. 150–161. Springer, Heidelberg (2010)

Gopal, M.: Anaphor Resolution in the Sanskrit Text Panchatantra: A Rule Based Approach to Resolve Lexical Anaphors in Sanskrit. LAP LAMBERT Academic Publishing, Verlag, Paperback (2012)

Hobbs, J.: Resolving pronoun references. Lingua **44**, 311–338 (1978)

Jha, G.N., et al.: Anaphors in Sanskrit. In: Johansson, C. (ed.) Proceedings of the Second Workshop on Anaphora Resolution (WAR-II), NEALT Proceedings Series vol. 2. pp. 11–25. Tartu University Library, NEALT, Estonia (2008)

Jha, G.N., Sobha, L., Mishra, D.: Discourse anaphor and resolution techniques in Sanskrit. In: Sobha L., Branco, A., Mitkov, R. (eds.) Proceedings of the 7th Discourse Anaphora and Anaphora Resolution Colloquium (DAARC 2009), pp. 135–150. AU-KBC Research Centre, Chennai (2009)

Jha, G.N., Gopal, M., Mishra, D.: Annotating Sanskrit corpus: adapting IL-POSTS. In: Vetulani, Z. (ed.) LTC 2009. LNCS (LNAI), vol. 6562, pp. 371–379. Springer, Heidelberg (2011)

Lappin, S., Leass, H.J.: An algorithm for pronominal anaphora resolution. Comput. Linguist. **20** (4), 535–561 (1994)

Pralayankar, P., Devi, S.L.: Anaphora resolution algorithm for Sanskrit. In: Jha, G.N. (ed.) Sanskrit Computational Linguistics. LNCS, vol. 6465, pp. 209–217. Springer, Heidelberg (2010)