Extending the core functionalities of Aṣṭādhyāyī 2.0

Oliver Hellwig
University of Düsseldorf
hellwig7@gmx.de

Wiebke Petersen
University of Düsseldorf
petersen@phil.uni-duesseldorf.de

1 Introduction

The paper describes new layers of linguistic annotation and explorative tools that were added to the project 'Aṣṭādhyāyī 2.0'. These additions make it possible to execute complex research queries in the digital version of Pāṇini's grammar with minimal knowledge both of Sanskrit and database query languages. In the project 'Aṣṭādhyāyī 2.0', we have developed a digital edition of Pāṇini's grammar of Sanskrit. Pāṇini introduced linguistic concepts, such as thematic roles, abstract derivation levels, rewrite rules, and pre-concepts of phonemes and morphemes, all of which are used intensively in contemporary Linguistics. In addition, the Aṣṭādhyā-yī compresses the grammar rule system into text form by making use of inheritance structures, a sophisticated meta-language, and a marker system (Kiparsky 2009). Although these concepts and methods continue to be of highest scientific interest, the Aṣṭādhyāyī is rarely studied in modern formal Linguistics because readers who are not thoroughly acquainted with the Sanskrit grammatical tradition don't understand the terse Sanskrit text.

The research environment of 'Aṣṭādhyāyī 2.0' now opens up the content, the formal structure, and the encoding mechanisms of the Aṣṭādhyāyī to a wider scientific audience. We have built a web-based database edition with annotations on several linguistic levels.[1] While Petersen & Soubusta (2013) and Petersen & Hellwig (forthcoming) have dealt with the core database structure and the linguistic analysis, this paper describes the upper layers of annotation and introduces a customizable query mechanism for our database.

In Section 2, we will describe the annotation of *anuvṛtti* inheritance and of word-semantic concepts. Section 3 deals with the implementation of the search engine. Section 4 illustrates what kind of research problems can be tackled with the research environment Aṣṭādhyāyī 2.0.

[1] The web interface is accessible at http://panini.phil-fak.uni-duesseldorf.de/panini/.

2 Annotating semantic concepts and anuvṛttis

As described in Petersen & Soubusta (2013) and Petersen & Hellwig (forthcoming), we created the morphological and lexical annotation of the Aṣṭādhyāyī by closely following the analysis presented in Katre (1987). The same guidelines were adopted for the double checked annotation of the *anuvṛttis*. *Anuvṛtti* information is stored in a separate database table that records the number of a *sūtra* and the unique identifier of the component that is inherited in this *sūtra* according to Katre (1987). In several cases, components are not inherited over a continuous range of subsequent *sūtras*, but are missing in some of them. These rule blockings are discussed intensively in the Pāṇinian tradition and are sometimes controversial issues. So, adhering strictly to the *anuvṛttis* given in Katre (1987) was the most appropriate way of obtaining a homogeneous primary annotation. The *anuvṛtti* table contains a Boolean flag with which these blocked rules are marked in individual *sūtras*.

A modified version of the OpenCyc ontology[2] was used as the sense inventory for word semantic annotation. After having completed the annotation, we reduced the full ontology to an upper ontology by removing all branches that do not contain semantic concepts found in the Aṣṭādhyāyī. The remaining upper ontology has been reordered using Protégé[3], thereby performing simultaneously simplification of the concept hierarchy and domain adaptation. As could be expected, the structure of a modern Western ontology does not fit well the conceptual space of Indian texts. Animals, for instance, are ordered according to a Western scientific taxonomy in OpenCyc, while Indian texts frequently employ a conceptual subspace that resembles the structures found in the Amarakośa (Nair & Kulkarni 2010).[4] Another example for domain adaptation is the entry "writing" that is defined as "reading matter; anything expressed in letters of the alphabet" in the original ontology, and whose subclass "sacred text" contains the sibling classes "Veda" and "Mahā-bhārata". Obviously, Indian tradition would not insert the Vedas and the Mahābhārata under the same parent class (refer to the ordering described in Scharf (forthcoming)), and the overall parent class would not be considered to be a written, but rather an orally transmitted text. As the internal structure of the ontology influences the search results, this kind of domain adaptation is of highest importance for building a well usable query mechanism.

[2] http://www.cyc.com

[3] http://protege.stanford.edu

[4] Also refer to Hellwig (2014) for a comparative discussion of Western and Indian scientific taxonomies.

3 The search engine

Our data model is stored in a complex relational database that is tailored exactly to the research questions we are interested in (Petersen & Soubusta 2013). However, this database may be difficult to query for researchers without deeper knowledge of SQL. As systems are more likely to be used if their functionality is easy to understand (Davis, 1989), we decided to construct a simple query language that triggers the corresponding SQL-statements. This query language abstracts from the underlying relational database and reduces user queries to structured concatenations of keyword-argument pairs in the form of KEY(ARG). Arguments are either from fixed ranges of allowable values for a given keyword (e.g., gen(itive) or dat(ive) for the keyword "case"), or plain text, in which case the wildcards '*' (zero or any number of symbols) and '?' (exactly one symbol) are allowed. The logical operators AND, OR, and NOT are used to connect keyword-argument pairs. Users may employ brackets to assign priority to sub-clauses.

In order to allow the formulation of queries that are not supposed to be resolved on the level of a full *sūtra*, but on the level of a single component (which is usually a word), a special bracket type ('<','>') is reserved to bundle key-argument terms referring to one *sūtra*. For example, while 'case:dat AND num:pl' searches for all *sūtras* in which a dative component and a plural component occur, '<case:dat AND num:pl >' searches for all *sūtras* in which a component occurs that is a dative plural form.

Additionally, the search can be restricted to ranges of the Aṣṭādhyāyī by using the key term "snr" (*sūtra* number). 'snr(1-2)', for example, restricts the search to the first two books. More fine-grained selections are possible as well: 'snr(1.3—1.4)' searches in the third and fourth chapter of the first book, 'snr(1.2.4- 1.2.21)' restricts the search to the fourth to 21st *sutra* of the second chapter of the first book.

4 Research applications

Aṣṭādhyāyī 2.0 provides a customizable php web interface for viewing the linguistic annotation of Pāṇini's grammar. However, the higher level annotations and the query engine make it possible to handle complex research driven queries that go beyond the capacities of databases usually created in the Humanities:

- Combining query criteria: Our query language makes it easy to combine criteria from different annotation levels into one query statement. One example is Pāṇini's use of noun cases as metalinguistic markers. Using our query engine, one can search for lexemes in certain nominal cases ("all instances of the word '*cu*' ('palatal') in the genitive case"), which are frequently constituents of rewrite rules.
- Querying semantic concepts: Aṣṭādhyāyī 2.0 started from the idea of making Pāṇini's grammar accessible to researchers with a limited knowledge of Sanskrit. Searching for semantic concepts is, therefore, an important step towards achieving this aim. Researchers can search for passages that contain particular semantic concepts or combinations of them ("all *sūtras* that contain a word meaning 'horse' or 'cow'") or even classes of semantic concepts ("all *sūtras* that contain one of the *pratyāhāras*"). Hellwig & Petersen (forthcoming) present a first case study for such queries.
- Making *anuvṛttis* visible: If desired, queries include the complete linguistic annotation of inherited elements for each *sūtra*. When searching for the concept *guṇa*, for example, the engine will retrieve, among others, *sūtra* 1.1.4 (*na dhātulope ārdhadhātuke*) that inherits the term *guṇa* from 1.1.3.

References

Davis, F. (1989), 'Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology', MIS Quarterly 13(3), 319-340.

Hellwig, O. (2014), 'Materialgruppen in der indischen Alchemie', Zeitschrift der Deutschen Morgenländischen Gesellschaft 164(2), 451-468.

Hellwig, O. & Petersen, W. (forthcoming), What's Pāṇini got to do with it? The use of *gaṇa*-headers from the Aṣṭādhyāyī in Sanskrit literature from the perspective of Corpus Linguistics, in 'Proceedings of the WCS 2015'.

Katre, S. M. (1987), Aṣṭādhyāyī of Pāṇini, University of Texas Press, Austin TX.

Kiparsky, P. (2009), On the architecture of Pāṇini's grammar, in G. Huet, A. Kulkarni & P. Scharf, eds, 'Sanskrit Computational Linguistics', Vol. 5402 of Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, pp. 33-94.

Nair, S. & Kulkarni, A. (2010), The Knowledge Structure in Amarakośa, in G. Jha, ed., 'Sanskrit Computational Linguistics', Springer Berlin / Heidelberg, pp. 173-189.

Petersen, W. & Hellwig, O. (forthcoming), Annotating and analyzing the Aṣṭādhyāyī, in 'Proceedings of CILC 2014'.

Petersen, W. & Soubusta, S. (2013), Structure and implementation of a digital edition of the Aṣṭādhyāyī, in M. Kulkarni, ed., 'Recent Researches in Sanskrit Computational Linguistics', D.K. Printworld, pp. 84-

103.

Scharf, P. (forthcoming), Providing access to manuscripts in the digital age, in ‘Writing the East: History and New Technologies in the Study of Asian Manuscript Traditions’, Schoenberg Center for Electronic Text and Imaging, Philadelphia.