# Multi-layer Annotation of the Ṛgveda

**Oliver Hellwig**[1] **, Heinrich Hettrich**[2] **, Ashutosh Modi**[3] **, Manfred Pinkal**[3]

[1]SFB 991, Düsseldorf University, ohellwig@phil-fak.uni-duesseldorf.de
[2]Würzburg University, heinrich.hettrich@uni-wuerzburg.de
[3]Department of Language Science and Technology, Saarland University, {ashutosh,pinkal}@coli.uni-saarland.de

**Abstract**

The paper introduces a multi-level annotation of the ṚGVEDA, a fundamental Sanskrit text composed in the 2. millenium BCE that is important for South-Asian and Indo-European linguistics, as well as Cultural Studies. We describe the individual annotation levels, including phonetics, morphology, lexicon, and syntax, and show how these different levels of annotation are merged to create a novel annotated corpus of Vedic Sanskrit. Vedic Sanskrit is a complex, but computationally under-resourced language. Therefore, creating this resource required considerable domain adaptation of existing computational tools, which is discussed in this paper. Because parts of the annotations are selective, we propose a bi-directional LSTM based sequential model to supplement missing verb-argument links.
**Keywords:** Sanskrit, multi-layer corpus, verb-argument structures, Ṛigveda

## 1. Introduction

In this paper, we introduce a multi-layer annotation of the complete ṚGVEDA (ṚV). The ṚV is central for research in Indo-European linguistics, because it is the oldest sample of this language family for which a sizeable text corpus has been transmitted (Witzel, 1995). The Vedic core corpus consists of four collections of hymns called Vedas ("knowledge"), which deal mainly with the worship of the Vedic pantheon and details of the ritual. The ṚV is the oldest among these four collections. It may have been composed around 1,500 BCE, has been transmitted orally for at least two millenia, and has, in spite of its age, remained a foundational text for the religious, cultural, and linguistic history of South Asia. Ideas and actors mentioned in the ṚV are constantly referred to in later texts produced on the Indian subcontinent (Gonda, 1975).

The corpus presented in this paper merges the phonetic, morphological, and lexical analyses for each word of the ṚV with a verb-argument (VA) annotation that links each verbal form to its main syntactic arguments. The first part of the paper describes how the annotation was performed, and provides a quantitative overview of size and structure of the resulting corpus. The second part of the paper presents a basic argument identification algorithm for Vedic Sanskrit. The VA annotation was created in a linguistic research context, but not as a building block of an NLP pipeline. As a consequence, parts of the case semantic information are not encoded explicitely, but need to be supplemented by a human reader. We will use the presented argument identification algorithm for complementing these missing parts of the VA annotation, and, later on, for annotating other central texts of the Vedic corpus.

This paper makes two important contributions. First, it introduces a novel resource for Sanskrit with deep linguistic annotations, ranging from the phonetic up to the syntactic level. The full annotated corpus is available via `https://git.adwmainz.net/open/rigveda` under the Creative Commons Attribution 4.0 International Public License. We expect that our ṚV annotation will become a standard reference resource for (diachronic) Indo-European linguistics, and for religious and cultural studies. Second, we describe how we merged two independent linguistic annotations to build a large digital corpus for a challenging South-Asian language, even though Vedic Sanskrit is strongly under-resourced from the viewpoint of NLP. Moreover, initial experiments with an argument identification algorithm that are reported in this paper, open up interesting perspectives for future research in automatic verb-argument detection.

The rest of the paper is organized as follows. Section 2. gives an overview of related research in Sanskrit CL and Vedistic studies. Sections 3. and 4. describe the morpho-lexical and verb-argument annotation of the data, and the necessary domain adaptation. Section 5. describes how these two annotation levels were merged into a single consistent format. Section 6. describes the algorithm developed for argument identification, and Sec. 7. summarizes the paper.

## 2. Related Research

Several authors studied the Vedic case system in general as well as the semantic functions of individual cases (Haudry, 1977; Hettrich, 2007; Kulikov, 2009). Detail studies such as Dahl (2014) also assessed how certain semantic roles are realized at the morpho-syntactic level in early Vedic. While these contributions focus strongly on language use in the ṚV, other important aspects of Vedic syntax such as word order or morpho-syntactic alignment were only studied for later Vedic prose (Delbrück, 1888) or with limited material from the ṚV. We are confident that our multi-layer annotation of the ṚV will provide the basis for large-scale studies of such phenomena in the oldest layer of Vedic.

To our knowledge, there exist no computational processing tools nor publicly available annotated corpora for the Vedic language. Hellwig (2009) introduced a stochastic morpho-lexical tagger for classical Sanskrit, which was extended to early Vedic for this paper. The systems described by Huet (2006), Kulkarni and Shukla (2009), and Jha (2009) aim at classical Sanskrit and are strongly influenced by the Pāṇinian framework of grammar, so that an extension to Vedic is not easily feasible.

| | | | |
|---|---|---|---|
| Input text in Devanagari | वासयोषसः | | श्रवसे |
| Input text in Latin | vāsayoṣasaḥ | | śravase |
| Sandhi split | vāsaya $_{(a+u=o)}$ uṣasaḥ | | śravase |
| Lexemes | vāsay | uṣas | śravas |
| Morphology | 2. sg., imp. | acc. pl. | dat. sg. |
| Word meanings | to make shine | dawn | fame |
| Translation | **"For fame make the dawns shine."** | | |

Figure 1: Levels of linguistic analysis for ṚV, 1.134.3. Abbreviations: sg.: singular, imp.: imperative; pl.: plural, acc.: accusative, dat.: dative. Translation taken from Jamison and Brereton (2014, 304)

## 3. Morpho-lexical Annotation

We used a tagger that was originally developed for Classical Sanskrit (Hellwig, 2015a) for creating a morpho-lexical analysis of the ṚV in the edition of van Nooten and Holland (1994). This tagger produces all possible tokenizations of the input text that consist of morphologically and lexically valid word forms. Tokenization of Sanskrit is a challenging task, because individual words are merged by a set of phonetic rules called Sandhi ("connection"), whose resolution is non-deterministic and, therefore, guided by the morphological, lexical, and semantic composition of a sentence.[1] This tokenization step results in a trellis of possible readings for each line of text. A dynamic programming approach that operates with a trigram language model (Brants, 2000) is used to find the most probable lexical path through this trellis. Final fine-grained morphological decisions are made by applying a Conditional Random Field (Lafferty et al., 2001) model to the most probable lexical path. The solutions are ordered by decreasing linguistic probability, given the data from the language model. The first author of this paper finally validated all proposed system analyses in a manual correction step, resulting in a morphological and lexical gold annotation of the complete ṚV. Figure 1 shows a schematic overview of annotation levels for a part of hymn ṚV, 1.134.

### 3.1. System Adaptation

Although Classical Sanskrit developed out of a late form of Vedic Sanskrit, which was described by the grammarian Pāṇini, they represent two separate layers of Old Indo-Aryan. Therefore, we needed to perform domain adaptation of the tagger in three linguistic areas:

1. The inflectional system of Vedic Sanskrit has preserved Indo-European traits that are extinct in Classical Sanskrit, and formation of verbal forms is partly opaque in older Vedic.[2] We extended the morphological rule base and the full form dictionary of the tagger on per case basis, using Macdonell (1916). Figure 2 sets the number of newly added verbal forms (y-axis) in relation to the progress of annotation (x-axis). The plot shows that the number of cases in which we had to extend the full form database manually decreases over time, indicating improving adaptation to the new linguistic domain.

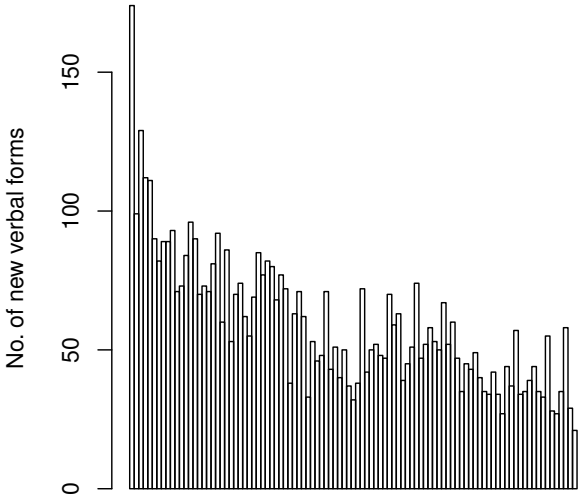


Figure 2: Distribution of the 5,908 newly added verbal forms over the annotation process (x-axis), illustrating the increasing domain adaptation of the tagger.

2. Due to the chronological distance of approximately 1,000 years and fundamental differences in genres and topics, Vedic and Classical Sanskrit use rather different vocabularies (Hellwig, 2017). Vedic texts in general and especially the ṚV contain many words that have disappeared in Classical Sanskrit. In addition, lexical semantics differ strongly between Vedic and Classical Sanskrit. The noun *vadha*, for example, can denote a tool for killing in the ṚV (e.g., ṚV 10.102.3), while it only denotes the act of killing in Classical Sanskrit. Bayesian models of semantic change (Frermann and Lapata, 2016) or diachronically motivated word embeddings (Hamilton et al., 2016) are not easily applicable, because the Vedic subcorpus is small,[3] and the text historical research in older Sanskrit literature is full of uncertainties (Fosse, 1997). The lexical database of the tagger was adapted to the Vedic vocabulary using the specialized dictionary of Grassmann (1873), and Geldner's German translation of the text (Geld-

[1] The words *rājā* 'the king' and *uvāca* 'he said', for example, are merged into the string *rājovāca* by the Sandhi rule *ā+u=o*. Sandhi complicates the linguistic processing of Sanskrit, because different Sandhi rules can result in the same merged phoneme. In the given example, the merged phoneme *o* could also have been produced by the rules *a+u*, *a+ū*, or *ā+ū*. Refer to Kielhorn (1888, 6ff.) and Hellwig (2015b) for further details.

[2] Consider the form *akrān* "he shouted", which is derived from the root *krand* by the following sequence of phonetic operations: lengthening of the root vowel (*krānd*), hardening of the final consonant (*krānt*), adding inflectional suffixes expressing number and person to the root (*krāntt*), dropping multi-consonant clusters at the end of the root (*krān*), and prefixing an augment (*akrān*).

[3] It may contain less than 5 million tokens, 5-10% of which have been analyzed so far.

ner, 1951 1957), from which we have integrated 923 and 1,599 new lexical meanings, respectively.

3. Basic syntactic structures have changed considerably from Vedic to Classical Sanskrit. While Vedic Sanskrit uses syntactic constructions such as relative and subordinate adverbial clauses (Hettrich, 1988; Hock, 2013), Classical Sanskrit tends to express subordination through compounding (Lowe, 2015) and absolutives (Tikkanen, 1991). In order to integrate syntactic changes in the tagging process, its trigram language model was split into two submodels, one trained on Classical Sanskrit, and one on the combination of the Classical and Vedic subcorpora.

### 3.2. Evaluation

This paper focuses on resource building, and the corpus of old Vedic is, more or less, restricted to the ṚV. Therefore, we did not evaluate the performance of the adapted tagger systematically. In order to get an idea of its performance, the main author of this paper recorded errors made by the system when analyzing the two hymns 8.102 and 8.103, both of which are dedicated to Agni, the god of fire and sacrifice. The evaluation distinguishes between three types of errors (Hellwig, 2015a):

1. Tokenization error: The system fails to propose the correct tokenizing split of a string. Example: *upastutāsaḥ*; correct: no split ('those [nom. pl. m.] who are praised'); system proposal: *upastutā-asaḥ* ('you [nom. sg. f.] will be the praised one'). Tokenization errors invalidate the analysis of a whole string.

2. Lemmatization error: The system fails to choose the right lexeme in a correctly tokenized string. Example: *rātahavyaḥ*: correct 'who has bestowed the oblation' (*rāta* = past participle of the verb *rā* 'to give'); system proposal: 'the oblation of Rāta' (*rāta* = name of a man).

3. Morphological error: Both preceding steps are solved correctly, but the system proposes a wrong morphological analysis of a token. Example: *haviṣkṛtaḥ*: correct: 'of him who prepares the oblation' (gen. sg.); system proposal: nom. pl. ('those who prepare the oblation'). Note that this form has 13 valid morphological readings.

Table 1 reports the error levels for the two Agni hymns. The highest error rates are observed on the morphological level. Given the morphological complexity of Vedic Sanskrit, this outcome is not really surprising. On the other hand, the system has a remarkably high tokenization accuracy. This somehow unexpected result is to a large degree due to the scholarly preprocessing of the ṚV, whereby Sandhis are resolved as far as possible in order to facilitate word search. When run on unpreprocessed (*saṃhitā*) texts, the system will certainly make a higher number of tokenization errors.

| Type | Number | Proportion |
|---|---|---|
| 8.102 (208 strings, 228 tokens) | | |
| Tokenization | 7 | 3.4% |
| Lemmatization | 5 | 2.1% |
| Morphology | 15 | 6.6% |
| 8.103 (189 strings, 209 tokens) | | |
| Tokenization | 4 | 2.1% |
| Lemmatization | 10 | 4.8% |
| Morphology | 22 | 10.5% |

Table 1: Error evaluation for the two hymns 8.102 and 8.103. Proportions are calculated w.r.t. the number of strings for tokenization, and the number of tokens for lemmatization and morphology.

## 4. Verb-Argument Annotation

The second component of the corpus provides verb-argument structure. It is based on the manual annotation of 27,104 verbal heads in the complete ṚV by one of the authors of this paper (Hettrich, 2001; Hettrich, 2007), an expert on Indo-European languages, and Vedic Sanskrit in particular. Verbs and dependents are connected with labelled edges. In many cases, part of speech and semantic type of the head noun are annotated in addition.

The original motivation for the annotation was its use in personal linguistic research. As a consequence, (1) the used inventory of relation types was independently developed and does not follow a standard dependency-grammar or role-semantic annotation schema, and (2) the labelling is selective, i.e., relations that are evident and can be easily supplemented by the reader are not labelled.

Concerning problem (1), the verb-argument annotations basically refer to the level of grammatical relations, with a number of semantically motivated refinements (for example, *comitative*, *separative*, *partitive*, *oblique agent*). The statistics across all 54,038 manually labelled edges shows a distribution with about 20 frequently occurring labels, from instrumental (2286) and adverbial (2153) to comparison (71) and predicative adverbial (20). The long tail of infrequent tags comprises standard tags labeled as uncertain or unusual, or combinations of them. We left the label inventory unchanged, leaving the mapping to coarser or more standard relation inventories (e.g., Universal Dependency Grammar) to the user.

Problem (2), the incompleteness of annotation, is a greater challenge. In particular, edges are missing for 9,585 occurrences of subjects and 8,573 occurrences of direct objects (non-oblique cases). In these cases, the whole construction is manually tagged to express the information that it comes with a subject and/ or object, but the location of the dependent is not disambiguated in the Sanskrit text. Section 5. describes how we deal with these cases.

## 5. Merging the Annotation Layers

Because morpho-lexical and verb argument annotations were performed independently of each other, and their results were stored in different data formats, we needed to merge them into a uniform multi-layer format. Merging was performed in two steps. In the first step, we established a mapping between the individual verbal roots that serve as heads of the verbal constructions (strings *vāsaya* in Fig. 3). As Indological researchers use different systems for en-

coding the lemmata of verbal roots, and for disambiguating homonymous verbal roots, this step involved a large amount of manual intervention, including the definition of 67 mapping rules between verbal roots.[4] 867 verbal heads can neither be mapped directly nor by applying the 67 special mapping rules. Moreover, we encountered 927 cases of copula omission, in which the verb-argument annotation does not disambiguate the arguments of the unexpressed copula.[5] These cases require a manual annotation step.
In the second step, we connect the arguments with their verbal heads. Apart from the problems with lemma encoding (see above), most non-oblique arguments are not disambiguated (see Sec. 4.). We deal with undisambiguated arguments in two ways:

1. 96.4% of all 24,861 disambiguated arguments occur in the same line of text as their verbal heads. Therefore, if verb $v$ has an undisambiguated argument in case $c$, and the morpho-lexical annotation records exactly one word $w$ with case $c$ in the same text line as $v$, $w$ is automatically disambiguated as the argument of $v$. This step produces 2,329 heuristic argument annotations.

2. If the preconditions for applying this heuristic are not met, because one line of text contains more than one word with case $c$, we use the labeling algorithm described in Section 6. for pre-annotating the arguments, and ask a human annotator to correct the output of the labeler.

The merged annotation contains 21,218 verbal heads, 20,438 of which are linked with a verbal form, while the remaining ones constitute sentences with missing copulae. Each verb has an average of 2.2 arguments (verbs without arguments: 6,399; with one arg.: 7,350; with 2-4 args.: 7,222; with more than 4 args.: 247).[6]

## 6. An Algorithm for Argument Identification

As mentioned in Sec. 4. and 5., the verb-argument annotation is selective. Therefore, we designed a basic argument identification algorithm that supports the re-annotation of non-oblique cases. Semantic role labeling is an active field of research in CL, and distinguishes between argument identification and argument classification (Gildea and Jurafsky, 2002). A wide range of learning algorithms such as probabilistic frameworks (Gildea and Jurafsky, 2002),

[4]The verbal roots are referenced by strings in the VA annotation and by unique numeric IDs on the morpho-lexical level. The 67 mapping rules need to disambiguate homonymous verbal roots such as *vas*, which can mean "to dwell" (*vasati*), "to wear" (*vaste*), or "to shine" (*ucchati*).

[5]The VA annotation indicates that a line of text contains a copula construction, but does not disambiguate the involved nominatives. – Use of copulae is optional in Sanskrit, with a strong tendency of not using it. So, the statement "Rāma is rich" can be expressed as *rāmo dhanyo 'sti* (*[a]sti* is the copula), or, more frequently as *rāmo dhanyaḥ*. Combined with the lack of punctuation, copula omission makes many Sanskrit texts highly ambiguous.

[6]Differences to numbers given for the VA annotation in Sec. 4. are due to mapping problems during the merging process.

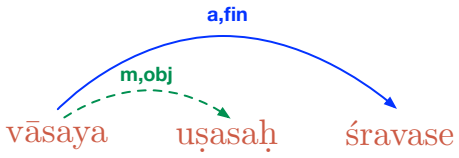


Figure 3: Full annotation of ṚV, 1.134.3 (refer to Fig. 1 for the morpho-lexical annotation). Labels on the arcs indicate the syntactic functions (obj[ect], fin[al]) and coarse word semantic classes (m = human, a = generic expression) of the arguments. Dashed arcs indicate arguments that are not disambiguated in the VA annotation.

Large Margin classifiers (Pradhan et al., 2008), and (recurrent/recursive) neural networks (Collobert et al., 2011; Roth and Lapata, 2016) was applied for semantic role labeling, and most authors emphasize the importance of high-quality parse trees as input features. While parse tree are not available for Vedic nor for Classical Sanskrit, the morpho-lexical annotations (Sec. 3.) provide a rich set of linguistic features that can be assumed to serve as proxies for syntactic relations in a weakly configurational language such as Sanskrit. Consequently, role identification is more challenging than role classification in Vedic Sanskrit, because role classification can make use of morphological features, as soon as a relation between a verb and an argument has been established. Under these circumstances, semantic role labeling can be viewed as a binary sequence annotation task: Given a sequence of noun forms and a verb, the classifier should select those noun forms that are arguments of the verb. Following recent research in CL and ML, we choose a recurrent neural network for this task.

Argument identification is performed in two steps. In the first step, an input line is processed with the morpho-syntactic tagger (Sec 3.), to get verbal forms $V$ and their possible arguments $A$ (all nouns and adjectives). In the second step, a neural network based sequential model is used to predict, if an argument at a given position $t$ is related to a verb $v$ or not. Note that the model works with text lines instead of sentences, because Sanskrit has not punctuation that indicates sentence ends. A single line of text can therefore contain more than one main verb. The neural network model has two bi-directional recurrent layers (Schuster and Paliwal, 1997). Each recurrent layer consists of 100 LSTM cells (Hochreiter and Schmidhuber, 1997). Prediction is done with a binary softmax activation function at the output. The model is trained by optimizing cross-entropy cost function and RMSProp schedule (Tieleman and Hinton, 2012), using an initial learning rate of 0.001, and a batch size of 8. Figure 4 shows the architecture for the model.

Each word of a text line is fed sequentially to the model. For each input word, lookup matrices are used to obtain the embeddings for the corresponding lemma and morphological information (e.g. case, gender, etc.). The embeddings for the lemma and morphological information are concatenated, and go as input to the first LSTM layer. The lemma embeddings are obtained by pre-training a word2vec skip-gram model (Mikolov et al., 2013) on the full corpus of

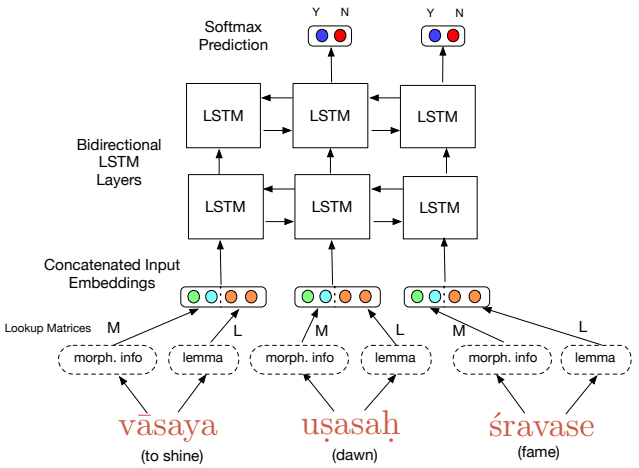


Figure 4: Bidirectional LSTM based neural network model for argument identification, unfolded for the task of labeling the sentence *vāsaya uṣasaḥ śravase* ("For fame make the dawns shine"; see Fig. 1). The model is trained to predict the words *uṣas* and *śravas* as argument of the verb *vāsay*.

| Config. | P | R | F | N |
|---|---|---|---|---|
| all cases | 72.74 | 69.35 | 71.00 | 20743 |
| oblique cases | 76.55 | 79.11 | 77.81 | 14318 |
| non-obl. cases | 60.74 | 46.55 | 52.71 | 6425 |
| nom. | 56.36 | 40.67 | 47.25 | 2606 |
| acc. | 61.84 | 47.69 | 53.85 | 2714 |
| ins. | 78.12 | 82.08 | 80.05 | 5078 |
| dat. | 75.51 | 81.66 | 78.46 | 3744 |
| abl. | 72.46 | 78.76 | 75.48 | 881 |
| gen. | 58.21 | 38.41 | 46.28 | 930 |
| loc. | 79.20 | 82.47 | 80.80 | 3685 |

Table 2: P(recision), R(ecall), and F score of the recurrent argument identifier. Column N gives the total number of training samples available for each configuration.

Classical and Vedic Sanskrit. The embeddings corresponding to morphological information are randomly initialized and learnt during training.

Since the dataset is small, the model is evaluated using 10-fold cross-validation with disambiguated and heuristically annotated arguments (Sec. 1). The model obtains an overall F score of 71.00 for roles in all cases, and of 77.81 when only oblique cases are considered (see Tab. 2). While this result compares favorably with results reported for verb argument detection (identification) tasks in English (Carreras et al., 2008; Das et al., 2013), one should keep in mind that the use of morpho-lexical gold information for Sanskrit, missing punctuation, and the small size of the Sanskrit corpus, when compared with corpora of modern languages, make a direct comparison impossible.

## 7. Summary

We have described the construction of a large-scale, multi-level annotation of the ṚGVEDA. In spite of the linguistic challenges raised by this complex Sanskrit text, we managed to merge two independent data sources into a consistent corpus, which we expect to become a standard reference tool in linguistic and cultural research. Post-processing and disambiguation of non-oblique arguments is work in progress. We designed an algorithm for argument identification that supports us in this ongoing task. In the future, we plan to extend this algorithm into a full-fledged verb-argument labeler for Sanskrit. Such a labeler will first be applied to Vedic texts that resemble the ṚV on the linguistic level (e.g., the metrical Atharvaveda), and later to the large corpus of Vedic prose texts.

## Acknowledgements

Research for this project was partially funded by the Cluster of Excellence "Multimodal Computing and Interaction" of German Science Foundation (DFG). We thank the Akademie der Wissenschaften und der Literatur Mainz for hosting the annotated corpus.

## References

Brants, T. (2000). TnT - a statistical part-of-speech tagger. In *Proceedings of the 6th Applied NLP Conference*, Seattle.

Carreras, X., Litkowski, K. C., and Stevenson, S. (2008). Semantic Role Labeling: An introduction to the special issue. *Computational Linguistics*, 34(2).

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Dahl, E. (2014). The morphosyntax of the experiencer in early Vedic. In Silvia Luraghi et al., editors, *Perspectives on Semantic Roles*, pages 181–204. John Benjamins Publishing Company.

Das, D., Chen, D., Martins, A. F. T., Schneider, N., and Smith, N. A. (2013). Frame-semantic parsing. *Computational Linguistics*, 40(1):9–56.

Delbrück, B. (1888). *Altindische Syntax*. Verlag der Buchhandlung des Waisenhauses, Halle.

Fosse, L. M. (1997). *The Crux of Chronology in Sanskrit Literature*. Scandinavian University Press, Oslo.

Frermann, L. and Lapata, M. (2016). A Bayesian model of diachronic meaning change. *Transactions of the Association for Computational Linguistics*, 4:31–45.

Geldner, K. F. (1951–1957). *Der Rig-Veda. Aus dem Sanskrit ins Dt. übersetzt und mit einem laufenden Kommentar versehen von Karl Friedrich Geldner*. Harvard University Press, Cambridge, Mass.

Gildea, D. and Jurafsky, D. (2002). Automatic labeling of semantic roles. *Computational linguistics*, 28(3):245–288.

Gonda, J. (1975). *Vedic literature (Saṃhitās and Brāhmaṇas)*. A History of Indian Literature, Vol. I, Fasc. 1. Otto Harrassowitz, Wiesbaden.

Grassmann, H. (1873). *Wörterbuch zum Rig-veda*. Otto Harrassowitz, Wiesbaden.

Hamilton, W. L., Leskovec, J., and Jurafsky, D. (2016). Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the 54th Annual Meeting of the ACL*, pages 1489–1501.

Haudry, J. (1977). *L'emploi des cas en Vedique*. Les Hommes et les Lettres. Service de reproduction des thèses, Université de Lille III.

Hellwig, O. (2009). SanskritTagger, a stochastic lexical and POS tagger for Sanskrit. In Gérard Huet, et al., editors, *Sanskrit Computational Linguistics. First and Second International Symposia*, Lecture Notes in Artificial Intelligence, 5402, pages 266–277, Berlin. Springer Verlag.

Hellwig, O. (2015a). Morphological disambiguation of Classical Sanskrit. In Cerstin Mahlow et al., editors, *Systems and Frameworks for Computational Morphology*, pages 41–59, Cham. Springer.

Hellwig, O. (2015b). Using Recurrent Neural Networks for joint compound splitting and Sandhi resolution in Sanskrit. In Zygmunt Vetulani et al., editors, *Proceedings of the 7th LTC*, pages 289–293.

Hellwig, O. (2017). Coarse semantic classification of rare nouns using cross-lingual data and recurrent neural networks. In *Proceedings of the IWCS*.

Hettrich, H. (1988). *Untersuchungen zur Hypotaxe im Vedischen*. de Gruyter, Berlin, New York.

Hettrich, H. (2001). Die Erarbeitung einer Kasussyntax des Ṛgveda auf der Grundlage elektronisch gespeicherter Daten. In *Maschinelle Verarbeitung altdeutscher Texte*, pages 73–81. Max Niemeyer Verlag, Tübingen.

Hettrich, H. (2007). *Materialien zu einer Kasussyntax des Ṛgveda*. Universität Würzburg, Würzburg.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Hock, H. H. (2013). Some issues in Sanskrit syntax. In Peter M. Scharf et al., editors, *Proceedings of the Seminar on Sanskrit syntax and discourse structures*, Paris.

Huet, G. (2006). Lexicon-directed segmentation and tagging of Sanskrit. In B. Tikkanen et al., editors, *Themes and Tasks in Old and Middle Indo-Aryan Linguistics*. Motilal Banarsidass, Delhi.

Jamison, S. W. and Brereton, J. P. t. (2014). *The Rigveda: the Earliest Religious Poetry of India*. Oxford University Press, New York.

Jha, G. N. (2009). Inflectional morphology analyzer for Sanskrit. In Gérard P. Huet, et al., editors, *Sanskrit Computational Linguistics*, pages 219–238. Springer, Berlin, Heidelberg.

Kielhorn, F. (1888). *Grammatik der Sanskrit-Sprache*. Dümmler Verlag, Berlin.

Kulikov, L. (2009). Evolution of case systems. In Andrej Malchukov et al., editors, *The Oxford Handbook of Case*, pages 439–457. Oxford University Press.

Kulkarni, A. and Shukla, D. (2009). Sanskrit morphological analyser: Some issues. *Indian Linguistics*, 70(1-4):169–177.

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289.

Lowe, J. J. (2015). The syntax of Sanskrit compounds. *Language*, 91(3):71–115.

Macdonell, A. A. (1916). *A Vedic Grammar for Students*. Clarendon Press, Oxford.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. In *Proceedings of ICLR Workshop*.

Pradhan, S. S., Ward, W., and Martin, J. H. (2008). Towards robust semantic role labeling. *Computational Linguistics*, 34(2):289–310.

Roth, M. and Lapata, M. (2016). Neural semantic role labeling with dependency path embeddings. In *Proceedings of the 54th Annual Meeting of the ACL*, pages 1192–1202.

Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Tieleman, T. and Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*.

Tikkanen, B. (1991). On the syntax of Sanskrit gerund constructions: A functional approach. In Hans Heinrich Hock, editor, *Studies in Sanskrit Syntax. A Volume in Honor of the Centennial of Speijer's Sanskrit Syntax (1886-1986)*, pages 197–207. Motilal Banarsidass Publishers, Delhi.

van Nooten, B. and Holland, G. (1994). *Rig Veda: A Metrically Restored Text with an Introduction and Notes*. Harvard oriental series. Harvard University.

Witzel, M. (1995). Early Indian history: Linguistic and textual parametres. In George Erdosy, editor, *The Indo-Aryans of Ancient South Asia. Language, Material Culture and Ethnicity*, volume 1, pages 85–125. Walter de Gruyter, Berlin, New York.