Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 5137–5146
Marseille, 11–16 May 2020

The Treebank of Vedic Sanskrit

Oliver Hellwig, Salvatore Scarlata, Elia Ackermann, Paul Widmer
Department of Comparative Language Science, University of Zurich

Abstract

This paper introduces the first treebank of Vedic Sanskrit, a morphologically rich ancient Indian language that is of central importance for linguistic and historical research. The selection of the 4,000 sentences contained in this treebank reflects the development of metrical and prose texts over a period of 600 years. We discuss how these sentences are annotated in the Universal Dependencies scheme and which syntactic constructions required special attention. In addition, we describe a syntactic labeler based on neural networks that supports the initial annotation of the treebank, and whose evaluation can be helpful for setting up a full syntactic parser of Vedic Sanskrit.

Keywords: Treebank, Vedic Sanskrit, Universal dependencies

1. Introduction

Vedic Sanskrit (VS) is an ancient Indo-Aryan language, one of the oldest transmitted Indo-European languages and the precursor of Classical Sanskrit.[1] The large corpus of Vedic poetry and prose is important for reconstructing the early linguistic history of Indo-European and as a source for socio-cultural developments in South Asia during the second and first millenia BCE. The transmission of VS starts with its most famous text, the Ṛgveda (ṚV), composed presumably in the 2nd millenium BCE and comprising religious hymns. While the second oldest Vedic text, the Atharvaveda, focusses on royal and medicinal rites, the bulk of the following Vedic literature discusses the Vedic ritual and ends, at around 500-300 BCE, with texts that demarcate the transition to the early Buddhist culture (Witzel, 1997; Witzel, 2009).

Inspired in part by the famous Vedic grammarian Pāṇini, Indo-European and Vedistic research have studied the content and the linguistic structure of VS for over 150 years. Most of these studies are, however, based on small subcorpora of VS, often only on parts of the ṚV, and therefore do not cover but a small part of the sociolinguistic and spatiotemporal variation actually encountered in VS. Moreover, the results of previous studies are often difficult to reproduce, when larger, more diverse text samples are considered. The composition of the Vedic treebank (VTB) introduced in this paper is motivated by the need for a resource that can be used for data-driven, quantitatively robust diachronic and synchronic investigations of linguistic phenomena in, and starting with, the oldest layers of VS.

One topic we are particularly interested in is the question whether the early metrical texts show a higher degree of non-configurationality than the later prose texts (Gillon, 1996; Kulkarni et al., 2015; Reinöhl, 2016). Consider the passage Ṛgveda 1.51.5c, where the discontinuous elements of the NP *piproḥ puraḥ* 'the strongholds of Pipru (a man)' are printed in bold:

tvam	**piproḥ**	nṛmaṇaḥ	prārujaḥ	**puraḥ**
you	Pipru	manly	broke	strongholds

"You broke through the strongholds of Pipru, o you of manly mind." (Jamison and Brereton, 2014, 164)

While in the Ṛgveda example the dependent *piproḥ* is separated from its head *puraḥ*, the following prose example shows the elements of the NP *devānāṃ viśaḥ* 'subjects of the gods' in continuous placement (Aitareya Brāhmaṇa 1.9.5):

marutaḥ	vai	**devānām**	**viśaḥ**
Maruts	surely	gods'	subjects

"The Maruts are the subjects of the gods." (Keith, 1920, 113)

But the same prose text shows discontinuous placement as well (Aitareya Brāhmaṇa 7.13.1):

tasya	ha	parvata-nāradau	**gṛhe**	ūṣatuḥ
his		Parvata, Nārada	house	lived

"Parvata and Nārada lived in his house.

While such phenomena were only discussed with manually selected examples in previous Vedistic literature, the VTB will make it possible to assess them on a much larger scale and to extend quantitative research to questions such as change in word order (cmp. Gulordava and Merlo (2015)). The treebank described in this paper was built from scratch, including the compilation of the annotation guideline, but working syntactic parsers for VS are not available. Human resources for this task are limited, because annotating syntactic structures, especially in the oldest texts, requires a thorough knowledge of VS, and native speakers can obviously not be recruited (see the discussion in Saavedra and Passarotti (2014)). Therefore, an important aspect of this paper is the design of a machine learning tool that generates proposals for labeling edges in the dependency trees and that should be easy to retrain while the treebank is growing.

The rest of the paper is structured as follows. After a short overview of related research in Sec. 2., Sec. 3. describes the composition of the VTB, the annotation process and some salient problems we met during the annotation. It also discusses the inter-annotator agreement. Section 4. introduces the syntactic labeler. In Section 5., we summarize the central results of this paper and indicate directions for future research. – The treebank, the annotation guideline and the Python code of the labeler can be

[1] Abbreviations used in this paper: AB: Aitareya Brāhmaṇa, MS: Maitrāyaṇī Saṃhitā, ṚV: Ṛgveda; ŚB: Śatapatha Brāhmaṇa; ŚS: Śaunaka Saṃhitā of the Atharvaveda; VS: Vedic Sanskrit. All citations from Vedic texts are given without accents. Sandhis are resolved in all examples.

found at https://github.com/OliverHellwig/sanskrit/tree/master/papers/2020lrec, and we are planning to integrate the treebank into the next official UD release.

2. Related Work

To our knowledge, there exists no treebank of VS. Dwivedi and Easha (2017) describe a small treebank which contains 230 sentences from a narrative text composed in Classical Sanskrit and annotated according to the UD scheme.[2] This treebank cannot be used for our research, because the linguistic structures of Classical Sanskrit differ strongly from those of VS (comparable to the difference between Old and Early Modern English), so that results found with a treebank of Classical Sanskrit cannot easily be transferred to the earlier level of VS. The University of Hyderabad has released a large sample of texts composed in classical Sanskrit, some of which contain structural annotations of compounds.[3] As compounding is much less relevant in Vedic than in Classical Sanskrit (see Sec. 3.2.) and a mapping from the Hyderabad annotations to the data used in this paper is difficult, we did not use this resource for training the syntactic labeler described in Sec. 4.

Contrary to the situation for other ancient languages such as Latin (Ponti and Passarotti, 2016) or Greek (Prokopidis and Papageorgiou, 2014), working dependency parsers have not been developed or tested for any historical level of Sanskrit. A limited number of papers has dealt with designing such a parser from a Pāṇinian (Huet, 2006; Kulkarni et al., 2019) or purely data-driven perspective (Hellwig, 2009), but the authors did not perform systematic experiments assessing the quality of the proposed models. In addition, these parsers are designed for Classical Sanskrit and expect complete sentences. Both conditions are not fulfilled by our data set. Apart from speeding up the annotation process, the syntactic labeler described in Sec. 4. is therefore also meant to explore meaningful features for developing a complete syntactic parser of (Vedic) Sanskrit in the future.

3. Treebank

3.1. Text selection

The composition of the Vedic treebank is primarily motivated by research questions about word order and configurationality in Vedic. Previous non-quantitative studies restricted themselves to descriptive prose texts when studying such phenomena (see, e.g., Delbrück (1888, 15ff.) or Speyer (1896, 76ff.)). These studies often came to the rather general conclusion that there exist certain preferences in the word order of Vedic prose, which may, however, vary substantially across genres, texts, and linguistic conditions (e. g., pragmatic structure, clause and phrase type).

In order to obtain large-scale quantitative, less biased data for exploring the full range and amount of these phenomena, the VTB contains annotations of continuous text samples from the oldest layer of Vedic prose, which may have been composed between 1000 and 700 BCE in Northern India. As a contrast group, we added text samples from the two oldest metrical texts, the Ṛgveda (1300–1000 BCE) and the Śaunaka Saṃhitā of the Atharvaveda, whose metrical parts are largely contemporaneous with the later sections of the Ṛgveda. Thus, the VTB reflects the linguistic development of Vedic over a time range of approximately 600 years.

Text	# Sen.	# Tokens
Metrical		
ṚV	298	2042
ŚS	1108	7107
Prose		
MS	635	3481
AB	1503	10775
ŚS 15	248	2321
ŚB	212	1454
	4004	27180

Table 1: Composition of the Vedic treebank, in approximately descending chronological order

The **prose sections** are collected from the following sources:

- The Maitrāyaṇī Saṃhitā (MS) is the oldest text of the Yajurveda tradition. It contains metrical hymns to be recited during rituals along with their prose explanations (Amano, 2009). MS 2.5.1–11, a discussion of optional sacrifices (*kāmyeṣṭi*), has been annotated completely.
- The Aitareya Brāhmaṇa is generally assumed to belong to the oldest layer of Brāhmaṇa prose, which is slightly younger than the Saṃhitā prose of the MS (Witzel, 1995, 113). The annotation covers AB 1.1–30 and AB 2.1–19, where the performance of the Soma ritual is described, as well as the tale of Śunaḥśepa in AB 7.13-18, which belongs to a younger layer of Brāhmaṇa prose.
- The 15th book of the ŚS contains the earliest known description of the vrātyas, a sodality worshipping the Vedic god Rudra (Falk, 1986).
- ŚB 1.8.1 relates the story of Manu and the fish, a variant of the deluge tale, and is presumably the latest text sample in the VTB. While the extracts from the MS and the Aitareya Brāhmaṇa use a more formal, exegetical style, the sample from the ŚB is a narrative text that includes dialogues, and thus represents a different style of Vedic prose.

The samples of **metrical texts** are taken from the Ṛgveda and the metrical part of the Śaunaka Saṃhitā (first Grand Division mainly; ŚS). The metrical parts of the Śaunaka Saṃhitā are slightly older than the oldest Vedic prose, but may contain younger linguistic material as well (Witzel, 1995, 113). Table 1 gives an overview of the composition of the VTB in its current state.

3.2. Annotation

We use the main syntactic relations defined in the Universal Dependencies standard, v. 2.0 for dependency annotation

[2] https://github.com/UniversalDependencies/UD_Sanskrit-UFAL

[3] http://sanskrit.uohyd.ac.in/Corpus/

(Nivre et al., 2017). The last column in Tab. 8 records the number of annotated instances per dependency relation.

While the UD standard covers most of the syntactic phenomena found in our texts, five structures required special attention during annotation. **(Head) ellipsis**, though being covered by the UD standard, occurs significantly more often in our texts than, for instance, in non-literary English or German.[4] The preference for ellipsis is partly caused by the poetic, often enigmatic diction of Vedic poetry (see Fig. 1a and the translation from Jamison and Brereton (2014) for an example); but it is also found in the prose texts, which tend to abbreviate parallel enumerations as much as possible.

The markup of **compounds** made it necessary to deviate from the UD standard. While English noun compounds mostly consist of two members, post-Ṛgvedic Sanskrit increasingly uses compounds to express complex syntactic structures (Lowe, 2015). The information that a sequence of lexical units forms a compound structure can be deduced from the morphological annotation extracted from the DCS (see Sec. 3.4.), and therefore does not need to be encoded in the dependency annotation. We therefore decided to annotate compounds as if their elements occurred in non-composed form. In Fig. 1b, for example, *jā* 'born from', the final element of the compound *vāta-abhra-jāḥ* 'born from wind and clouds', retains the argument information of the verb *jan* 'be born', from which it is derived. As a consequence, 'wind' and 'clouds' are connected to *jāḥ* with the relation `obl`, which would be used to annotate the respective argument of the verb *jan*. As a consequence, most UD relations can also be used when annotating compounds.

In the oldest metrical texts, **preverb particles** are often separated by several words from the verb with which they enter into a grammatical relation (tmesis; see the phrase *sam ... gamemahi* in Fig. 2, which would read *śrutena saṃgamemahi* in later Vedic and Classical Sanskrit). This phenomenon gradually disappears in post-Ṛgvedic texts, which regularly prefix the preverb to the verbal stem (preverbification). There has been a long and still unsettled discussion about the precise syntactic and semantic relations between preverbs and verbs in Vedic, both from a descriptive and diachronic perspective (Hettrich et al., 2004). Contrary to the UD standard which uses `compound:prt` for comparable phenomena e.g. in German, we decided to label preverbs in tmesis as `advmod`, as they often have adverbial function in the oldest layers of Vedic.

Sanskrit does not distinguish formally between **direct and indirect speech**. Instead, both alternatives are expressed using the quotation marker *iti*, which immediately follows the quoted speech, either as ([speech verb] [statement] *iti*) or ([statement] *iti* [speech verb]). The statement component can consist of multiple and/or nested clauses. We annotate the statement as `ccomp` of the speech verb, link the particle *iti* as `mark` to the rightmost independent verb in the statement, and link multiple sentences in direct speech with `parataxis` to their rightmost element.

Being a pro-drop language, Vedic often omits the subject controlling verbal agreement in a clause. **Secondary predicates** cannot be connected with an element of the clause in this case, because their head (the subject) is not overtly expressed. Since many secondary predicates have an adverbial meaning, we decided to use preferably `advcl` in such cases. In formal terms, such expressions are often undistinguishable from subjects, and we leave it open to the annotator to label them as `csubj`, if required by the context. Consider the following phrase, in which the subject 'he' is not overtly expressed:

paśu-	kāmaḥ	yajati
cattle-	desire	sacrifices

Depending on the textual context, the compound *paśu-kāmaḥ* 'wanting cattle' can be connected as `advcl` or `csubj` to its head *yajati*, resulting in the English translations 'he sacrifices because he wants cattle' or 'the one who wants cattle performs a sacrifice', respectively.

Rev.	LAS	UAS	LOS
–	0.704 [0.716]	0.790 [0.811]	0.764 [0.785]
1	0.755 [0.771]	0.853 [0.873]	0.770 [0.797]

Table 2: Cohen's kappa and proportional agreement (in square brackets) for the labeled attachment score (LAS), unlabeled attachment score (UAS) and label-only score (LOS) before revising the annotations (first row) and after the first revision (second row)

3.3. Annotation workflow and quality

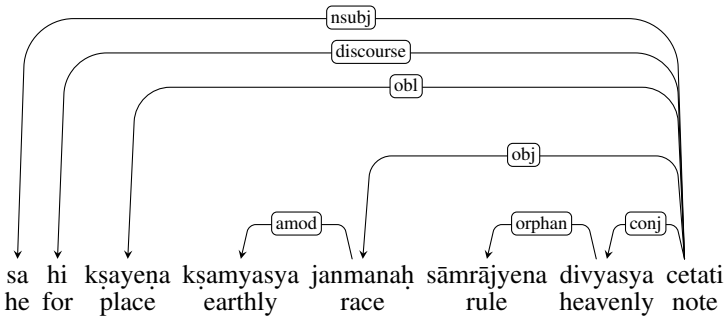
Syntactic labeling was performed by three authors of this paper, each of whom annotated a different part of the VTB. One annotator is a specialist in Ṛgvedic studies, one has a general knowledge of Vedic without further specialization, and the third one is a PhD student. All annotators hold degrees in Indian and/or Vedic Studies. During the whole annotation process, critical decisions (see Sec. 3.2.) were discussed, and the decisions were documented in a guideline. We are aware that this workflow is not optimal, as we cannot determine the inter-annotator agreement on large samples; but, as stated in Sec. 1., this approach was the only feasible one given the lack of qualified annotators and time restrictions. We counter-checked our annotations using accepted modern translations of the texts in the VTB.[5]

In order to obtain an approximation of the IAA, the first two annotators twice re-annotated thirty randomly selected sentences annotated by the other annotator, once before the first revision and once after. The results in Tab. 2 show that the agreement clearly rises between these revisions. While the UAS is close to the value reported by Bamman et al. (2009) for Ancient Greek (87.4%)[6], LOA and LAS

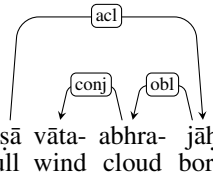
[4] While we observe 0.6% of `orphan` labels among all labels in the VTB, non-literary German (de_hdt-ud-train-b) has 0.011% and non-literary English (en_ewt-ud-train) has 0.014%. The rates rise for modern literary texts in German and especially for Greek (0.361%) and Latin (0.371%). Krisch (2009) discusses ellipsis in ancient Indo-European languages from a linguistic perspective.

[5] The following translations were used: Ṛgveda: Jamison and Brereton (2014); Atharvaveda: Whitney and Lanman (1905); Maitrāyaṇī Saṃhitā: Amano (2009); Aitareya Brāhmaṇa: Keith (1920); Śatapatha Brāhmaṇa: Eggeling (1882 1900)

[6] It is not clear if Bamman et al. (2009) report Cohen's kappa or the uncorrected rate of agreement.



(a) Promotion of the dependents of the elided verb *cetati* at ṚV 7.46.2ab ("For in consequence of his dwelling place he takes cognizance of the earthly race and, in consequence of his universal rule, of the heavenly.").



(b) Annotation of a compound; "a bull born from wind and clouds"

Figure 1: Sample annotations

are clearly below the respective values for Ancient Greek (85.3% and 80.6%).

Further analysis showed four main sources for annotator disagreement. The first source, secondary predication with non-overt subjects has already been discussed in Sec. 3.2. Second, it was not always clear which elements should be labeled `root` and `nsubj` in copular clauses (potentially even with zero copula). Consider Śaunaka Saṃhitā 5.5.8ab, where the unclear context makes it difficult to decide for topic and comment (thema/rhema):

silācī	nāma	kānīnaḥ	pitā	tava
Silācī	by name	of a young woman	father	your

While the first two words of this line, which refer back to a woman mentioned earlier, can safely be translated as '[You are] Silācī by name', it is not clear if the 'father' or the '[person] born by a young woman' should be set as the root of the second sentence. Third, we found that the labeling of the accusative singular neuter of pronouns and adjectives was often disputed, because these elements can express direct objects as well as adverbial modifications (Maitrāyaṇī Saṃhitā 2.5.2):

yat	prathamam	tamaḥ	apāghnan
when/ what	first.ACCSG	darkness.ACCSG	they removed

Here, *prathamam* was connected with `acl` to *tamaḥ* by one annotator ('when they removed the darkness as the first [of a number of items to remove]'), while the other annotator chose an adverbial meaning by connecting it with `advmod` to the verb *apāghnan* ('when they first [in a number of trials] removed the darkness'). Fourth, the two annotators often disagreed about labeling particles as `advmod` or `discourse`; existing handbooks of Sanskrit syntax as Delbrück (1888) are not really helpful in resolving these problems. In all cases, disagreement arises because of competing content-related options, not because of uncertainties concerning formal analyses. While we discussed and adjudicated such ambiguous cases for the IAA subset, eventually updating the guidelines, the final decision is left to the respective annotator in the default workflow.

3.4. Text data and annotation interface

Within specific lexical, phrasal and clausal domains, Sanskrit merges invidual words into longer strings using a set of phonetic rules called Sandhi (Whitney, 1879, 33ff.), so that Sanskrit texts need to be split into words before word based annotations can be added. Instead of manually splitting the texts mentioned in Sec. 3.1., we attach the dependency annotations on top of the Sandhi-split texts provided by the Digital Corpus of Sanskrit (DCS[7], Hellwig (2019)). The DCS provides lemmatized texts with manually validated morphological information, along with POS tags that are automatically induced from morpho-lexical information (Hellwig et al., 2018).

Similar to other ancient languages (see, e.g., Guibon et al. (2014) or Zemánek (2007)), Sanskrit texts do not demarcate sentence boundaries in a consistent manner (Hellwig, 2016). Metrical texts mark metrical units, which often coincide with sentence boundaries, while boundary marking in prose texts depends, more or less, on the personal preferences of modern editors. Therefore, a single text line may consist of multiple sentences, and sentences may transgress boundary markers in metrical texts. We decided to adhere to the form of the edited texts as closely as possible in the annotation interface, in order to facilitate philological research. Therefore, we allow individual text lines to contain more than one syntactic root, or sentences to extend over multiple text lines.

Although brat (Stenetorp et al., 2012) or WebAnno (Eckart de Castilho et al., 2016) provide APIs for task specific actions, we decided to set up a lightweight web-based interface for collaborative dependency annotation using PHP, JQuery and Ajax and to integrate the labeler into it (see Fig. 2). The Sandhi-split words along with their morpho-lexical analyses are stored in a MySQL database and are loaded dynamically on user request. Edges between words are created by dragging the dependent of a syntactic relation on its head term. This dragging event asynchronously calls the syntactic labeler described in Sec. 4., which generates a popup dialog containing the sorted proposals for the edge annotation along with their probabilities. Figure 2 shows the complete annotation of the

[7]Data dump available at `https://github.com/OliverHellwig/sanskrit/tree/master/dcs/data`.

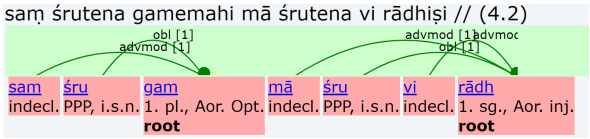


Figure 2: Annotation of the hemistych ŚS 1.1.4cd (*sam śrutena gamemahi mā śrutena vi rādhiṣi* 'May we be endowed with knowlegde. Let me not be deprived of knowledge.') in the annotation interface, illustrating the split into two sentences induced by the depedency annotation

text line ŚS 1.1.4cd and demonstrates how two sentences emerge from the dependency annotation of a text line without sentence boundaries. Note that the published data in conllu format contain only one syntactic root per chunk.

4. The syntactic labeler

This section describes the syntactic labeler that is integrated in the annotation interface. Its design reflects how data are annotated: The annotator has chosen the syntactic dependent and is about to connect it with its head. The task of the labeler is to decide which label should be chosen for this new edge in the dependency graph.

The development of the labeler is guided by three requirements. First, while producing reliable results, it should be fast to train and generate predictions quickly while the annotation proceeds. Second, it should use deep learning techniques, which have been shown to outperform flat discriminative techniques in most areas of NLP. Third, it needs to be integrated in our web-based architecture for collaborative annotation and thus adhere to a client-server model. Given these requirements, we decided to implement the labeler in `tensorflowJS`.[8] Note that the experiments reported in this section were performed with `tensorflow` in Python, but they use a subset of functions that is also available in `tensorflowJS`.

As mentioned in Sec. 2., we consider the development of the labeler as a pre-study for implementing a full syntactic parser of VS. The evaluation of the results therefore concentrates on the question which features have the highest discriminative power in syntactic edge labeling.

4.1. Features

The labeler makes predictions on the basis of n-grams of morpho-lexical features. These features are extracted from the DCS (see Sec. 3.4.) and therefore represent manually validated information. We use the following basic features provided by the DCS:

- The lemma and POS tag (on which see Hellwig et al. (2018)) of each word.
- For nouns, adjectives and nominalized verbal forms (e.g. participles): case, number, gender
- For finite verbal forms: person, number, tense; passive or active voice
- For nominalized verbal forms, we additionally extract the type of the form (participle, past participle, gerund, absolutive, infinitive). This information appears relevant, because these types can be strongly correlated with syntactic relations, as, for example, participles are mostly annotated as `acl`.
- If the head and the dependent have nominal inflection, additional binary flags indicate their case, number and gender agreement.
- A binary flag indicates if the dependent occurs to the left or the right of its head.

[8] `https://www.tensorflow.org/js`

Unigrams
lemma; POS; case; number; gender; verb: person, tense, active/passive; type of finite verbs; case, number and gender agreements; left/right of the head
Bigrams
$\text{POS}_i\ \text{POS}_{i+1}$, $\text{POS}_i\ \text{case}_{i+1}$, $\text{POS}_i\ \text{num}_{i+1}$, $\text{case}_i\ \text{POS}_{i+1}$, $\text{case}_i\ \text{case}_{i+1}$, $\text{num}_i\ \text{POS}_{i+1}$, $\text{num}_i\ \text{num}_{i+1}$ $\text{case}_i\ \text{num}_i$
Trigrams
$\text{POS}_{i-1}\ \text{POS}_i\ \text{POS}_{i+1}$, $\text{case}_{i-1}\ \text{case}_i\ \text{case}_{i+1}$ $\text{case}_i\ \text{num}_i\ \text{gen}_i$

Table 3: N-grams of basic features used for the syntactic labeler; num = number. In order to save space, bigrams of the form $\text{A}_i\ \text{B}_{i+1}$ imply that $\text{B}_{i-1}\ \text{A}_i$ is also used.

Following Chen and Manning (2014) and Shen et al. (2016), we create embeddings of the basic features for individual words as well as for contextual bi- and trigrams. This means that for word w_i at position i in a text line, bigram features are created for w_{i-1}, w_i and w_i, w_{i+1}, and a trigram feature for w_{i-1}, w_i, w_{i+1}. Table 3 shows the evaluated combinations, whose influence on the classification accuracy is evaluated in Sec. 4.3.2.

The lemma information is encoded in pre-trained distributed word embeddings, which are created by running word2vec (Mikolov et al., 2013) on the complete DCS, including late Vedic and Classical Sanskrit texts. We explore four settings for adapting these pre-trained embeddings to syntactic labeling. Apart from leaving the embeddings unchanged (setting 1, -emb -lex), we back-propagate the network error to the embedding layer (setting 2, +emb -lex). Note that embeddings of words not contained in the training set are not adapted in this setting. In setting 3 (-emb +lex), the embeddings are held constant, but a fully connected layer is inserted after the embedding layer, which is expected to encode adaptations of the embeddings required for syntactic labeling and to share these adaptations with words that are not observed during training. Setting 4 (+emb +lex) combines settings 2 and 3 by adapting the embeddings and using the additional fully connected layer.

4.2. Models

The `feedforward` model is a feed-forward neural network (see Fig. 3a). The input concatenates the feature embeddings (i.e. the pre-trained word embeddings and the embeddings of the features according to Tab. 3) of the words corresponding to the dependent and head nodes, and is fed through two hidden layers with $\tanh$ activations and

dropout regularization (Srivastava et al., 2014). The dependency labels are predicted using a softmax layer with cross-entropy loss.

While the input of the `feedforward` model incorporates contextual information by using bi- and trigrams of atomic features, one may hypothesize that a representation of the complete sentence may help in labeling long-range dependencies and improve the performance on critical oppositions such as `xcomp` vs. `acl`, where the correct decision depends on the syntax and semantics of the surrounding text (see Delbrück (1888), Reinöhl (2016)). Therefore, we additionally evaluate two sequential models that combine the dependent-head information of `feedforward` with a representation of the containing text line.[9]

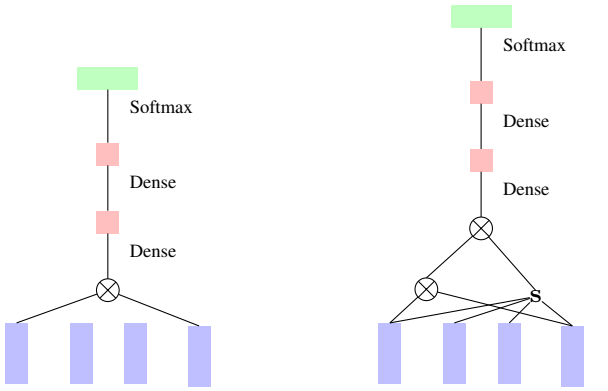
All sequential models use the same overall architecture shown in Fig. 3b, but differ in how the representation of the text line is created. Let $\mathbf{W} \in \mathbb{R}^{L \times F}$ denote the input representation of a text line with L words, each of which is described by the concatenation of its feature embeddings with a total length F. Now, each sequential model creates a sentence embedding vector $\mathbf{s}$ from $\mathbf{W}$:

- `sum` sums up the concatenated embeddings along the sentence axis, and passes the resulting vector of length F through an additional feed-forward layer $f(\dots)$ with tanh activation: $\mathbf{s} = f(\sum_l^L W_{l*})$. Although this model of the sentence context may appear simplistic, generating sentence embeddings from the sums or averages of their word embeddings turned out to be a good baseline in several tasks (see, e.g. Kenter et al. (2016)).
- `bidirnn` uses a bidirectional recurrent neural network (Schuster and Paliwal, 1997) with LSTM units (Hochreiter and Schmidhuber, 1997). The concatenated feature embeddings are fed elementwise into the input layer of the RNN. The sentence embedding $\mathbf{s}$ consists of the concatenated outputs of the forward and backward layers of the network, i.e. of the output of time step L of the forward layer and time step 1 of the backward layer. – We also experimented with stacked bidirectional RNNs (see, e.g. Kiperwasser and Goldberg (2016)), but could not observe better performance, presumably due to the limited size of the training set (details not reported).

4.3. Experiments

4.3.1. Model comparison

We start the evaluation by comparing the three architectures introduced in Sec. 4.2. using a 5×2 cv test (Dietterich, 1998). All models are trained for 20 iterations with the Adagrad algorithm (Duchi et al., 2011). We did not specifically optimize the model architecture (sizes of embeddings and hidden layers, activation functions) nor the hyperparameters.[10] Although Tab. 4 shows that the



(a) Baseline model (b) Sequential models

Figure 3: Predicting the dependency label (`vocative`) for *jyāke* → *nama* in the sentence *jyāke pari ṇo nama* 'O bowstring, bend around us.' The node labeled $\mathbf{s}$ in Fig. 3b contains the representation of the containing text line.

`feedforward` model performs worst of all models, neither of the 5×2 tests that contrast pairs of models produces a p-value lower than 10%, indicating that the differences observed in Tab. 4 are not systematic. Notably, the recurrent archictecture of `bidirnn`, which has produced state of the art results in many NLP tasks, did not outperform the other models at a significant margin. This somehow unexpected result can probably be explained by the fact that most dependency arcs are short. In 83% of all cases, there are maximally two other words between the dependent and the head of a syntactic relation. In theses cases, bi- and trigram features of the `feedforward` model cover the part of the phrase between dependent and head, and the sequential models, which encode the full text line, do not seem to provide relevant additional information. This hypothesis is supported when the classification results of `feedforward` and `bidirnn` are stratified by the distance between dependent and head, and the stratified results are compared using the McNemar test. While the test shows no significant difference between `feedforward` and `bidirnn` for syntactic arcs shorter than four words ($p = 0.68$), the difference is significant with $p = 0.008$ for longer arcs, although the accuracy differs by less than 1%.

Comparing the quality of our labeling models with those described by previous research is complicated. Our labeler works with morpho-lexical gold data. On the other hand, other approaches often use much larger treebanks for training as well as the unsupervised output of (state of the art) POS taggers and lemmatizers, whose error levels are low for well-resourced languages such as English. Papers dealing with the re-labeling of complete trees are also not easily comparable, because they can model interdependencies between the proposed labels using a Markov chain assumption (see, for instance, the CRF output layer used by Shen et al. (2016)). Taking these restrictions into account, the `feedforward` model obtains a micro-averaged PRF of 81.3%/77.6%/78.9% and an overall accuracy of 84.0%

[9] As mentioned in Sec. 3.2., Sanskrit has no reliable sentence boundary markers. We therefore use text lines as the closest approximations of the true sentences.

[10] We use the following settings: size of the first dense layer in Fig. 3: 20; of the penultimate layer: 40; batch size: 32; dropout rate: 20%; learning rate: 0.001.

Model	P	R	F	A	A@3
feedforward	79.43	74.95	76.59	82.35	95.14
sum	**79.62**	75.19	76.79	**82.78**	**95.16**
bidirnn	79.50	**75.45**	**76.92**	82.75	95.13

Table 4: Micro-averaged p(recision), r(ecall), F (score), a(ccuracy) and accuracy @3 for the models described in Sec. 4.2. All features described in Sec. 4.1. (including the +lex +emb adaptation) are activated for these experiments. The four models are compared using a 5×2 cv test (Dietterich, 1998). None of the pairwise tests produced a p-value that is significant at the 10% level.

Feature	P	R	F	A
-lex -emb	-2.69	-2.05	-2.09	-1.43***
+lex -emb	-0.46	-0.69	-0.66	-0.31***
-lex +emb	-3.01	-1.59	-1.83	-1.37***
Unigrams	-6.30	-6.74	-6.94	-3.63***
Bigrams	-2.12	-1.49	-1.83	-0.87***
Trigrams	-0.85	-0.77	-0.94	-0.31***

Table 5: Feature ablation study with the `feedforward` model. The first column indicates which feature was deactivated. Asterisks and dots after the accuracy value indicate the significance levels of a McNemar χ^2 test ($< 0.001, < 0.01, < 0.05, < 0.1$) that compares the respective model with full sum model.

when tested with a tenfold cross-validation (detailed data in Tab. 8). These results appear acceptable given the current size of the treebank.

When considering the tradeoff between accuracy, training duration (which is by far longest for `bidirnn`) and simplicity of the architecturee, the `feedforward` model combines high accuracy with a simple context representation and high training speed. We therefore use this model in the following experiments.

4.3.2. The influence of features

In order to estimate the influence of individual features, we perform an ablation study with the `feedforward` model. The statistical significance of differences to the full baseline model (all features activated) are assessed using the McNemar χ^2 test statistics. The results shown in Tab. 5 demonstrate that the adaptation of the pretrained lexical embeddings (-emb +lex), unigrams and bigrams have the strongest effect on the classification accuracy, while the influence of the embedding adaptations and the trigrams is limited. We hypothesize that the trigram features are too sparse given the current size of the VTB and therefore do not improve the labeling accuracy.

Because Tab. 5 shows that the unigrams have the strongest influence on the quality of the model, we examine their individual influence in a second ablation study. Here, the `feedforward` model uses full lexical adaptation (+lex +emb), but bi- and trigrams are discarded completely, because they contain the unigram information. The results in Tab. 6 show that cases and lexical information are, by a large margin, most relevant for syntactic labeling, followed by the binary flag indicating the relative position

Feature	P	R	F	A
lemma	-6.82	-6.84	-7.29	-2.35***
POS	-1.67	-0.62	-0.65	-0.79***
case	-9.00	-8.67	-8.72	-11.99***
number	-0.44	+0.46	+0.34	-0.05
gender	+0.01	+0.63	+0.60	-0.17.
verb: person	-0.25	-0.19	-0.06	-0.07
verb: tense	-0.71	+0.17	+0.11	-0.02
verb: passive	-0.43	+0.16	+0.10	+0.04
verb: nomin.	+0.01	-0.30	-0.36	-0.13
case agr.	-0.58	-0.36	-0.36	-0.28**
number agr.	-1.05	-0.23	-0.34	-0.31***
gender agr.	-0.98	-0.48	-0.55	-0.16.
full agr.	-0.62	+1.09	+0.78	-0.01
left/right	-1.16	-0.04	-0.31	-0.92***

Table 6: Feature ablation study for the unigram features. In the delexicalized setting (lemma), all words are set to the UNK tag.

Setting	c h	c ¬h	¬c h	¬c ¬h
+lex +emb	**83.73*****	**80.46**	71.91	73.03
-lex -emb	82.29	79.09	70.28	71.96
-lex +emb	82.42	78.54	69.46	70.89
+lex -emb	83.37	79.59	**72.30**	**73.67**

Table 7: Details for the lexical subtests in Tab. 5. Column labels show if the dependent (resp. head) lemma of a syntactic relation is contained in the training set (c, h) or OOV (¬c, ¬h); ¬c ¬h thus indicates that both dependent and head lemmata are OOV. Highest values per dependent/head combination are printed in bold. The indicator of the significance level after the best value per column, if any, is based on a McNemar test comparing the best with the second best setting (see Tab. 5 for the significance levels).

of the dependent, POS, as well as gender, case and number agreement. Morphological information about the verb plays almost no role. The importance of the case information is obvious, as it is relevant for distinguishing verbal roles (`nsubj` vs. `obj` etc.) as well as for nominal modification.

The important function of the lexicon has been observed in previous research, as, for instance, in the lexicalized parser introduced by Collins (2003). Table 7 therefore gives a more detailed evaluation of the four types of lexical adaptation. Here, we examine how the adaptation type (+lex -emb etc.; see Sec. 4.1.) influences the classification result if the dependent or head of a syntactic relation are OOV. First, one can observe that relations in which the dependent is OOV (¬c x) are clearly harder to classify than those in which the head is OOV (x ¬h). Second, Tab. 7 shows that the fully connected layer inserted for lexical adaptation (+lex x) works as intended: The two best settings (+lex +emb, +lex -emb) use this kind of lexical adaptation. Besides, this layer produces the best results when the dependent lemma is OOV, indicating that this layer transfers syntactic information to OOV words.

Returning to Tab. 6, a detailed evaluation shows that the

model without the left/right flag makes numerous errors when distinguishing `advmod` from `discourse` elements. This can be explained by our annotation practice, because we label enclitic particles (dependent directly to the right of its head) such as *vai* 'surely' or *ha* 'indeed' mostly as `discourse`, while adverbial particles such as *iha* 'here' often appear to the left of their heads (mostly the verbs). Given the limited amount of training data, the difference between these two kinds of particles has apparently not been captured in the embeddings, but is rather induced from the relative positions. We also found notable the high relevance of gender information. Here, direct objects of a verb (`obj`) were frequently misclassified as `advmod` by the model without gender information. VS can use pronouns and adjectives in accusative singular neuter as adverbial modifiers as in the sentence *tad yaḥ . . . yajate* 'who therefore sacrifices . . . ', where *tad* is the acc. sg. neuter of the pronoun *tad* (see the discussion of the annotator disagreement in Sec. 3.3.).

4.3.3. Error analysis

In order to estimate how the `feedforward` model performs in real prediction tasks, we performed a tenfold cross-validation, the results of which are displayed in Tab. 8. In general, we observe a weak positive correlation between the number of annotated instances and the F-score of each syntactic relation ($\tau = 0.39$, $p = 0.056$).

Notably, many errors were made in copular sentences with zero copula, in which the subject is not correctly labeled. Since the subject and the root agree in case (and often in number as well) in such sentences, a frequent mislabeling is `conj` as at Aitareya Brāhmaṇa 1.29.7: *somaḥ vai rājā induḥ* 'The drop [is] indeed King Soma'. Here, the arc from *soma* to *indu-* 'drop' should be labeled as `nsubj`, but is as `conj`. We also observe high confusion rates with `xcomp`, as at Aitareya Brāhmaṇa 2.8.3: *tasmāt ajaḥ medhyaḥ abhavat* 'Therefore, the goat became ritually clean', where *aja* 'goat' is wrongly labeled as `xcomp`; all labelers propose the correct `nsubj` as their second best options. Note that the analysis as `xcomp` would also result in a meaningful sentence ("therefore, he became a ritually pure goat"), which does, however, not fit into the textual context.

Another frequent error is `amod` instead of `acl`, which reflects the formal underspecification of the differences between attributive adjectives and secondary predicates (see Sec. 3.2.). Take, for example, the following sentence from Maitrāyaṇī Saṃhitā 2.5.3 which relates how cattle emerges from the killed demon Vṛtra:

tasmāt	viṣvañcaḥ	paśavaḥ	vyudāyan
from him	moving to all sides.ADJ	cattle	went out

'The cattle went out of him to all sides.'

Although *viṣvañcaḥ* is morphologically an adjective, it serves as an event-oriented secondary predicate; a clunky translation would render the sentence as 'the cattle went out of him as moving to all sides'. This error also occurs when words that are listed as adjectives in the dictionary are actually frozen verbal forms and could thus also be read as compounds involving a participle. Such a case occurs at Śaunaka Saṃhitā 15.13.5c, where the gold annotation labels *aparimitāḥ lokāḥ* 'infinite worlds' as `acl`, because *aparimita* 'infinite' is the negated past participle of *pari-mā* 'measure' (note that Latin 'infinitus' is derived in a similar way from the verb *finire*).

Relation	P	R	F	Freq.
acl	77.79	76.22	76.99	1190
advcl	73.69	73.24	73.47	826
advmod	84.85	87.01	85.92	3103
amod	75.60	75.88	75.74	792
appos	64.73	55.14	59.56	243
case	79.70	84.05	81.82	257
cc	94.94	96.16	95.55	625
ccomp	80.12	73.57	76.70	367
conj	75.70	77.40	76.54	1743
cop	91.97	97.52	94.66	282
csubj	69.23	33.96	45.57	53
det	86.20	86.32	86.26	760
discourse	71.39	69.08	70.22	773
iobj	87.13	89.43	88.26	492
mark	90.17	89.81	89.99	756
nmod	89.45	92.07	90.74	1538
nsubj	87.67	91.97	89.77	3411
nummod	85.20	83.92	84.56	199
obj	86.32	92.51	89.31	2510
obl	89.07	85.16	87.07	1799
orphan	69.01	47.34	56.16	207
parataxis	72.45	47.65	57.49	149
vocative	95.48	99.17	97.29	362
xcomp	73.37	58.70	65.22	460

Table 8: Detailed results of a tenfold cross-validation of the `feedforward` model

5. Summary

This paper has described the composition and annotation of the first treebank of Vedic Sanskrit, which is, at the same time, the first large treebank of a premodern Indian language. While the UD standard, which is used for the annotation, covers most of the syntactic phenomena we encountered in our text sample, the annotation of compounds deviates from the official annotation scheme. Given that Classical Sanskrit often uses nominal compounding to express various kinds of events and states, we consider our decision as an appropriate basis for annotating Classical Sanskrit texts as well.

The paper also discussed a syntactic labeler that greatly sped up the annotation process. The analysis of the results of the labeler provided important clues about which features may be useful for setting up a complete syntactic parser of (Vedic) Sanskrit. As soon as enough data are available, we plan to address this issue by using neural parser architectures developed for morphologically rich languages (e.g. Legrand and Collobert (2016)).

6. Bibliographical References

Amano, K. (2009). *Maitrāyaṇī Saṃhitā I–II. Übersetzung der Prosapartien mit Kommentar zur Lexik und Syntax der älteren vedischen Prosa*. Hempen, Bremen.

Bamman, D., Mambrini, F., and Crane, G. (2009). An ownership model of annotation: The Ancient Greek dependency treebank. In *Proceedings of the Eighth International Workshop on Treebanks and Linguistic Theories*, pages 5–15.

Chen, D. and Manning, C. (2014). A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750.

Collins, M. (2003). Head-driven statistical models for natural language parsing. *Computational Linguistics*, 29(4):589–637.

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

Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, 10(7):1895–1923.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.

Dwivedi, P. and Easha, G. (2017). Universal Dependencies for Sanskrit. *International Journal of Advance Research, Ideas and Innovations in Technology*, pages 479–482.

Eckart de Castilho, R., Mújdricza-Maydt, É., Yimam, S. M., Hartmann, S., Gurevych, I., Frank, A., and Biemann, C. (2016). A web-based tool for the integrated annotation of semantic and syntactic structures. In *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*, pages 76–84, Osaka. The COLING 2016 Organizing Committee.

Eggeling, J. (1882-1900). *The Satapatha-Brāhmaṇa. According to the Text of the Mādhyandina School*. Clarendon Press, Oxford.

Falk, H. (1986). *Bruderschaft und Würfelspiel. Untersuchungen zur Entwicklungsgeschichte des vedischen Opfers*. Hedwig Falk, Freiburg.

Gillon, B. S. (1996). Word order in classical Sanskrit. *Indian Linguistics*, 57(1-4):1–35.

Guibon, G., Tellier, I., Constant, M., Prévost, S., and Gerdes, K. (2014). Parsing poorly standardized language. Dependency on Old French. In *Proceedings of the Thirteenth International Workshop on Treebanks and Linguistic Theories (TLT13)*, pages 51–61.

Gulordava, K. and Merlo, P. (2015). Diachronic trends in word order freedom and dependency length in dependency-annotated corpora of Latin and ancient Greek. In *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*, pages 121–130, Uppsala. Uppsala University, Uppsala, Sweden.

Hellwig, O., Hettrich, H., Modi, A., and Pinkal, M. (2018). Multi-layer annotation of the Ṛgveda. In *Proceedings of the LREC*.

Hellwig, O. (2009). Extracting dependency trees from Sanskrit texts. In Amba Kulkarni et al., editors, *Sanskrit Computational Linguistics. Third International Symposium*, Lecture Notes in Artificial Intelligence, 5406, pages 106–115, Berlin. Springer.

Hellwig, O. (2016). Detecting sentence boundaries in Sanskrit texts. In *Proceedings of the COLING*, pages 288–297.

Hellwig, O. (2019). DCS - The Digital Corpus of Sanskrit. Technical report, Berlin.

Hettrich, H., Casaretto, A., and Schneider, C. (2004). Syntax und Wortarten der Lokalpartikeln im Ṛgveda. IV: I. Allgemeines, II. úpa, III. áva. *Münchener Studien zur Sprachwissenschaft*, 64:17–130.

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

Huet, G. (2006). Shallow syntax analysis in Sanskrit guided by semantic nets constraints. In *Proceedings of the 2006 International Workshop on Research Issues in Digital Libraries*, pages 1–10. ACM.

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

Keith, A. B. (1920). *Rigveda Brahmanas: The Aitareya and Kauṣītaki Brāhmaṇas of the Rigveda*. Harvard University Press, Cambridge, Massachusetts.

Kenter, T., Borisov, A., and de Rijke, M. (2016). Siamese CBOW: Optimizing word embeddings for sentence representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 941–951, Berlin, Germany, August. Association for Computational Linguistics.

Kiperwasser, E. and Goldberg, Y. (2016). Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327.

Krisch, T. (2009). On the "syntax of silence" in Proto-Indo-European. In Roland Hinterhölzl et al., editors, *Information Structure and Language Change New Approaches to Word Order Variation in Germanic*, pages 191–222. Mouton de Gruyter, Berlin, New York.

Kulkarni, A., Shukla, P., Satuluri, P., and Shukl, D. (2015). How free is 'free' word order in Sanskrit? In Peter M. Scharf, editor, *Sanskrit Syntax*, pages 269–304.

Kulkarni, A., Vikram, S., and Sriram, K. (2019). Dependency parser for Sanskrit verses. In *Proceedings of the 6th International Sanskrit Computational Linguistics Symposium*, pages 14–27.

Legrand, J. and Collobert, R. (2016). Deep neural networks for syntactic parsing of morphologically rich languages. In *Proceedings of the ACL*, pages 573–578, Berlin.

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

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

Nivre et al., J. (2017). Universal dependencies 2.0. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Ponti, E. M. and Passarotti, M. (2016). Differentia compositionem facit. A slower-paced and reliable parser for Latin. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 683–688, Portorož.

Prokopidis, P. and Papageorgiou, H. (2014). Experiments for dependency parsing of Greek. In *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages*, pages 90–96, Dublin, Ireland. Dublin City University.

Reinöhl, U. (2016). *Grammaticalization and the Rise of Configurationality in Indo-Aryan*. Oxford University Press, Oxford, UK.

Saavedra, B. G. and Passarotti, M. (2014). Challenges in enhancing the Index Thomisticus treebank with semantic and pragmatic annotation. In Verena Henrich, et al., editors, *Proceedings of the Thirteenth International Workshop on Treebanks and Linguistic Theories (TLT13)*, pages 265–270, Tübingen.

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

Shen, T., Lei, T., and Barzilay, R. (2016). Making dependency labeling simple, fast and accurate. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1089–1094.

Speyer, J. S. (1896). *Vedische und Sanskrit-Syntax*. Grundriss der Indo-arischen Philologie und Altertumskunde, III. Band, Heft A. Verlag von Karl J. Trübner, Strassburg.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., and Tsujii, J. (2012). brat: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107, Avignon, France, April. Association for Computational Linguistics.

Whitney, W. D. and Lanman, C. R. (1905). *Atharva-Veda Samhita*. Harvard University, Cambridge.

Whitney, W. D. (1879). *A Sanskrit Grammar*. Breitkopf and Härtel, Leipzig.

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*, pages 85–125. Walter de Gruyter, Berlin, New York.

Witzel, M. (1997). The development of the Vedic canon and its schools: The social and political milieu (Materials on Vedic Sakhas, 8). In Michael Witzel, editor, *Inside the Texts, Beyond the Texts. New Approaches to the Study of the Vedas*, pages 258–348. Cambridge.

Witzel, M. (2009). Moving targets? Texts, language, archaeology and history in the Late Vedic and early Buddhist periods. *Indo-Iranian Journal*, 52(2/3):287–310.

Zemánek, P. (2007). A treebank of Ugaritic: Annotating fragmentary attested languages. In *Proceedings of the Sixth Workshop on Treebanks and Linguistic Theories (TLT2007)*, pages 213–218.