SanskritShala: A Neural Sanskrit NLP Toolkit with Web-Based Interface for Pedagogical and Annotation Purposes

Jivnesh Sandhan[1], Anshul Agarwal[1], Laxmidhar Behera[1,3], Tushar Sandhan[1] and Pawan Goyal[2]
[1]IIT Kanpur, [2]IIT Kharagpur, [3]IIT Mandi
jivnesh@iitk.ac.in,pawang@cse.iitkgp.ac.in

Abstract

We present a neural Sanskrit Natural Language Processing (NLP) toolkit named SanskritShala[1] to facilitate computational linguistic analyses for several tasks such as word segmentation, morphological tagging, dependency parsing, and compound type identification. Our systems currently report state-of-the-art performance on available benchmark datasets for all tasks. SanskritShala is deployed as a web-based application, which allows a user to get real-time analysis for the given input. It is built with easy-to-use interactive data annotation features that allow annotators to correct the system predictions when it makes mistakes. We publicly release the source codes of the 4 modules included in the toolkit, 7 word embedding models that have been trained on publicly available Sanskrit corpora and multiple annotated datasets such as word similarity, relatedness, categorization, analogy prediction to assess intrinsic properties of word embeddings. So far as we know, this is the first neural-based Sanskrit NLP toolkit that has a web-based interface and a number of NLP modules. We are sure that the people who are willing to work with Sanskrit will find it useful for pedagogical and annotative purposes. SanskritShala is available at: https://cnerg.iitkgp.ac.in/sanskritshala. The demo video of our platform can be accessed at: https://youtu.be/x0X31Y9k0mw4.

1 Introduction

Sanskrit is a culture-bearing and knowledge-preserving language of ancient India. Digitization has come a long way, making it easy for people to access ancient Sanskrit manuscripts (Goyal et al., 2012; Adiga et al., 2021). However, we find that the utility of these digitized manuscripts is limited due to the user's lack of language expertise and various linguistic phenomena exhibited by the language. This motivates us to investigate how we can utilize natural language technologies to make Sanskrit texts more accessible.

The aim of this research is to create neural-based Sanskrit NLP systems that are accessible through a user-friendly web interface. The Sanskrit language presents a range of challenges for building deep learning solutions, such as the *sandhi* phenomenon, a rich morphology, frequent compounding, flexible word order, and limited resources (Sandhan et al., 2022d; Krishna et al., 2021; Sandhan et al., 2021, 2019). To overcome these challenges, 4 preliminary tasks were identified as essential for processing Sanskrit texts: word segmentation, morphological tagging, dependency parsing, and compound type identification. The word segmentation task is complicated by the *sandhi* phenomenon, which transforms the word boundaries (Sandhan et al., 2022d). The lack of robust morphological analyzers makes it challenging to extract morphological information, which is crucial for dependency parsing. Similarly, dependency information is essential for several downstream tasks such as word order linearisation (Krishna et al., 2019) which helps to decode possible interpretation of the poetic composition. Additionally, the ubiquitous nature of compounding in Sanskrit is difficult due to the implicitly encoded semantic relationship between its constituents (Sandhan et al., 2022c). These 4 tasks can be viewed as a preliminary requirement for developing robust NLP technology for Sanskrit. Thus, we develop novel neural-based linguistically informed architectures for all 4 tasks, reporting state-of-the-art performance on Sanskrit benchmark datasets (Sandhan et al., 2022c,d,a).

In this work, we introduce a neural Sanskrit NLP toolkit named SanskritShala[2] to assist computational linguistic analyses involving multiple tasks such as word segmentation, morphological tagging, dependency parsing, and compound type identification. SanskritShala is also deployed as a web

[1] It means 'a school of Sanskrit'.

[2] Roughly, it can be translated as 'a school of Sanskrit'.

application that enables users to input text and gain real-time linguistic analysis from our pretrained systems. It is also equipped with user-friendly interactive data annotation capabilities that allow annotators to rectify the system when it makes errors. It provides the following benefits: (1) A user with no prior experience with deep learning can utilise it for educational purposes. (2) It can function as a semi-supervised annotation tool that requires human oversight for erroneous corrections. We publicly release the source code of the 4 modules included in the toolkit, 7 word embedding models that have been trained on publicly available Sanskrit corpora and multiple annotated datasets such as word similarity, relatedness, categorization, analogy prediction to measure the word embeddings' quality. To the best of our knowledge, this is the first neural-based Sanskrit NLP toolkit that contains a variety of NLP modules integrated with a web-based interface.

Summarily, our key contributions are as follows:

- We introduce the first neural Sanskrit NLP toolkit to facilitate automatic linguistic analyses for 4 downstream tasks (§4).
- We release 7 pretrained Sanskrit embeddings and suit of 4 intrinsic evaluation datasets to measure the word embeddings' quality (§5).
- We integrate SanskritShala with a user-friendly web-based interface which is helpful for pedagogical purposes and in developing annotated datasets (§5).
- We publicly release codebase and datasets of all the modules of SanskritShala which currently mark the state-of-the-art results.[3]

2 Related Work on Sanskrit NLP Tools

Recently, the Sanskrit Computational Linguistics (SCL) field has seen significant growth in building web-based tools to help understand Sanskrit texts. Goyal and Huet (2016a) introduced the Sanskrit Heritage Reader (SHR), a lexicon-driven shallow parser that aids in the selection of segmentation solutions. Samsādhanī is another web-based tool consisting of various rule-based modules (Kulkarni and Sharma, 2019; Kulkarni et al., 2020; Sriram et al., 2023). Recently, Terdalkar and Bhattacharya (2021, 2022) introduced a web-based annotation tool for knowledge-graph construction and a metrical analysis.

In short, tools for NLP can be divided into two groups: rule-based and annotation tools. Rule-based tools have limitations such as not providing a final solution, limited vocabulary coverage, and lacking user-friendly annotation features. Annotation tools, on the other hand, do not have the recommendations of rule-based systems, relying solely on annotators. To address these limitations, a web-based annotation framework called SHR++ (Krishna et al., 2020c) was proposed. It combines the strengths of both types of tools by offering all possible solutions from rule-based system SHR for tasks like word segmentation and morphological tagging, allowing annotators to choose the best solution rather than starting from scratch.

Our proposal, SanskritShala, goes a step further by integrating a neural-based NLP toolkit that combines state-of-the-art neural-based pre-trained models with rule-based suggestions through a web-based interface. Each module of SanskritShala is trained to predict the solutions from the exhaustive candidate solution space generated by rule-based systems. Hence, it makes predictions in real time using neural-based models that have already been trained. Thus, a complete solution is shown to the users / annotators, which was not possible in any of the previous attempts.

Further, annotators can easily correct the mispredictions of the system with the help of user-friendly web-based interface. This would significantly reduce the overall cognitive load of the annotators. To the best of our knowledge, SanskritShala is the first NLP toolkit available for a range of tasks with a user friendly annotation interface integrated with the neural-based modules.

3 About Sanskrit

Sanskrit is an ancient language known for its cultural significance and knowledge preservation. However, it presents challenges for deep learning due to its morphological complexity, compounding, free word order, and lack of resources. Sanskrit's intricate grammar, with its combination of roots, prefixes, and suffixes, requires advanced algorithms to analyze and understand. Compounding adds another layer of complexity as multiple words combine to form new words with unique meanings (Krishna et al., 2016; Sandhan et al., 2022c). The free word order in Sanskrit complicates tasks

[3] https://github.com/Jivnesh/SanskritShala

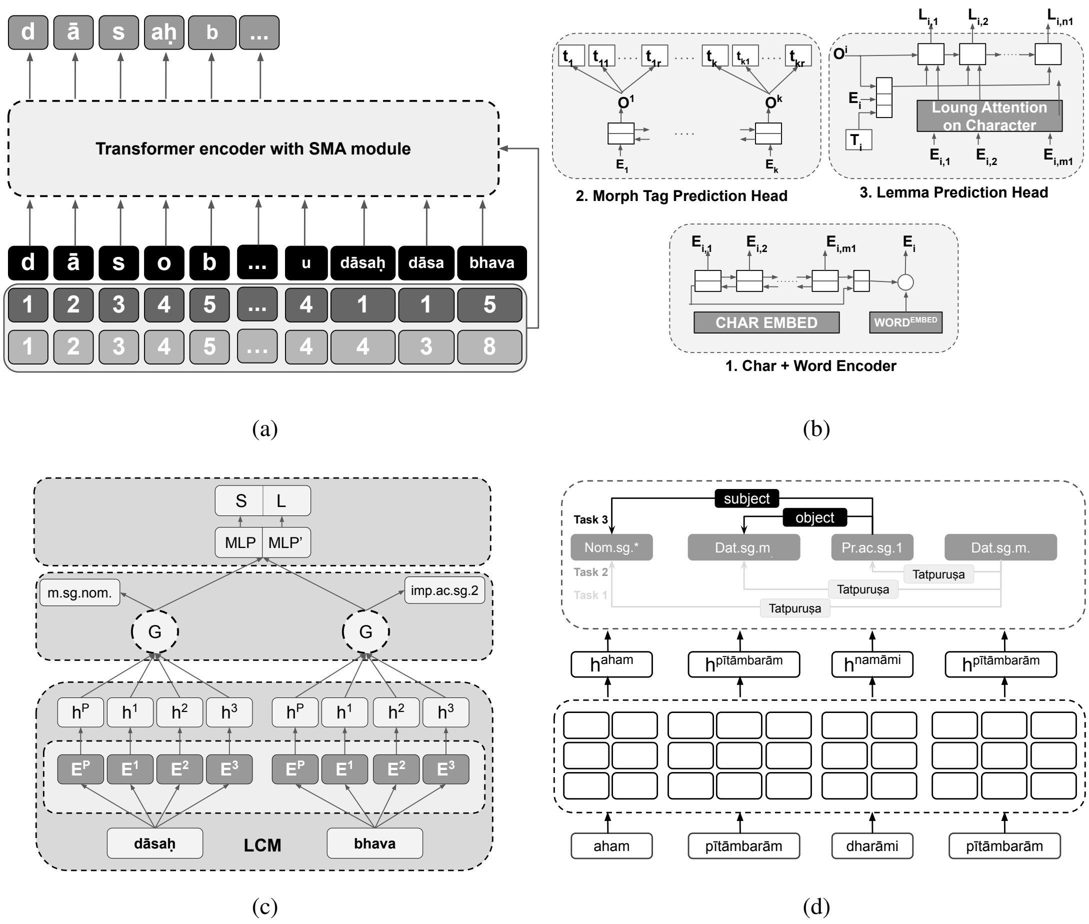


Figure 1: (a) Toy illustration of the TransLIST system."*dāsobhava*". Translation: "Become a servant." (b) LemmaTag architecture in which multi-task learning formulation is leveraged to predict morphological tags and lemmas by employing bidirectional RNNs with character-level and word-level representations. (c) Proposed ensembled architecture for dependency parsing integrated with the LCM pretraining. LCM is acronym for three auxiliary tasks: Lemma prediction, Case prediction and Morphological tag prediction. (d) Toy example illustrating the context-sensitive multi-task learning system: "*aham pīta-ambaram dharāmi*" (Translation: "I wear a yellow cloth") where '*pīta-ambaram*' is a compound having *Tatpuruṣa* semantic class according to the context presented.

like parsing and understanding, requiring models to comprehend meaning regardless of word placement (Krishna et al., 2023, 2019). Moreover, Sanskrit is considered a low-resource language, lacking extensive datasets and pre-trained models (Sandhan et al., 2021). Overcoming these challenges necessitates linguistic expertise, computational techniques, and sufficient language resources. Developing specialized models to handle Sanskrit's morphology, compounding, and word order is essential. Creating annotated datasets, lexicons, and corpora will also contribute to advancing research and applications in Sanskrit (Sandhan et al., 2022b, 2023). Despite the obstacles, utilizing deep learning to explore Sanskrit benefits the preservation of cultural heritage and facilitates a deeper understanding of India's literature and philosophy, while also pushing the boundaries of natural language processing.

4 A Neural NLP Sanskrit Toolkit

In this section, we describe SanskritShala, which is a neural Sanskrit NLP toolkit designed to aid computational linguistic analysis including various tasks, such as word segmentation, morphological tagging, dependency parsing, and compound type identification. It is also available as a web application that allows users to input text and obtain real-time linguistic analysis from our pretrained algorithms. We elucidate SanskritShala by first elaborating on its key modules.

Word Tokenizer: Earlier *lexicon-driven* systems for Sanskrit word segmentation (SWS) rely on Sanskrit Heritage Reader (Goyal and Huet, 2016b, SHR), a rule-based system, to obtain the exhaustive solution space for segmentation, followed by diverse approaches to find the most valid solution. However, these systems are rendered moot while stumbling out-of-vocabulary words. Later, *data-driven* systems for SWS are built using the most recent techniques in deep learning, but can not utilize the available candidate solution space. To overcome the drawbacks of both lines of modelling, we build a **Trans**former-based **L**inguistically-**I**nformed **S**anskrit **T**okenizer (TransLIST) (Sandhan et al., 2022d) containing (1) a component that encodes the character-level and word-level potential candidate solutions, which tackles *sandhi* scenario typical to SWS and is compatible with partially available candidate solution space, (2) a novel soft-masked attention for prioritizing selected set of candidates and (3) a novel path ranking module to correct the mispredictions. Figure 1(a) illustrates the TransLIST architecture, where the candidate solutions obtained from SHR are used as auxiliary information. In terms of the perfect match (PM) metric, TransLIST outperforms with 93.97 PM compared to the state-of-the-art (Hellwig and Nehrdich, 2018) with 87.08 PM.

Morphological Tagger: Sanskrit is a morphologically-rich fusional Indian language with 40,000 possible labels for inflectional morphology (Krishna et al., 2020b; Gupta et al., 2020), where homonymy and syncretism are predominant (Krishna et al., 2018). We train a neural-based architecture (Kondratyuk et al., 2018, LemmaTag) on Sanskrit dataset (Krishnan et al., 2020). Figure 1(b) illustrates the system architecture in which multi-task learning formulation is leveraged to predict morphological tags and lemmas by employing bidirectional RNNs with character-level and word-level representations. Our system trained on the Sanskrit dataset stands first with 69.3 F1-score compared to the second position with 69.1 F1-score on the Hackathon dataset (Krishnan et al., 2020) leaderboard.[4]

Dependency Parser: Due to labelled data bottleneck, we focus on low-resource techniques for Sanskrit dependency parsing. Numerous strategies are tailored to improve task-specific performance in low-resource scenarios. Although these strategies are well-known to the NLP community, it is not obvious to choose the best-performing ensemble of these methods for a low-resource language of interest, and not much effort has been given to gauging the usefulness of these methods. We investigate 5 low-resource strategies in our ensembled Sanskrit parser (Sandhan et al., 2022a): data augmentation, multi-task learning, sequential transfer learning, pretraining, cross/mono-lingual and self-training. Figure 1(c) shows our ensembled system, which supersedes with 88.67 Unlabelled Attached Score (UAS) compared to the state-of-the-art (Krishna et al., 2020a) with 87.46 UAS for Sanskrit and shows on par performance in terms of Labelled Attached Score.

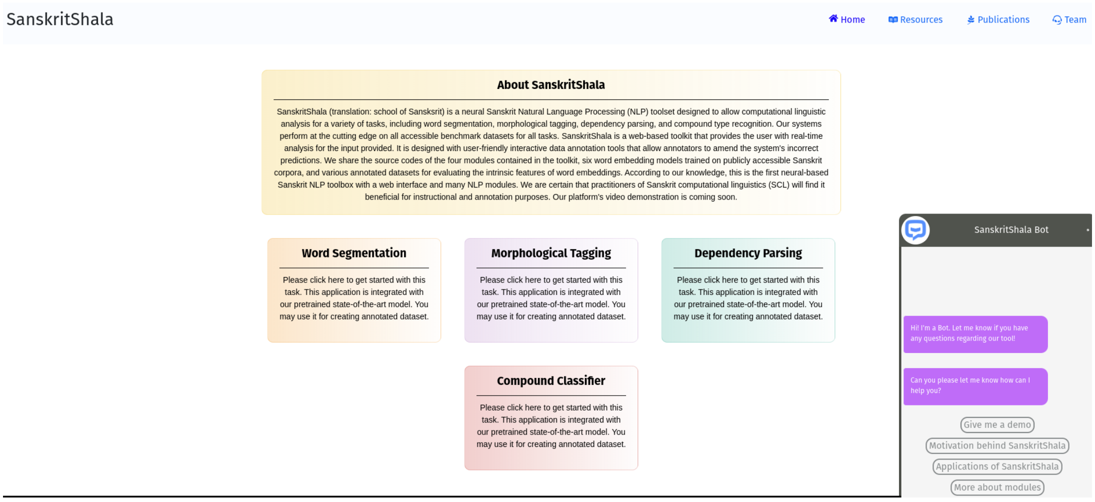
Sanskrit Compound Type Identifier (SaCTI) is a multi-class classification task that identifies semantic relationships between the components of a compound. Prior methods only used the lexical information from the constituents and did not take into account the most crucial syntactic and contextual information for SaCTI. However, the SaCTI task is difficult mainly due to the implicitly encrypted context-dependent semantic relationship between the compound's constituents. Thus, we introduce a novel multi-task learning approach (Sandhan et al., 2022c) (Figure 1(d)) which includes contextual information and enhances the complementary syntactic information employing morphological parsing and dependency parsing as two auxiliary tasks. SaCTI outperforms with 81.7 F1-score compared to the state-of-the-art by Krishna et al. (2016) with 74.0 F1-score.

5 Sanskrit Resources in SanskritShala

In this section, we describe 7 word embeddings pretrained on Sanskrit corpora and suit of 4 intrinsic tasks datasets to assess the quality of word embeddings, followed by the description of web interface.

Pretrained word embeddings for Sanskrit: There are two types of embedding methods: static and contextualized. Table 1 shows how they are categorized based on the smallest unit of input to the embedding model, such as character, subword, or token level. The paper focuses on two token-level word embeddings: Mikolov et al. (2013, word2vec) and Pennington et al. (2014, GloVe). Word2vec is the foundation for all subsequent embeddings and works on the local context window, while GloVe considers the global context. To address the OOV issue, subword (Wieting et al., 2016; Bojanowski

[4] Hackathon leaderboard: https://competitions.codalab.org/competitions/35744#results



About SanskritShala

SanskritShala (translation: school of Sansksrit) is a neural Sanskrit Natural Language Processing (NLP) toolset designed to allow computational linguistic analysis for a variety of tasks, including word segmentation, morphological tagging, dependency parsing, and compound type recognition. Our systems perform at the cutting edge on all accessible benchmark datasets for all tasks. SanskritShala is a web-based toolkit that provides the user with real-time analysis for the input provided. It is designed with user-friendly interactive data annotation tools that allow annotators to amend the system's incorrect predictions. We share the source codes of the four modules contained in the toolkit, six word embedding models trained on publicly accessible Sanskrit corpora, and various annotated datasets for evaluating the intrinsic features of word embeddings. According to our knowledge, this is the first neural-based Sanskrit NLP toolbox with a web interface and many NLP modules. We are certain that practitioners of Sanskrit computational linguistics (SCL) will find it beneficial for instructional and annotation purposes. Our platform's video demonstration is coming soon.

Word Segmentation

Please click here to get started with this task. This application is integrated with our pretrained state-of-the-art model. You may use it for creating annotated dataset.

Morphological Tagging

Please click here to get started with this task. This application is integrated with our pretrained state-of-the-art model. You may use it for creating annotated dataset.

Dependency Parsing

Please click here to get started with this task. This application is integrated with our pretrained state-of-the-art model. You may use it for creating annotated dataset.

Compound Classifier

Please click here to get started with this task. This application is integrated with our pretrained state-of-the-art model. You may use it for creating annotated dataset.

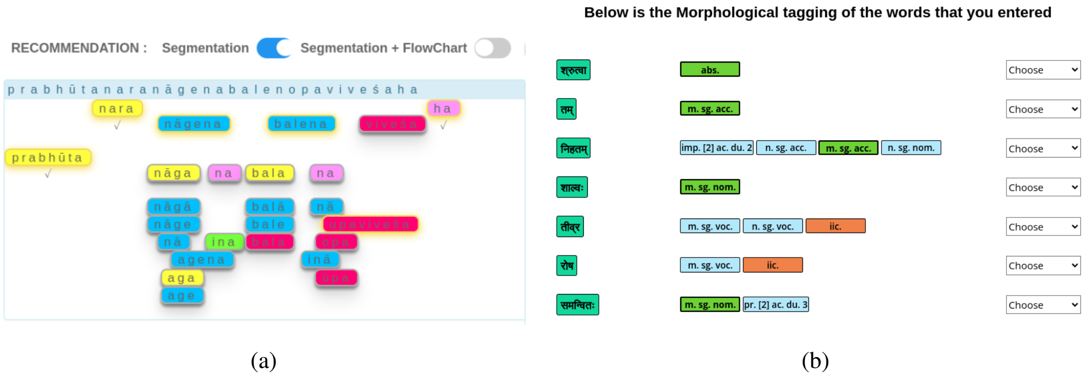
SanskritShala Bot

Hi! I'm a Bot. Let me know if you have any questions regarding our tool!

Can you please let me know how can I help you?

Give me a demo
Motivation behind SanskritShala
Applications of SanskritShala
More about modules

Figure 2: The web interface of the SanskritShala. At the bottom right, a rule-based chatbot is added to navigate users on the platform to give users a user-friendly experience.


RECOMMENDATION : Segmentation Segmentation + FlowChart
p r a b h ū t a n a r a n ā g e n a b a l e n o p a v i v e ś a h a
nara
✓
nāgena
balena
viveśa
ha
✓
prabhūta
✓
nāga
na
bala
na
nāgā
balā
nā
nāge
bale
upaviveśa
nā
ina
bala
opa
agena
inā
aga
ūpa
age

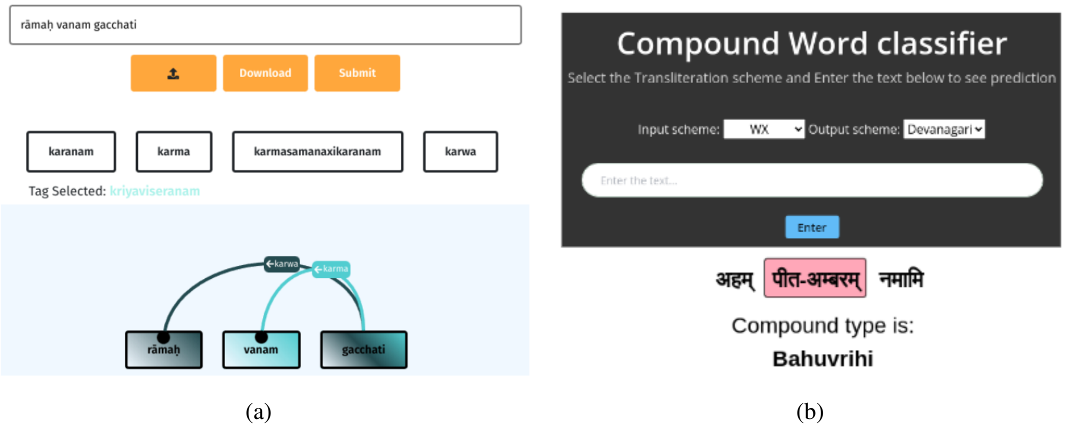
(a)

Below is the Morphological tagging of the words that you entered

श्रुत्वा	abs.	Choose
तम्	m. sg. acc.	Choose
निहतम्	imp. [2] ac. du. 2 · n. sg. acc. · m. sg. acc. · n. sg. nom.	Choose
शाल्वः	m. sg. nom.	Choose
तीव्र	m. sg. voc. · n. sg. voc. · iic.	Choose
रोष	m. sg. voc. · iic.	Choose
समन्वितः	m. sg. nom. · pr. [2] ac. du. 3	Choose

(b)

Figure 3: (a) The candidate solution space generated by SHR for the word segmentation task and the predicted solution by our pretrained model is recommended for the sequence ‘*prabhūtanaranāgena balenopaviveśa ha*’ using a yellow highlight. (b) Morphological Tagger: For each word, we show possible morphological analyses suggested by SHR as well as our system prediction in green if it falls in SHR’s candidate space, otherwise in orange.


rāmaḥ vanam gacchati

Tag Selected: kriyaviseranam

(a)

Compound Word classifier

Select the Transliteration scheme and Enter the text below to see prediction

Input scheme: WX Output scheme: Devanagari

अहम् पीत-अम्बरम् नमामि

Compound type is:

Bahuvrihi

(b)

Figure 4: (a) Dependency parser: Interactive module for the dependency parsing task which directly loads predicted dependency trees from our pretrain model and allows user to correct mispredictions using our interactive interface. (b) Illustration of compound identifier

Class	Input type	Systems
Static	character	charLM
	subword	fastText
	token	word2vec, gloVe, LCM
Contextualized	character	ELMo
	subword	ALBERT

Table 1: Overview of Sanskrit pretrained embeddings.

et al., 2017; Heinzerling and Strube, 2018) and character-level (Kim et al., 2016; Jozefowicz et al., 2016) modeling have been proposed. We also explore two contextualized embeddings: ELMo (Peters et al., 2018) and ALBERT (Lan et al., 2020), a lighter version of BERT. We trained these 6 embedding methods on Sanskrit corpora and made the pretrained models publicly available (Sandhan et al., 2023).[5] The following section describes our proposed pretraining for low-resource settings.

LCM Pretraining: We propose a supervised pretraining, which automatically leverages morphological information using the pretrained encoders. In a nutshell, LCM integrates word representations from multiple encoders trained on three independent auxiliary tasks into the encoder of the neural dependency parser. LCM is acronym for three auxiliary tasks: Lemma prediction, Case prediction and Morphological tag prediction. LCM follows a pipeline-based approach consisting of two steps: pretraining and integration. Pretraining uses a sequence labelling paradigm and trains encoders for three independent auxiliary tasks. Later, these pretrained encoders are combined with the encoder of the neural parser via a gating mechanism similar to Sato et al. (2017). The LCM consists of three sequence labelling-based auxiliary tasks, namely, predicting the dependency label between a modifier-modified pair (**LT**), the monolithic morphological label **(MT)**, and the case attribute of each nominal **(CT)**. We encourage readers to refer Sandhan et al. (2021, LCM) for more details.

Datasets: The quality of word embedding spaces is evaluated through intrinsic and extrinsic methods. This study focuses on intrinsic evaluation, which involves assessing semantic and syntactic information in the words without testing on NLP applications. It is based on works such as Mikolov et al. (2013) and Baroni et al. (2014). These evaluations require a query inventory containing a query word and a related target word. However, such query inventories are not readily available for Sanskrit. To address this, we annotated query inventories for 4 intrinsic tasks: analogy prediction, synonym detection, relatedness, and concept categorization. The inventories were constructed using resources such as Sanskrit WordNet (Kulkarni, 2017), Amarakoṣa (Nair and Kulkarni, 2010), and Sanskrit Heritage Reader (Goyal and Huet, 2016b; Huet and Goyal, 2013).

Web Interface: Figure 2 shows our SanskritShala toolkit that offers interactive web-based predictions for various NLP tasks. The toolkit is built using React framework, which makes it user-friendly and easy to use. One of the tasks it handles is the word segmentation task, which is built on top of the web-based application called SHR++. The SHR++ demonstration is depicted in Figure 3(a). The user inputs a Sanskrit string, which is then sent in real-time to SHR for potential word splits. The system prediction is then obtained from the pretrained word tokenizer. The human annotator is presented with the candidate solution space, with the system prediction highlighted in yellow. The toolkit also features a flask-based application for morphological tagging, which takes user input and scrapes possible morphological tags for each word using SHR. As shown in Figure 3(b), the predictions of the pretrained morphological tagger are displayed in green or orange, depending on whether they are present in the candidate solution of SHR or not. The user can also add a new tag if the actual tag is missing in the SHR solution space or the system's prediction. For the dependency parsing module, we have built a react-based front-end. The user input is passed to the pretrained model to generate a dependency structure. As illustrated in Figure 4(a), the front-end automatically loads the predicted dependency tree and allows the user to make corrections if there are any mispredictions. Additionally, Figure 4(b) shows a flask-based application for the compound type identifier, where users can give input to the system through its web interface. The final annotations can be downloaded after each individual module. We plan to maintain the progress of Sanskrit NLP and offer an overview of available datasets and existing state-of-the-art via the leaderboard for various tasks.

Interactive Chatbot: SanskritShala-bot is a rule-based chatbot that makes it easy to automate simple

[5] https://github.com/Jivnesh/SanskritShala/tree/master/EvalSan

and repetitive user requests, like answering frequently asked questions and directing users to relevant resources. It is also easier to set up and maintain than AI-powered chatbots, which are more complicated. SanskritShala-bot is useful for addressing frequently asked standard queries. It helps familiarize users with the platform by providing them with information and guidance on how to use it. It can answer questions about the platform's features, help users find their way around it, and explain step-by-step how to do certain tasks. This can make it easier for users to get started and leading to a better user experience.

6 Conclusion

We present the first neural-based Sanskrit NLP toolkit, SanskritShala which facilitates diverse linguistic analysis for tasks such as word segmentation, morphological tagging, dependency parsing and compound type identification. It is set up as a web-based application to make the toolkit easier to use for teaching and annotating. All the codebase, datasets and web-based applications are publicly available. We also release word embedding models trained on publicly available Sanskrit corpora and various annotated datasets for 4 intrinsic evaluation tasks to assess the intrinsic properties of word embeddings. We strongly believe that our toolkit will benefit people who are willing to work with Sanskrit and will eventually accelerate the Sanskrit NLP research.

Limitations

We plan to extend SanskritShala by integrating more downstream tasks such as Post-OCR correction, named entity recognition, verse recommendation, word order linearisation, and machine translation. Improving the performance of existing tasks would be important. For example, the current dependency parser is very fragile (performance drops by 50%) in the poetry domain.

Ethics Statement

Our work involves the development of a platform for annotating Sanskrit text. We believe that this platform will be useful for people who are willing to work with Sanskrit for research and educational purposes. We have ensured that our platform is designed ethically and responsibly. We do not foresee any harmful effects of our platform on any community. However, we caution users to use the platform carefully as our pretrained models are not perfect, and errors can occur in the annotation process. All our systems are built using publicly available benchmark datasets, and we have released all our pretrained models and source codes publicly for future research. We are committed to transparency and open access in our work, and we believe that sharing our resources will benefit the wider NLP community. We also acknowledge that NLP research can have potential ethical implications, particularly in areas such as data privacy, bias and discrimination. We are committed to continuing to consider these ethical implications as we develop our platform, and we welcome feedback from the community on how we can improve our ethical practices.

Acknowledgements

We are thankful to Oliver Hellwig for the DCS dataset and Gerard Huet for the Sanskrit Heritage Engine. We are grateful to Hackathon organizers[6] who encouraged us to build the best performing morphological tagger. We thank Amrith Krishna, Uniphore for providing the SHR++ interface as a starting point for the web interface of SanskritShala. We appreciate the assistance of Bishal Santra and Suman Chakraborty, IIT Kharagpur in deploying SanskritShala on the IIT Kharagpur server. We are grateful to Hritik Sharma, IIT Kanpur for helping us build a React-based front-end for SanskritShala. We are thankful to Hrishikesh Terdalkar, IIT Kanpur for helpful discussions on deploying systems. We appreciate the anonymous reviewers' insightful suggestions for enhancing this work. We'd like to say thanks to everyone who helped us make the different neural models for SanskritShala. The work was supported in part by the National Language Translation Mission (NLTM): Bhashini project by Government of India. The work of the first author is supported by the TCS Fellowship under the Project TCS/EE/2011191P.

References

Devaraja Adiga, Rishabh Kumar, Amrith Krishna, Preethi Jyothi, Ganesh Ramakrishnan, and Pawan Goyal. 2021. Automatic speech recognition in Sanskrit: A new speech corpus and modelling insights. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5039–5050, Online. Association for Computational Linguistics.

[6] https://sanskritpanini.github.io/index.html

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, Baltimore, Maryland. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Pawan Goyal, Gérard Huet, Amba Kulkarni, Peter Scharf, and Ralph Bunker. 2012. A distributed platform for Sanskrit processing. In *Proceedings of COLING 2012*, pages 1011–1028, Mumbai, India. The COLING 2012 Organizing Committee.

Pawan Goyal and Gérard Huet. 2016a. Design and analysis of a lean interface for sanskrit corpus annotation. *Journal of Language Modelling*, 4:145.

Pawan Goyal and Gérard Huet. 2016b. Design and analysis of a lean interface for sanskrit corpus annotation. *Journal of Language Modelling*, 4:145.

Ashim Gupta, Amrith Krishna, Pawan Goyal, and Oliver Hellwig. 2020. Evaluating neural morphological taggers for Sanskrit. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 198–203, Online. Association for Computational Linguistics.

Benjamin Heinzerling and Michael Strube. 2018. BPEmb: Tokenization-free pre-trained subword embeddings in 275 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Oliver Hellwig and Sebastian Nehrdich. 2018. Sanskrit word segmentation using character-level recurrent and convolutional neural networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2754–2763, Brussels, Belgium. Association for Computational Linguistics.

Gérard Huet and Pawan Goyal. 2013. Design of a lean interface for sanskrit corpus annotation.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2016. Character-aware neural language models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 2741–2749. AAAI Press.

Daniel Kondratyuk, Tomáš Gavenčiak, Milan Straka, and Jan Hajič. 2018. LemmaTag: Jointly tagging and lemmatizing for morphologically rich languages with BRNNs. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4921–4928, Brussels, Belgium. Association for Computational Linguistics.

Amrith Krishna, Ashim Gupta, Deepak Garasangi, Jeevnesh Sandhan, Pavankumar Satuluri, and Pawan Goyal. 2023. Neural approaches for data driven dependency parsing in Sanskrit. In *Proceedings of the Computational Sanskrit & Digital Humanities: Selected papers presented at the 18th World Sanskrit Conference*, pages 1–20, Canberra, Australia (Online mode). Association for Computational Linguistics.

Amrith Krishna, Ashim Gupta, Deepak Garasangi, Pavankumar Satuluri, and Pawan Goyal. 2020a. Keep it surprisingly simple: A simple first order graph based parsing model for joint morphosyntactic parsing in Sanskrit. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4791–4797, Online. Association for Computational Linguistics.

Amrith Krishna, Bishal Santra, Sasi Prasanth Bandaru, Gaurav Sahu, Vishnu Dutt Sharma, Pavankumar Satuluri, and Pawan Goyal. 2018. Free as in free word order: An energy based model for word segmentation and morphological tagging in Sanskrit. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2550–2561, Brussels, Belgium. Association for Computational Linguistics.

Amrith Krishna, Bishal Santra, Ashim Gupta, Pavankumar Satuluri, and Pawan Goyal. 2020b. A graph-based framework for structured prediction tasks in Sanskrit. *Computational Linguistics*, 46(4):785–845.

Amrith Krishna, Bishal Santra, Ashim Gupta, Pavankumar Satuluri, and Pawan Goyal. 2021. A Graph-Based Framework for Structured Prediction Tasks in Sanskrit. *Computational Linguistics*, 46(4):785–845.

Amrith Krishna, Pavankumar Satuluri, Shubham Sharma, Apurv Kumar, and Pawan Goyal. 2016. Compound type identification in Sanskrit: What roles do the corpus and grammar play? In *Proceedings of the 6th Workshop on South and Southeast Asian Natural Language Processing (WSSANLP2016)*, pages 1–10, Osaka, Japan. The COLING 2016 Organizing Committee.

Amrith Krishna, Vishnu Sharma, Bishal Santra, Aishik Chakraborty, Pavankumar Satuluri, and Pawan Goyal. 2019. Poetry to prose conversion in Sanskrit as a linearisation task: A case for low-resource languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1160–1166, Florence, Italy. Association for Computational Linguistics.

Amrith Krishna, Shiv Vidhyut, Dilpreet Chawla, Sruti Sambhavi, and Pawan Goyal. 2020c. SHR++: An interface for morpho-syntactic annotation of Sanskrit corpora. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 7069–7076, Marseille, France. European Language Resources Association.

Sriram Krishnan, Amba Kulkarni, and Gérard Huet. 2020. Validation and normalization of dcs corpus using sanskrit heritage tools to build a tagged gold corpus.

Amba Kulkarni, Pavankumar Satuluri, Sanjeev Panchal, Malay Maity, and Amruta Malvade. 2020. Dependency relations for Sanskrit parsing and treebank. In *Proceedings of the 19th International Workshop on Treebanks and Linguistic Theories*, pages 135–150, Düsseldorf, Germany. Association for Computational Linguistics.

Amba Kulkarni and Dipti Sharma. 2019. Pāṇinian syntactico-semantic relation labels. In *Proceedings of the Fifth International Conference on Dependency Linguistics (Depling, SyntaxFest 2019)*, pages 198–208, Paris, France. Association for Computational Linguistics.

Malhar Kulkarni. 2017. *Sanskrit WordNet at Indian Institute of Technology (IITB) Mumbai*, pages 231–241. Springer Singapore, Singapore.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26:3111–3119.

Sivaja Nair and Amba Kulkarni. 2010. The knowledge structure in amarakosa. pages 173–189.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Jivnesh Sandhan, Laxmidhar Behera, and Pawan Goyal. 2022a. Systematic investigation of strategies tailored for low-resource settings for sanskrit dependency parsing.

Jivnesh Sandhan, Ayush Daksh, Om Adideva Paranjay, Laxmidhar Behera, and Pawan Goyal. 2022b. Prabhupadavani: A code-mixed speech translation data for 25 languages. In *Proceedings of the 6th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 24–29, Gyeongju, Republic of Korea. International Conference on Computational Linguistics.

Jivnesh Sandhan, Ashish Gupta, Hrishikesh Terdalkar, Tushar Sandhan, Suvendu Samanta, Laxmidhar Behera, and Pawan Goyal. 2022c. A novel multi-task learning approach for context-sensitive compound type identification in Sanskrit. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4071–4083, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Jivnesh Sandhan, Amrith Krishna, Pawan Goyal, and Laxmidhar Behera. 2019. Revisiting the role of feature engineering for compound type identification in Sanskrit. In *Proceedings of the 6th International Sanskrit Computational Linguistics Symposium*, pages 28–44, IIT Kharagpur, India. Association for Computational Linguistics.

Jivnesh Sandhan, Amrith Krishna, Ashim Gupta, Laxmidhar Behera, and Pawan Goyal. 2021. A little pretraining goes a long way: A case study on dependency parsing task for low-resource morphologically rich languages. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 111–120, Online. Association for Computational Linguistics.

Jivnesh Sandhan, Om Adideva Paranjay, Komal Digumarthi, Laxmidhar Behra, and Pawan Goyal. 2023. Evaluating neural word embeddings for Sanskrit. In *Proceedings of the Computational Sanskrit & Digital Humanities: Selected papers presented at the 18th World Sanskrit Conference*, pages 21–37, Canberra, Australia (Online mode). Association for Computational Linguistics.

Jivnesh Sandhan, Rathin Singha, Narein Rao, Suvendu Samanta, Laxmidhar Behera, and Pawan Goyal. 2022d. Translist: A transformer-based linguistically informed sanskrit tokenizer.

Motoki Sato, Hitoshi Manabe, Hiroshi Noji, and Yuji Matsumoto. 2017. Adversarial training for cross-domain Universal Dependency parsing. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 71–79, Vancouver, Canada. Association for Computational Linguistics.

Krishnan Sriram, Amba Kulkarni, and Gérard Huet. 2023. Validation and normalization of DCS corpus and development of the Sanskrit heritage engine's segmenter. In *Proceedings of the Computational*

Sanskrit & Digital Humanities: Selected papers presented at the 18th World Sanskrit Conference, pages 38–58, Canberra, Australia (Online mode). Association for Computational Linguistics.

Hrishikesh Terdalkar and Arnab Bhattacharya. 2021. Sangrahaka: A tool for annotating and querying knowledge graphs. In *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE 2021, page 1520–1524, New York, NY, USA. Association for Computing Machinery.

Hrishikesh Terdalkar and Arnab Bhattacharya. 2022. Chandojnanam: A sanskrit meter identification and utilization system.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Charagram: Embedding words and sentences via character n-grams. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1504–1515, Austin, Texas. Association for Computational Linguistics.