

A Novel Multi-Task Learning Approach for Context-Sensitive Compound Type Identification in Sanskrit

Jivnesh Sandhan¹, Ashish Gupta², Hrishikesh Terdalkar¹, Tushar Sandhan¹,
Suvendu Samanta¹, Laxmidhar Behera^{1,3} and Pawan Goyal²

¹IIT Kanpur, ²IIT Kharagpur, ³IIT Mandi

jivnesh@iitk.ac.in, ashishgupta2598@gmail.com,

pawang@cse.iitkgp.ac.in

Abstract

The phenomenon of compounding is ubiquitous in Sanskrit. It serves for achieving brevity in expressing thoughts, while simultaneously enriching the lexical and structural formation of the language. In this work, we focus on the *Sanskrit Compound Type Identification (SaCTI)* task, where we consider the problem of identifying semantic relations between the components of a compound word. Earlier approaches solely rely on the lexical information obtained from the components and ignore the most crucial contextual and syntactic information useful for SaCTI. However, the SaCTI task is challenging primarily due to the implicitly encoded context-sensitive semantic relation between the compound components.

Thus, we propose a novel multi-task learning architecture which incorporates the contextual information and enriches the complementary syntactic information using morphological tagging and dependency parsing as two auxiliary tasks. Experiments on the benchmark datasets for SaCTI show 6.1 points (Accuracy) and 7.7 points (F1-score) absolute gain compared to the state-of-the-art system. Further, our multilingual experiments demonstrate the efficacy of the proposed architecture in English and Marathi languages.¹

1 Introduction

A compound is defined as a collection of one or more entities that act as a single meaningful entity. The process of decoding an implicit semantic relation between the components of a compound in Sanskrit is called *Sanskrit Compound Type Identification (SaCTI)*. Alternatively, it is also termed as *Noun Compound Interpretation (NCI)* (Ponkiya et al., 2021, 2020). In the literature, the NCI problem has been formulated in two related but different ways. Let's take *mango juice* as an example. In the

¹The code and datasets are publicly available at: <https://github.com/ashishgupta2598/SaCTI>

first formulation, the relation between the two components is labeled from a set of semantic relations (*MADEOF*) (Dima and Hinrichs, 2015; Fares et al., 2018; Ponkiya et al., 2021). The second formulation uses paraphrasing to illustrate the semantic relations (*a juice made from mango*) (Lapata and Keller, 2004; Ponkiya et al., 2018a, 2020).

In this work, we use the first formulation that frames the task as a multi-class classification problem. The task is challenging and often depends upon the context or world knowledge about the entities involved (Krishna et al., 2016). For instance, the semantic type of the compound *rāma-īśvaraḥ* can be classified into one of the following semantic types depending on the context: *Karmadhāraya*², *Bahuvrīhi* and *Tatpuruṣa*. Although the compound has the same components as well as the final form, the implicit relationship between the components can be decoded only with the help of available contextual information (Kulkarni and Kumar, 2013; Krishna et al., 2016). Due to such instances, the downstream Natural Language Processing (NLP) applications for Sanskrit such as question answering (Terdalkar and Bhattacharya, 2019) and machine translation (Aralikatte et al., 2021), etc. show sub-optimal performance when they stumble on compounds. For example, while translating *rāma-īśvaraḥ* into English, depending on the semantic type, there are three possible meanings: (1) Lord who is pleasing (in *Karmadhāraya*) (2) the one whose God is Rama (in *Bahuvrīhi*) (3) Lord of Rama (in *Tatpuruṣa*). Therefore, the SaCTI task can be seen as a preliminary pre-requisite to building a robust NLP technology for Sanskrit. Further, this dependency on contextual information rules out the possibility of storing and doing a lookup to identify a compound's semantic types.

²There are 4 broad semantic types of compounds: *Avyayībhāva*, *Bahuvrīhi*, *Dvandva*, and *Tatpuruṣa*. *Karmadhāraya* is considered as sub-type of *Tatpuruṣa*. We encourage readers to refer Krishna et al. (2016) for more details on these semantic types.

With the advent of recent contextual models (Peters et al., 2018; Devlin et al., 2019; Conneau et al., 2020), there has been upsurge in performance of various downstream NLP applications (Kondratyuk and Straka, 2019; Liu et al., 2019; Yang et al., 2019). Nevertheless, there have been no efforts to build context-dependent models in SaCTI.³ This may be attributed to the fact that while most of the natural language technology is built for resource-rich languages such as English (Joshi et al., 2020), compounding is not a predominant phenomenon in them (Krishna et al., 2016). There is also lack of task-specific context-sensitive labeled data.

Earlier approaches (Kulkarni and Kumar, 2013; Krishna et al., 2016; Sandhan et al., 2019) for SaCTI solely rely on lexical information obtained from the components and are blind to potentially useful contextual and syntactic information. The context is the most feasible, cheaply available information. As per Paṇini’s grammar (Paṇini, 500 BCE; Kulkarni and Kumar, 2013), the morphological features have direct correlation with the semantic types. Sometimes, the dependency information also helps in disambiguation and can provide a medium to enrich contextual information. Thus, we propose a novel multi-task learning approach which (1) incorporates the contextual information, and (2) enriches the complementary syntactic information using morphological tagging and dependency parsing auxiliary tasks without any additional manual labeling. Summarily, our key contributions are:

- We propose a novel context-sensitive multi-task learning architecture for SaCTI (§ 2).
- We illustrate that morphological tagging and dependency parsing auxiliary tasks are helpful and serve as a proxy for explainability of system predictions (§ 4) for the SaCTI task.
- We report results with 7.71 points (F1) absolute gains compared to the current state-of-the-art system by Krishna et al. (2016) (§ 3.2).
- We show the efficacy of the proposed approach in English and Marathi (§ 4).
- We release our codebase and pre-processed datasets (including newly annotated Marathi dataset (§ 3.1)) and web-based tool (§ 4) for using our pretrained models.

³Refer to related work section (§ 5) for more details.

2 The proposed system

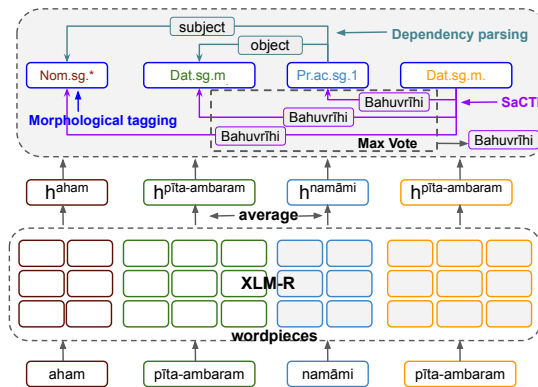


Figure 1: Illustration of the proposed multi-task learning architecture with an example “*aham pīta-ambaram namāmi*” (Translation: “I pray to Pītāmbara (Lord Vishnu).”) where ‘*pīta-ambaram*’ is a compound word belonging to the *Bahuvrīhi* semantic type as per the given context. We feed this context and the compound word at the end as an input to the system. Each token is split into wordpieces using a multi-lingual tokenizer (Kudo and Richardson, 2018). This sequence of wordpieces is passed to multi-lingual pretrained XLM-R encoder (Conneau et al., 2020). The hidden representation of each token is the average of its wordpieces’ representations obtained from the encoder. We apply our multi-task learning architecture which consists of three tasks, namely, Sanskrit compound type identification (SaCTI), morphological tagging and dependency parsing over the hidden representations. We formulate SaCTI as a pair-wise (a context word and the compound) classification task where the objective is to predict the semantic type of the target compound word (*Bahuvrīhi*). At test time, we apply the maximum voting policy to select a single prediction from the set of semantic relations predicted by such n pairs.

Figure 1 illustrates the proposed multi-task learning architecture with an example context, “*aham pīta-ambaram namāmi*” (Translation: “I pray to Pītāmbara (Lord Vishnu).”) where ‘*pīta-ambaram*’ is a compound belonging to the *Bahuvrīhi* semantic type as per the given context. As shown in Figure 1, we feed this context along with the compound word concatenated at the end, as an input to the system, and obtain hidden representations from the multi-lingual encoder as described below. On top of the hidden representations as obtained via the encoder module, we apply our multi-task learning architecture consisting of three tasks: SaCTI, morphological tagging, and dependency parsing. We formulate the SaCTI task as a pair-wise (a context word paired with the compound word) classification task

where the objective is to predict the semantic type of the target compound word (*Bahuvrīhi* for *pīta-ambaram* compound word in this example) for all the pairs as shown in Figure 1. At test time, we apply the maximum voting policy to select a single prediction from the set of semantic types predicted by these pairs. We formally discuss the details of each component below.

Multilingual Encoder: Sanskrit is a morphologically rich and low-resource language. In order to build powerful contextual representations for Sanskrit words, morphological richness poses the out-of-vocabulary problem and low-resource nature poses an unlabelled data scarcity problem. Thus, we opt for a multi-lingual encoder (Conneau et al., 2020, XLM-R) to mitigate these issues.

Given a compound c_p and its context $C = [c_1, c_2, \dots, c_n]$ such that p^{th} position ($1 \leq p \leq n$) in the context is the compound word, we append the compound word to the context such that $C = [c_1, c_2, \dots, c_n, c_{n+1}]$ where $c_{n+1} = c_p$. Each token (c_i) is further split into wordpieces ($c_i = [c_i^1, c_i^2, \dots, c_i^{m_i}]$) using a multi-lingual subword tokenizer (Kudo and Richardson, 2018; Kudo, 2018, Sentencepiece). Next, we pass the overall sequence ($C = [c_1^1, c_1^2, \dots, c_n^{m_n}, c_{n+1}^1, \dots, c_{n+1}^{m_{n+1}}]$) of wordpieces corresponding to the context C into the pretrained transformer. Finally, we obtain the contextual representation of all tokens as $h = (h_1, h_2, h_3, \dots, h_n, h_{n+1})$ where

$$h_i = \frac{1}{m_i} \sum_{k=1}^{m_i} Transformer(c_i^k) \quad (1)$$

SaCTI: Our context-sensitive classifier uses Bi-Affine attention, henceforth referred to as BiAFF. Given the hidden representations of i^{th} , j^{th} context words as h_i, h_j from the multi-lingual encoder, the scoring function $s_{i,j}$ indicates the system’s belief that the latter (j^{th}) (Eqn. 2) should be related to the former (i^{th}) in identifying the semantic type of the latter, where $q_i^T z_i$ indicates bias to capture the prior of contextual information in the i^{th} word.

$$s_{i,j} = z_i^T U z_j + q_i^T z_i \quad (2)$$

where $z_i = MLP(h_i)$, U and q_i are learnable parameters, MLP denotes a multi-layered perceptron. Similarly, a score for k^{th} possible semantic type relation between every pair of i^{th} context word ($\forall i \in [1, n]$) and the compound is computed by:

$$r_{i,k} = z_i^T U'_k z'_{n+1} + q'_k{}^T [z'_i; z'_{n+1}] + b'_k \quad (3)$$

where $z'_i = MLP'(h_i)$, U' , b'_k and q'_k are learnable parameters. Finally, model maximizes the following objective function during training.

$$\sum_{i=1}^n p(y_{n+1}|c_i, \theta) + p(y_{n+1}^l|c_i, y_{n+1}, \theta) \quad (4)$$

where y_{n+1} is the target compound appended in context C , y_{n+1}^l is the semantic type of $(y_{n+1}/c_i, \theta)$, θ denotes system’s parameters, $p(y_{n+1}|c_i, \theta) \propto \exp(s_{i,n+1})$, $p(y_{n+1}^l|c_i, y_{n+1}, \theta) \propto \exp(r_{i,l})$. At test time, we apply maximum voting policy to select a single prediction from the set of semantic relations predicted for n pairs $(c_i, c_{n+1}), \forall i \in [1, n]$.

Morphological tagging: The primary motivation behind using morphological tagging as an auxiliary task aligns well with grammatical rules from Paṇini’s grammar (Paṇini, 500 BCE; Kulkarni and Kumar, 2013). For instance, *Avyayībhāva* compounds are in neuter gender. *Tatpuruṣa* is a function of ‘case’ attribute of morphological features. The number attribute of a compound depends on the semantic type of the compound. Also, there are constraints based on inflection/derivational suffix. Summarily, these morphological features have direct correlation with the semantic classes. In our proposed system, the morphological tagging task leverages hidden representations from the multi-lingual encoder and decodes the pseudo-labels⁴ using a fully connected layer followed by a softmax layer.⁵ In this process, morphological information useful for the SaCTI task is enriched in the hidden representations. This can be seen as an implicit way to encode the grammatical rules in the system.

Dependency parsing: Lowe (2015) argues that compounding is mostly a syntactic phenomenon. *Bahuvrīhi* compounds are “exocentric” in nature, which attribute a property to an entity external to the compound with the adjective relationship. Sometimes, syntactic information can provide a complementary signal useful for compound type disambiguation. For instance, consider the following example: *aham nīla-utpalah saraḥ paśyāmi* (Translation: I watch the pond having a blue-lotus.) Here, *nīla-utpalam* qualifies to be *Bahuvrīhi* due to

⁴The benchmark datasets do not have a gold standard morphological information. We use predicted morphological information as pseudo-labels (§ 3.1).

⁵Note that all the parameters present in the multi-lingual encoder are trainable during the task-specific training.

presence of its referent *sarah* with an adjective relationship. However, in the absence of *sarah* in the context, ambiguity pops up in between *Bahuvrīhi* and *Tatpuruṣa*.⁶ This motivates us to investigate the usefulness of syntactic information for the SaCTI task. The benchmark datasets do not have a gold standard dependency information. We use predicted dependency trees as pseudo-labels (§ 3.1). Our dependency parsing component leverages Bi-Affine parser (Dozat and Manning, 2017) over hidden representations from multi-lingual encoder.

3 Experiments

3.1 Dataset and metrics

Table 1 reports the unique number of compounds, data statistics and the number of semantic types for the respective datasets used in this work. We restrict to binary compounds (compounds with two components) in all the datasets. These datasets consist of components, context and semantic type of a compound. The SaCTI datasets for Sanskrit are available with two levels of annotations: coarse (4 broad types) and fine-grained (15 sub-types).

Datasets	#Unique	#Train	#Dev	#Test	#Types
SaCTI-base	8,594	9,356	2,339	2,994	4 (15)
SaCTI-large	48,132	59,133	6,571	7,301	4 (15)
English	4,676	4,163	1,041	1,301	7
Marathi	368	659	99	114	4

Table 1: Data statistics for all the datasets

Sanskrit: We evaluate on two available context-sensitive benchmark datasets: *SaCTI-base* and *SaCTI-large*. We follow the same experimental settings as Krishna et al. (2016) in *SaCTI-base* to keep our results comparable with their state-of-the-art results. *SaCTI-base* is a subset of *SaCTI-large* dataset. In due course of time, more annotated data resulted in *SaCTI-large* dataset.

English: We use instance-based (context-dependent) noun-noun compound dataset released by Fares (2016). The compounds used in this dataset are extracted from the Wall Street Journal (WSJ) portion in the Penn Treebank (PTB).

Marathi: We create an annotated context-sensitive compound data for Marathi due to its unavailability. We extract compound words from

⁶The ambiguity is whether I am seeing the blue lotus or the pond having a blue lotus.

Marathi grammar textbooks. For Marathi, we restrict to the same 4 semantic types as in Sanskrit (coarse setting). In our dataset, the context corresponding to 75% data points is automatically leveraged from Wikipedia. In order to increase the difficulty level of the task, we ask one of the annotators to create the remaining 25% data points in such a way that the same compound with a different context leads to a different semantic type. Next, we provide these compound words and context information to 3 annotators (*A, B, C*)⁷ using a web-based platform. Refer to Appendix B for annotation interface. All annotators have their minimum academic qualification as Master in Arts in Marathi. These annotators have to choose the correct label from the multiple-choice options.⁸ Finally, we use the maximum voting policy amongst annotators to get the label for each data point. Initially, we start with 1,000 data points for annotation. Out of 1,000, we drop 128 data points where none of the 2 annotators have an agreement. The pair-wise inter-annotator agreement between annotators in terms of Cohen Kappa (κ) is as follows: $A - B : 0.40, B - C : 0.20$ and $A - C : 0.35$. The $\kappa \in [0.2, 0.4]$ is considered as fair agreement (McHugh, 2012).

Pseudo-labels for auxiliary tasks: The benchmark datasets do not have a gold standard dependency and morphological information. We use predicted labels as pseudo-labels. For Sanskrit, we obtain pseudo-labels from the Trankit model (Nguyen et al., 2021) trained on STBC dataset (Krishna et al., 2020) and morphological pseudo-labels from the LemmaTag model (Kondratyuk et al., 2018) trained on Hackathon dataset (Krishnan et al., 2020). For English, we use pseudo-labels from English XLM-R model⁹ for morphological and dependency parsing task. For Marathi, we do not find any such data or pretrained model to obtain pseudo-labels. Therefore, we do not activate morphological tagging and dependency parsing components in the proposed system while training on Marathi data.

Hyper-parameter settings: For the implementation of the proposed system, we modify on top of codebase by Nguyen et al. (2021). We use the following hyper-parameter settings for the best config-

⁷Note that the annotator who created the context for 25% data points is different from these 3 annotators.

⁸In case of confusion, we gave additional option to mark as “Not sure”.

⁹<https://spacy.io/models/en>

Datasets	System	Sanskrit (coarse)								Sanskrit (fine-grained)							
		w/o context				w/ context				w/o context				w/ context			
		A	P	R	F1	A	P	R	F1	A	P	R	F1	A	P	R	F1
SaCTI-base	ISCLS19	77.68	76.00	71.00	73.00	77.68	76.00	71.00	73.00	70.64	67.53	63.18	64.58	70.64	67.53	63.18	64.58
	COLING16	77.39	78.00	72.00	74.00	77.39	78.00	72.00	74.00	-	-	-	-	-	-	-	-
	SanALBERT	72.01	72.10	70.00	71.40	77.40	77.31	73.00	74.20	69.39	62.58	58.13	58.22	75.40	70.14	61.22	62.04
	IndicALBERT	71.87	51.60	53.90	52.00	78.63	77.47	75.83	76.47	68.06	53.05	53.19	52.30	71.93	57.29	56.49	55.25
	mBERT	76.12	77.43	71.68	73.10	81.42	83.12	73.54	78.09	75.00	67.65	72.69	68.56	78.36	77.41	69.59	69.37
	XLM-R	78.19	73.54	73.10	73.31	81.00	82.01	77.00	79.10	73.75	66.38	63.96	65.16	77.94	71.72	70.18	70.94
	Ours	80.21	72.31	74.50	73.38	83.45	79.65	83.87	81.71	78.25	72.94	73.81	72.68	82.47	76.87	79.08	77.20
SaCTI-large	ISCLS19	90.69	75.66	72.09	73.76	90.69	75.66	72.09	73.76	76.66	71.62	65.40	68.09	76.66	71.62	65.40	68.09
	SanALBERT	88.17	66.65	61.31	62.85	87.84	64.15	65.44	63.63	79.38	69.90	75.34	67.57	80.73	71.62	74.60	72.69
	IndicALBERT	87.77	68.82	49.58	56.05	92.95	84.98	74.90	79.32	79.03	68.00	64.14	63.67	83.13	74.23	79.77	76.11
	mBERT	92.29	78.51	77.45	77.41	93.52	83.13	80.82	81.83	81.56	70.82	76.91	72.74	80.98	70.00	79.13	72.80
	XLM-R	92.61	79.91	79.00	79.42	93.85	86.64	79.67	82.78	81.84	74.46	77.93	75.68	83.12	73.97	81.07	76.20
		Ours	93.54	81.30	81.65	81.47	94.78	83.89	87.61	85.64	82.85	74.94	78.12	76.49	84.73	78.53	80.30

Table 2: Evaluation on Sanskrit datasets in two levels of annotations (coarse and fine-grained) and two settings (w/o context and w/ context). The best results are bold. Results are averaged over 4 runs. The significance test between the best baseline XLM-R and the proposed system in terms of Recall/Accuracy metrics: $p < 0.01$ (as per t-test). We could not perform significance test with COLING16 (SaCTI-base-coarse-w/o) and report its results on all the datasets due to unavailability of its predictions and codebase. ISCLS19 and COLING16 baselines cannot utilize the context information; therefore, we report the same numbers in w/context as w/o context.

uration of the proposed system: number of epochs as 70, batch size 50 and a embedding dropout rate of 0.3 with a learning rate of 0.001. In our multi-task loss function, we penalize dependency component’s loss function by 0.01 to prioritize the performance on SaCTI. This penalty is identified based on hyper-parameter tuning. The rest of the hyper-parameters are kept the same as Nguyen et al. (2021). For multi-lingual baselines, we used Huggingface’s transformers repository (Wolf et al., 2020). We release our codebase and datasets publicly under the licence CC-BY 4.0. All the artifacts used in this work are publicly available for the research purpose.

Computing infrastructure used: We use a single GPU with Tesla P100-PCIE, 16 GB GPU memory, 3584 GPU Cores computing infrastructure for our experiments. Our proposed system takes approximately 1 hour for training SaCTI-base coarse w/o context setting dataset.

Evaluation metrics: Following Krishna et al. (2016); Sandhan et al. (2019), we report macro averaged Precision, Recall and F1-score for all our experiments. We also report micro-averaged Accuracy. We use Scikit-learn software (Pedregosa et al., 2011) to compute these metrics.

Baselines: We consider two context agnostic systems where Sandhan et al. (2019, ISCLS19) formulate SaCTI as a purely neural-based multi-class classification approach using static word embeddings of components of a compound and Krishna

et al. (2016, COLING16) deploy a hybrid system which leverages linguistically involved hand-crafted feature engineering with distributional information from Adaptor Grammar (Johnson et al., 2006). The COLING16 system is the current state-of-the-art system for the SaCTI task. Next, we opt for multi-lingual contextual language models due to the lack of sufficiently large unlabelled data available for Sanskrit. We consider three multi-lingual pretrained language models, namely, Kakwani et al. (2020, IndicALBERT) which is ALBERT model trained on 12 Indic languages excluding Sanskrit, BERT (Devlin et al., 2019, mBERT) trained on 104 languages having largest Wikipedia’s excluding Sanskrit and Conneau et al. (2020, XLM-R) trained on 100 languages including Sanskrit. Finally, we consider a mono-lingual ALBERT model (Sandhan et al., 2022, SanALBERT) trained on Sanskrit corpus (Hellwig, 2010) from scratch. In all the contextual baselines, we pass the classification token [CLS] representation of a sentence pair (compound word and its context separated by [SEP] token) to the classification head. **Ours:** This is our proposed system from § 2.

3.2 Results

Table 2 shows the performance for the best performing configurations of all the baselines on the test set of benchmark datasets for SaCTI. We report results on two levels of annotations (coarse and fine-grained) and two settings (w/o context and w/

context).¹⁰ Except for ISCLS19 and COLING16 systems, all systems utilize available context along with components of a compound.¹¹ XLM-R reports the best performance among all the baselines while using the context information.

Our proposed system outperforms all the competing systems in terms of all the evaluation metrics and reports 6.1 points (A) and 7.7 points (F1) absolute gain with respect to the current state-of-the-art system COLING16 (on SaCTI-base coarse w/context dataset). COLING16 outperforms the proposed system on SaCTI-base coarse w/o context. Notably, our proposed system outperforms the strong baseline XLM-R with large margins in fine-grained setting on SaCTI-base dataset (low-resourced setting). This confirms the usefulness of the proposed system in low-resourced settings with fine-grain labels. The large performance gap between the proposed system with context and COLING16/ISCLS19 baselines illustrates the efficacy of using contextual information and syntactic information. Summarily, we mark new state-of-the-art results with the help of the novel architecture, where the contextual component is integrated with syntax-based auxiliary tasks such as morphological tagging and dependency parsing. We find a similar trend in performance for the SaCTI-large dataset.

4 Analysis

In this section, we dive deep into the proposed system architecture for a detailed analysis as well as generalizability. We use SaCTI-base coarse dataset in the w/ context setting for the analysis.

(1) Ablation analysis: Here, we investigate the contribution of various components towards the overall improvements of the proposed system. Table 3 reports ablations in terms of all the evaluation metrics when a particular component is inactivated from the proposed system. For example, “-DP” denotes the system where the dependency parsing component is removed from the proposed system. We see that elimination of any of the components deteriorates the performance. Table 3 illustrates that ‘context’ component is the most critical towards improvements. Also, the deletion of the ‘BiAFF’ component has the second largest impact

¹⁰For the systems that require context, we feed the compound word only as the context.

¹¹These baselines cannot utilize the context; therefore, we report the same numbers in w/context as w/o context.

on the final performance.¹²

System	A	P	R	F1
Ours	83.45	79.65	83.87	81.71
-context	80.21	72.31	74.50	73.38
-BiAFF	81.00	82.01	77.00	79.10
-morph	82.87	80.00	81.21	80.60
-DP	81.89	79.35	81.62	80.26
-morph -DP	81.50	82.34	77.67	79.93

Table 3: Ablations of the proposed system in terms of all the metrics. Each ablation deletes a single component from the proposed system. For example, “-DP” deletes the dependency parsing task from the proposed system.

(2) How effective is the proposed system in reducing confusion between conflicting classes?

Figure 2 illustrates the confusion matrices in w/ context and w/o context scenarios. We observe a similar trend in both the scenarios. (1) Both systems mis-classify the predictions into the most populated type (*Tatpuruṣa*). This can be attributed to the imbalanced nature of the dataset.¹³ (2) The confusion between *Avyayībhāva* and *Tatpuruṣa* is due to these compounds having their first component as an indeclinable word. Notably, the system with context is able to reduce confusion by 15%. (3) One of the reasons for conflict between *Tatpuruṣa* and *Bahuvrīhi* is due to the specific subcategory of both the classes where the first component is a negation. With the help of enriched information, a system with context can reduce this miss-classification by 7%. Summarily, the system with contextual information always performs superior to one with no context. This substantiates the importance of contextual information in the proposed system.

(3) How well can we generalize the proposed system for other languages?

The primary motivation is to illustrate the efficacy of our language agnostic approach. The semantic type of compounds of the language of interest need not be similar to that of Sanskrit for the model to work. It is purely language agnostic model. To study the generalization ability of the proposed system, we consider 2 additional languages, namely, English (en) and Marathi (ma). We choose English due to its availability of context-sensitive annotated

¹²In the absence of the proposed ‘BiAFF’ component, we use [CLS] token for the sentence-level prediction, where this system is similar to XLM-R + DP + morph.

¹³In this work, we do not consider any strategy to tackle imbalanced classification. We plan to address this in future.

System	English								Marathi							
	w/o context				w/ context				w/o context				w/ context			
	A	P	R	F1	A	P	R	F1	A	P	R	F1	A	P	R	F1
ISCLS19	67.43	71.81	64.38	66.26	67.43	71.81	64.38	66.26	68.62	70.78	52.89	56.85	68.62	70.78	52.89	56.85
IndicALBERT	68.23	57.25	59.70	57.96	70.22	68.01	69.31	68.65	67.65	33.33	45.15	38.08	60.00	45.17	40.62	40.95
mBERT	70.98	70.00	69.87	69.97	72.48	76.08	73.47	74.30	77.45	58.70	61.28	59.48	71.05	51.45	53.34	52.18
BERT	74.19	72.79	71.12	71.62	74.71	75.83	74.48	75.12	78.43	71.21	68.04	69.20	75.43	67.07	66.07	66.43
XLM-R	72.21	71.53	68.88	69.20	74.33	77.12	76.86	76.57	76.47	67.36	62.90	63.33	74.56	65.50	62.07	62.75
Ours	74.69	74.79	75.19	75.19	77.81	79.17	79.19	79.12	78.12	71.98	70.00	70.57	80.43	66.54	77.00	69.12

Table 4: Evaluation on English and Marathi languages. The best results are bold. The significance test between the best baseline XLM-R and our system in terms of Recall/Accuracy metrics: $p < 0.01$ (as per t-test). ISCLS19 do not have power to utilize the context information; therefore, we report the same numbers in w/context as w/o context.

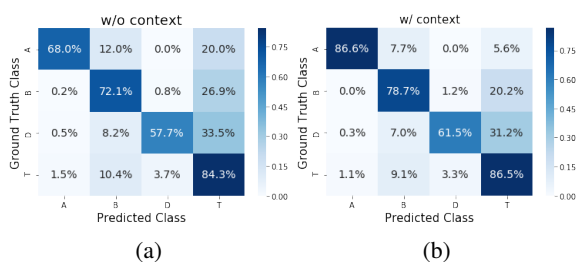


Figure 2: The confusion matrix for the proposed system trained (a) w/o context (b) w/ context. Semantic types: *Avyayībhāva*, *Bahuvrīhi*, *Dvandva*, and *Tatpuruṣa*

data and Marathi due to its closeness to Sanskrit. We freshly created annotated task-specific context-sensitive data for Marathi (§ 3.1) as no such dataset was previously available. Table 4 reports the results for these two languages. For English, all the baselines (with context) improve over their counterpart (without context). However, we do not find similar trend in Marathi possibly due to (1) lack of sufficient task-specific data, and (2) lack of both the auxiliary task¹⁴. For both languages, our system consistently outperforms all the competing systems. Across both the languages, it shows the average absolute gain of 4.7 points (A) and 4.4 points (F1) compared to the strong baseline XLM-R. Summarily, these empirical results prove the proposed approach’s efficacy in languages other than Sanskrit.

(4) Multi-lingual training and zero-shot cross-lingual transfer experiments for Marathi:

Here, we investigate the transferability of the SaCTI task for low-resourced languages. We experiment with the Marathi language (w/ context). Since the label space of Marathi is the same as that of the SaCTI coarse dataset, this makes it possible to experiment with cross-lingual zero-shot transfer

¹⁴We could not activate auxiliary tasks due to lack of datasets for Marathi.

and multi-lingual training. There is an isomorphic semantic type system with 4 types for Marathi as is the case for most of the Indian languages, due to a close connection with / inheritance from Sanskrit. In Table 5, we consider the mono-lingual (training on Marathi) results as a baseline. In multi-lingual training, we train the proposed system with mix of Marathi and SaCTI-base coarse dataset and evaluate on test set of Marathi.¹⁵ In the zero-shot transfer, we leverage the model trained on the SaCTI-base coarse dataset to get predictions on the test set of Marathi. In multi-lingual training experiment, we observe substantial improvements (3.4 points F1) over mono-lingual training. However, zero-shot cross-lingual transfer does not show encouraging results, where system predicts *Tatpuruṣa* type for majority test samples (80% predictions).

Tasks	A	P	R	F1
ours (Marathi)	80.43	66.54	77.00	69.12
zero-shot transfer	48.09	33.85	48.36	31.08
multi-lingual training	83.10	71.00	80.48	73.70

Table 5: Performance of the multilingual training and cross-lingual zero-shot transfer on Marathi.

(5) **Probing analysis:** Here, we probe the attention modules of the proposed system to investigate (1) How do different context words contribute towards final prediction? (2) Do these attentions serve as a proxy for explainability of correct/incorrect predictions? Figure 3 illustrates attention heatmaps for the SaCTI (Blue) and dependency parsing (Purple) tasks. The SaCTI heatmap shows how different context words contribute towards final prediction (Blue) and dependency parsing heatmap (Purple) serves as proxy for interpretation. We notice in the SaCTI attentions that all

¹⁵Here, we use pretrained models of Sanskrit for both the auxiliary tasks to obtain pseudo-labels for Marathi.

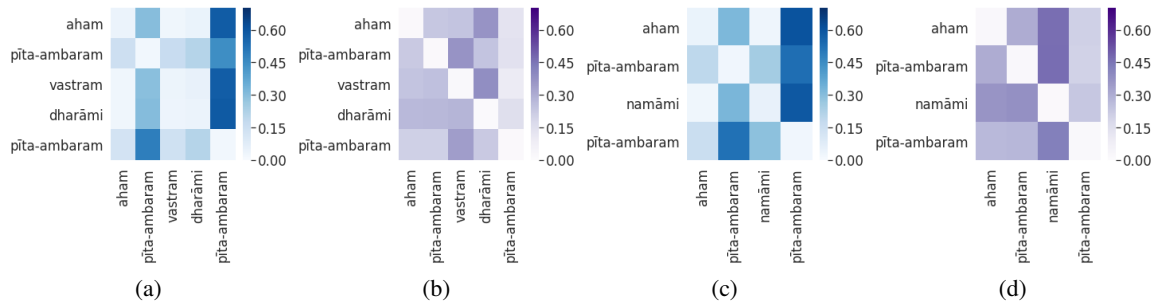


Figure 3: Attention heatmaps for the SaCTI (Blue) and dependency parsing (Purple) tasks. The SaCTI heatmap shows how different context words contribute towards final prediction (Blue) and dependency parsing heatmap (Purple) serve as proxy for interpretation. We illustrate how the same compound (*pīta-ambaram*) in two different contexts [(a-b) *aham pīta-ambaram vastram dharāmi* (I wear a yellow cloth) and (c-d) *aham pīta-ambarām namāmi* (I pray to the Lord Viṣṇu)] leads to different semantic type predictions (*Tatpuruṣa*: yellow cloth and *Bahuvrīhi*: Lord Viṣṇu). In the dependency heatmap of the first case, *pīta-ambaram* focuses on *vastram* (cloth) and in latter case it focuses on *namāmi* (the action of praying).

words mostly focus on the target compound word. Figure 3 illustrate how the same compound (*pīta-ambaram*) in two different contexts [(a-b) *aham pīta-ambaram vastram dharāmi* (I wear a yellow cloth) and (c-d) *aham pīta-ambarām namāmi* (I pray to the Lord Viṣṇu)] leads to different semantic type predictions (*Tatpuruṣa*: yellow cloth and *Bahuvrīhi*: Lord Viṣṇu). In the dependency heatmap of the first case, *pīta-ambaram* focuses on *vastram* (cloth) and in latter case, it focuses on *namāmi* (the action of praying). As per the grammatical rules, the morphological tagging task correctly predicts the gender information as neuter and masculine in these cases, respectively. Thus, this probing analysis suggests that auxiliary tasks not only help add complementary signals to the system but also serve as a proxy for explainability.

(6) Additional auxiliary tasks: With our proposed multi-task learning approach, we experiment with a few more additional sequence labeling auxiliary tasks (on SaCTI-base w/ context dev set), namely, the prediction of the case grammatical category (C), lemma prediction (L) and prediction of a relation (R) between modifier and its headword. The results in Table 6 show that except for the relation prediction task, all the remaining auxiliary tasks report improvements over the base system (with no auxiliary task). However, none of the combinations of these auxiliary tasks could outperform the proposed combination of morphological parsing and dependency parsing tasks. Therefore, we do not consider these additional auxiliary tasks in our final system.

Tasks	A	P	R	F1
BiAFF	87.99	85.48	87.90	86.64
+case (C)	87.64	85.65	88.90	87.19
+morph (M)	88.01	88.90	85.75	87.26
+relation (R)	86.61	85.53	84.99	85.22
+lemma (L)	87.43	88.27	86.21	87.18
+Dep. parse (DP)	89.14	86.49	90.10	88.01
M+C	87.55	88.30	85.28	86.72
M+C+L	87.08	87.74	84.98	86.30
M+C+R	86.10	86.04	83.28	84.55
M+DP	88.11	86.12	89.23	88.43

Table 6: The comparison (on SaCTI-base w/ context dev set) in between auxiliary tasks. ‘+’ denotes a system where the corresponding task is integrated with BiAFF.

(7) Web-based tool: We deploy our pretrained models as a web-based tool which facilitates the following advantages: (1) A naive user with no prior deep-learning expertise can use it for pedagogical purposes. (2) It can serve as a semi-supervised annotation tool keeping a human in the loop for the error corrections. (3) Our tool helps the user interpret the model prediction using model confidence on each semantic type and the probing analysis. Refer to Appendix A for our web-based tool’s interface. (4) It can be used for any general purpose classification task.

5 Related work

English Noun Compound Interpretation Prior to the deep learning era, various machine learning-based approaches have been proposed for Noun Compound Identification (Kim and Baldwin, 2005; Ó Séaghdha and Copestake, 2009; Tratz and Hovy,

2010). With the advent of deep-learning based approaches, Dima and Hinrichs (2015) and Fares et al. (2018) proposed a neural-based architecture where concatenated representations of a compound were fed to a feed-forward network to predict a semantic relation between the compound’s components. Shwartz and Waterson (2018) proposed an approach that combines labeling with paraphrasing. Recently, Ponkiya et al. (2021) proposed a novel approach using semantic label repository (Ponkiya et al., 2018b, FrameNet) where continuous label space embeddings are used to predict unseen labels. To the best of our knowledge, a context has never been used for the classification task for NCI. In paraphrasing line of modeling, Ponkiya et al. (2020) formulates paraphrasing as “fill-in-the-blank” problem to predict the “missing” predicate or preposition using pretrained language models.

Sanskrit Compound Type Identification task has garnered considerable attention of the researchers in the last decade. In order to decode the meaning of a Sanskrit compound, it is essential to figure out its constituents (Gérard, 2010; Mittal, 2010; Hellwig and Nehrlich, 2018), how the constituents are grouped (Kulkarni and Kumar, 2011), identify the semantic relation between them (Kumar, 2012) and finally generate the paraphrase of the compound (Kumar et al., 2009). Satuluri and Kulkarni (2013) and Kulkarni and Kumar (2013) proposed a rule-based approach where around 400 rules mentioned in Pāṇini’s grammar (Pāṇini, 500 BCE) were analysed from the perspective of compound generation and type identification, respectively. Recently, Sandhan et al. (2019) investigated whether a purely engineering data-driven approach competes with the performance of a linguistically motivated hybrid approach by Krishna et al. (2016). Summarily, no attention has been given to incorporating contextual information, which is crucial and cheaply available. We address this research gap and mark the new state-of-the-art results with substantial improvements.

6 Conclusion and Discussion

This work focused on Sanskrit compound type identification, where the task is to decode the semantic information hidden in the compound, which can be *context-sensitive*. This poses a limitation to the existing *context agnostic* approaches, thus we propose a novel multi-task learning architecture which incorporates the contextual information and also

enriches it with complementary syntactic information using morphological tagging and dependency parsing auxiliary tasks. Our probing analysis showcased that these auxiliary tasks also serve as a proxy for model prediction explainability. To the best of our knowledge, this is the first time that the importance of these auxiliary tasks has been showcased for SaCTI. Our experiments on benchmark datasets showed that the proposed system provides stunning improvements with 6.1 points (A) and 7.7 points (F1) absolute gain compared with the current state-of-the-art system. Our fine-grained analysis showcased some light on the inner engineering of the proposed system. Our multi-lingual experiments on English and Marathi languages proved the efficacy of the proposed system in other languages.

We limit our study to the *purely engineering* data-driven settings. We plan to extend the current work by augmenting logical rules (Li and Sriku-mar, 2019; Nandwani et al., 2019) derived from Pāṇinian grammar in the proposed approach.

Ethics Statement: We do not foresee any ethical concerns with the work presented in this manuscript.

Acknowledgements

We thank Amba Kulkarni for providing annotations for SaCTI datasets. We are grateful to Pavankumar Satuluri for helping us with insightful discussion. We would like to thank Narein Rao, Rathin Singha and the anonymous reviewers for their constructive feedback towards improving this work. We thank Om Adideva Paranjay and Ayush Daksh for helping us with the data pre-processing step. We would like to thank Jayesh Lakade, Prachi Rahangade and Aakash Biswas for helping us with the Marathi compound raw data collection. We thank Varsha Sandhan, Chitra Manjarekar, Khyati Deshpande and Rachit Bodhare for helping us with the Marathi data annotations. The work of the first author is supported by the TCS Fellowship under the Project TCS/EE/2011191P.

References

- 2022. Bootstrap 5.1. <https://getbootstrap.com/docs/5.1/getting-started/introduction/>.
- Rahul Aralikatte, Miryam de Lhoneux, Anoop Kunchukuttan, and Anders Søgaard. 2021. *Itihasa: A large-scale corpus for Sanskrit to English translation*.

- In *Proceedings of the 8th Workshop on Asian Translation (WAT2021)*, pages 191–197, Online. Association for Computational Linguistics.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Unsupervised cross-lingual representation learning at scale](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Corina Dima and Erhard Hinrichs. 2015. [Automatic noun compound interpretation using deep neural networks and word embeddings](#). In *Proceedings of the 11th International Conference on Computational Semantics*, pages 173–183, London, UK. Association for Computational Linguistics.
- Timothy Dozat and Christopher D. Manning. 2017. [Deep biaffine attention for neural dependency parsing](#). In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.
- Murhaf Fares. 2016. [A dataset for joint noun-noun compound bracketing and interpretation](#). In *Proceedings of the ACL 2016 Student Research Workshop*, pages 72–79, Berlin, Germany. Association for Computational Linguistics.
- Murhaf Fares, Stephan Oepen, and Erik Velldal. 2018. [Transfer and multi-task learning for noun-noun compound interpretation](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1488–1498, Brussels, Belgium. Association for Computational Linguistics.
- Huet Gérard. 2010. Sanskrit segmentation. In *SALA XXVIII*.
- Oliver Hellwig. 2010. Dcs-the digital corpus of sanskrit. *Heidelberg (2010-2022)*. URL <http://www.sanskrit-linguistics.org/dcs/index.php>.
- 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.
- Mark Johnson, Thomas L. Griffiths, and Sharon Goldwater. 2006. [Adaptor grammars: A framework for specifying compositional nonparametric bayesian models](#). In *NIPS*.
- Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. [The state and fate of linguistic diversity and inclusion in the NLP world](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.
- Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. [IndicNLP Suite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages](#). In *Findings of EMNLP*.
- Su Nam Kim and Timothy Baldwin. 2005. [Automatic interpretation of noun compounds using WordNet similarity](#). In *Second International Joint Conference on Natural Language Processing: Full Papers*.
- Dan Kondratyuk and Milan Straka. 2019. [75 languages, 1 model: Parsing Universal Dependencies universally](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2779–2795, Hong Kong, China. Association for Computational Linguistics.
- 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, Pavankumar Satuluri, and Pawan Goyal. 2020. [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, 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.
- Sriram Krishnan, Amba Kulkarni, and Gérard Huet. 2020. [Validation and normalization of dcs corpus using sanskrit heritage tools to build a tagged gold corpus](#).

- Taku Kudo. 2018. [Subword regularization: Improving neural network translation models with multiple subword candidates](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.
- Taku Kudo and John Richardson. 2018. [SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.
- Amba Kulkarni and Anil Kumar. 2011. Statistical constituency parser for sanskrit compounds. In *Proceedings of the International Conference on Natural Language Processing*.
- Amba Kulkarni and Anil Kumar. 2013. Clues from *asṭādhyāyī* for compound type identification. In *Proceedings of the Sanskrit Computational Linguistics Symposium*.
- Anil Kumar. 2012. *An automatic Sanskrit compound processing*. Ph.D. thesis, University of Hyderabad.
- Anil Kumar, V. Sheebasudheer, and Amba P. Kulkarni. 2009. Sanskrit compound paraphrase generator. In *The workshop on Panini and Paninians of 16th and 17th centuries, Pondicherry*.
- Mirella Lapata and Frank Keller. 2004. [The web as a baseline: Evaluating the performance of unsupervised web-based models for a range of NLP tasks](#). In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 121–128, Boston, Massachusetts, USA. Association for Computational Linguistics.
- Tao Li and Vivek Srikumar. 2019. [Augmenting neural networks with first-order logic](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 292–302, Florence, Italy. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized bert pretraining approach](#).
- John Lowe. 2015. [The syntax of sanskrit compounds](#). *Language*, 91:e71–e115.
- Mary McHugh. 2012. [Interrater reliability: The kappa statistic](#). *Biochemia medica : časopis Hrvatskoga društva medicinskih biokemičara / HDMB*, 22:276–82.
- Vipul Mittal. 2010. [Automatic Sanskrit segmentizer using finite state transducers](#). In *Proceedings of the ACL 2010 Student Research Workshop*, pages 85–90, Uppsala, Sweden. Association for Computational Linguistics.
- Yatin Nandwani, Abhishek Pathak, Parag Singla, et al. 2019. A primal dual formulation for deep learning with constraints. In *Advances in Neural Information Processing Systems*, pages 12157–12168.
- Minh Van Nguyen, Viet Dac Lai, Amir Pouran Ben Veysseh, and Thien Huu Nguyen. 2021. [Trankit: A light-weight transformer-based toolkit for multilingual natural language processing](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*.
- Diarmuid Ó Séaghdha and Ann Copestake. 2009. [Using lexical and relational similarity to classify semantic relations](#). In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 621–629, Athens, Greece. Association for Computational Linguistics.
- Paṇini. 500 BCE. [Asṭādhyāyī](#).
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Matthew E. 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.
- Girishkumar Ponkiya, Diptesh Kanojia, Pushpak Bhattacharyya, and Girish Palshikar. 2021. [FrameNet-assisted noun compound interpretation](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2901–2911, Online. Association for Computational Linguistics.
- Girishkumar Ponkiya, Rudra Murthy, Pushpak Bhattacharyya, and Girish Palshikar. 2020. [Looking inside noun compounds: Unsupervised prepositional and free paraphrasing](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4313–4323, Online. Association for Computational Linguistics.
- Girishkumar Ponkiya, Kevin Patel, Pushpak Bhattacharyya, and Girish Palshikar. 2018a. [Treat us like the sequences we are: Prepositional paraphrasing of noun compounds using LSTM](#). In *Proceedings of the*

27th International Conference on Computational Linguistics, pages 1827–1836, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Girishkumar Ponkiya, Kevin Patel, Pushpak Bhattacharyya, and Girish K Palshikar. 2018b. [Towards a standardized dataset for noun compound interpretation](#). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Armin Ronacher. Opening the flask. [URL: http://mitsuhiko.pocoo.org/flask-pycon-2011.pdf](http://mitsuhiko.pocoo.org/flask-pycon-2011.pdf).

Jivnesh Sandhan, Om Adideva, Digumarthi Komal, Laxmidhar Behera, and Pawan Goyal. 2022. [Evaluating neural word embeddings for sanskrit](#). In *the 18th World Sanskrit Conference*. Canberra, Australia.

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.

Pavankumar Satuluri and Amba Kulkarni. 2013. Generation of sanskrit compounds. In *Proceedings of the International Conference on Natural Language Processing*.

Vered Shwartz and Chris Waterson. 2018. [Olive oil is made of olives, baby oil is made for babies: Interpreting noun compounds using paraphrases in a neural model](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 218–224, New Orleans, Louisiana. Association for Computational Linguistics.

Hrishikesh Terdalkar and Arnab Bhattacharya. 2019. [Framework for question-answering in Sanskrit through automated construction of knowledge graphs](#). In *Proceedings of the 6th International Sanskrit Computational Linguistics Symposium*, pages 97–116, IIT Kharagpur, India. Association for Computational Linguistics.

Stephen Tratz and Eduard Hovy. 2010. [A taxonomy, dataset, and classifier for automatic noun compound interpretation](#). In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 678–687, Uppsala, Sweden. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#).

In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. [Xlnet: Generalized autoregressive pretraining for language understanding](#). In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

A Web-based tool

We deploy our pretrained models as a web-based tool. Figure 4 illustrates the web-based tool integrated with our best performing system. Input: “*aham pīta-ambaram namāmi*” (Translation: “I pray to Pītāmbara (Lord Vishnu).”) where ‘*pīta-ambaram*’ is a compound word. Our interface shows predicted morphological tags (color-coded with violet boxes), type-wise system confidence (bar plot), attention heatmaps. Our tool helps the user interpret the model prediction using model confidence on each semantic type.

B Marathi annotation details

We have built a user-friendly annotation tool for general-purpose classification tasks. The tool is a Flask-based (Ronacher) user-friendly web application styled by Bootstrap 5 (boo, 2022), and sports a simple administrative interface that lets the administrators easily control the class labels and classification context as well as export annotations. Figure 5 shows our web-based mobile friendly annotation interface, where the task is to select the correct option from multiple-choices for the given compound with the context. There are 4 broad categories of semantic types in Marathi. In case of confusion, we gave additional option to mark as “Not sure”. We also provide an option to add a comment to convey additional information about ambiguity or the concern for the corresponding example.

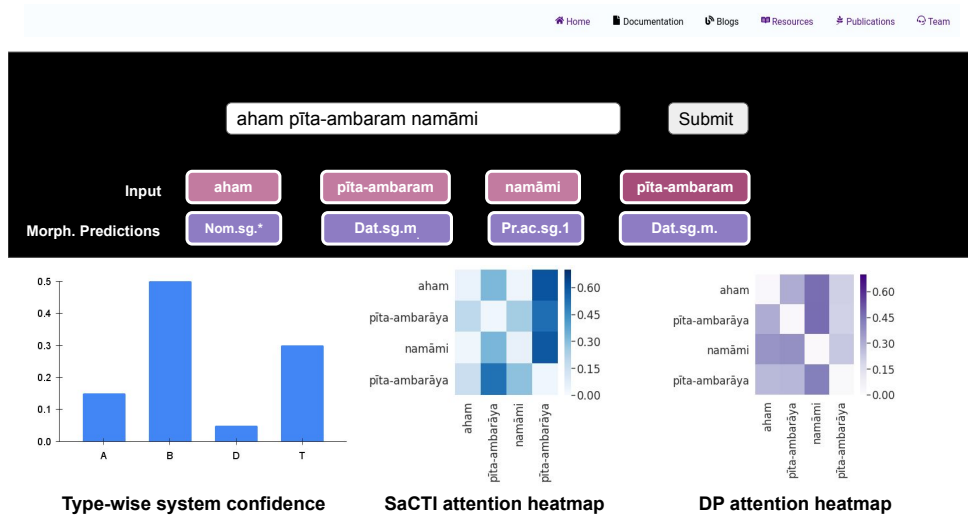


Figure 4: Illustration of web-based tool integrated with our best performing pretrained system. Input: “*aham pīta-ambaram namāmi*” (Translation: “I pray to Pītāmbara (Lord Vishnu).”) where ‘*pīta-ambaram*’ is a compound word. Our interface shows predicted morphological tags (color-coded with violet boxes), type-wise system confidence (bar plot), attention heatmaps.

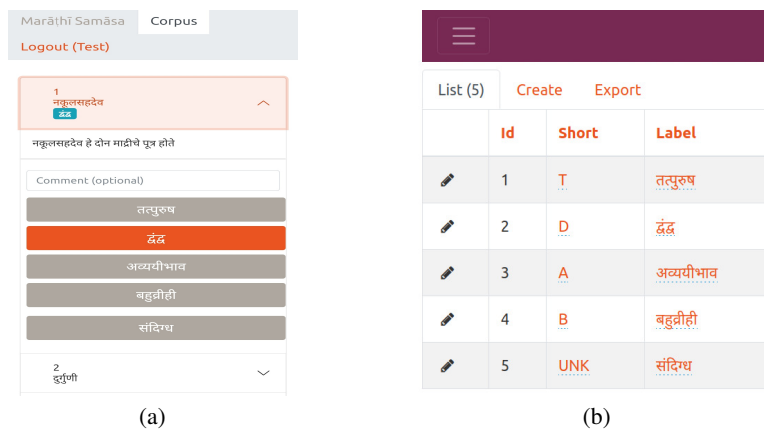


Figure 5: Illustration of our web-based mobile friendly (a) annotation interface, where the task is to select the correct option from multiple-choices for the given compound with the context. (b) Administrative interface, where the administrators can easily control the class labels and classification context as well as export annotations.