

mahaNLP: A Marathi Natural Language Processing Library

Vidula Magdum^{1,3}, Omkar Dhekane^{1,3}, Sharayu Hiwarkhedkar^{1,3}, Saloni Mittal^{1,3}
Raviraj Joshi^{2,3}

¹ Pune Institute of Computer Technology, Pune, Maharashtra India

² Indian Institute of Technology Madras, Chennai, Tamil Nadu India

³ L3Cube Pune

{vidulamagdum12, omkarjd1212, hiwarkhedkarsharayu, salonimittal12}@gmail.com
ravirajjoshi@gmail.com

Abstract

We present mahaNLP, an open-source natural language processing (NLP) library specifically built for the Marathi language. It aims to enhance the support for the low-resource Indian language Marathi in the field of NLP. It is an easy-to-use, extensible, and modular toolkit for Marathi text analysis built on state-of-the-art MahaBERT-based transformer models. Our work holds significant importance as other existing Indic NLP libraries provide basic Marathi processing support and rely on older models with restricted performance. Our toolkit stands out by offering a comprehensive array of NLP tasks, encompassing both fundamental preprocessing tasks and advanced NLP tasks like sentiment analysis, NER, hate speech detection, and sentence completion. This paper focuses on an overview of the mahaNLP framework, its features, and its usage. This work is a part of the L3Cube MahaNLP initiative, more information about it can be found at <https://github.com/l3cube-pune/MarathiNLP>.

1 Introduction

Natural Language Processing (NLP) is a major subset of artificial intelligence. It helps to clear up linguistic ambiguity and gives the data a useful quantitative structure for numerous downstream tasks. While NLP has the potential to have a huge impact on the ML community, recent models have primarily focused on English and 6 other languages¹ with a significantly high amount of resources (Joshi et al., 2020). There are around 7,000 languages spoken worldwide². Out of which, approximately 22+ languages are existing Indian languages³ that are widely spoken not only in India but also throughout the world. Developing models that work for these languages is important for a variety of reasons, including bridging the existing language divide and promoting explo-

¹<https://datasaur.ai/blog-posts/nlp-work-in-different-languages>

²https://en.wikipedia.org/wiki/Lists_of_languages

³https://en.wikipedia.org/wiki/Languages_of_India

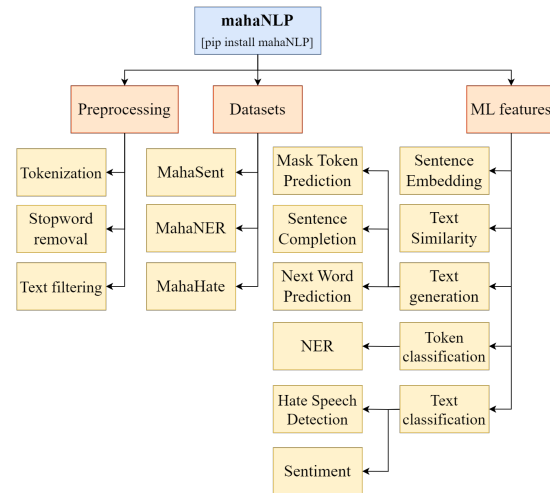


Figure 1: A brief overview of the features in mahaNLP library

ration and research for non-English languages in the growing NLP field⁴.

The Marathi language is 11th in the list of popular languages across the globe⁵. Despite being a widely spoken language, Marathi-specific NLP monolingual resources are still limited in comparison to other natural languages (Joshi, 2022a). There are many popular open-source tools like Spacy⁶, Stanza⁷, iNLTK⁸, IndicNLP (Kunchukuttan, 2020), providing features to create effective multilingual NLP systems. Though these libraries do support the multilingual functionality for Marathi, they are not exhaustive with respect to the functionalities they support and have their own set of limitations and discrepancies, such as the usage of older architectures to implement multilingual models (Ruder et al., 2021). Basic

⁴<https://ruder.io/state-of-multilingual-ai/index.html>

⁵https://en.wikipedia.org/wiki/Marathi_language

⁶<https://spacy.io/>

⁷<https://stanfordnlp.github.io/stanza/>

⁸<https://inlntk.readthedocs.io/en/latest/>

	Lemmatization	Part of Speech Tagging	Dependency Parsing	Multi-word token expansion	Sentence Segmentation	Sentence Embedding	Remove Foreign Languages	Sentence Similarity	Sentiment Analysis	Hate Speech Detection	Named Entity Recognition	Next Token Prediction
iNLTK						✓	✓	✓				✓
Stanza	✓	✓	✓	✓	✓						✓	✓
spaCy					✓	✓		✓				
mahaNLP					✓	✓	✓	✓	✓	✓	✓	✓

Figure 2: A comparison of a few existing libraries and mahaNLP specific to the Marathi language

features like Marathi sentiment analysis, named entity recognition, and hate speech detection is missing in almost all the current multilingual libraries. Additionally, the absence of out-of-the-box features present in other libraries (yet served by L3Cube-mahaNLP library) makes it difficult for the comparative analysis of these models.

The L3Cube-MahaNLP⁹ (Joshi, 2022b) initiative is an umbrella for various tools, datasets, and models that greatly assist in Marathi language processing. With this work, we aim to provide a broader set of functionalities to developers. In this paper, we propose mahaNLP library - a python-based NLP toolkit focused on the Indian language Marathi and majorly built on top of Hugging Face transformer models¹⁰. Along with basic language processing features, the open-source toolkit wraps the state-of-the-art monolingual Marathi transformer models for text analysis. It encompasses a myriad of features like sentiment and hate analysis, named entity recognition, and a variety of other Marathi language processing features as shown in Figure 1. Thus, the mahaNLP library aims to make Marathi NLP more accessible. The demonstration video¹¹ and example colab¹² are shared publicly.

2 Related Work

The communities of NLP and machine learning have a prolonged history of developing open-source tools and libraries. There are numerous user-friendly, all-purpose NLP libraries available. NLTK (Bird, 2006), Stanford CoreNLP (Manning et al., 2014), Spacy (Honnibal and Montani, 2017), AllenNLP (Gardner et al., 2018), Flair (Akbik et al., 2019), Stanza (Qi et al., 2020), Hugging Face Transformers (Wolf et al., 2019) and iNLTK (Arora, 2020) are some of the libraries that are primarily concerned with NLP tasks. These libraries

provide NLP tasks to the ML community like tokenization, sentence encoding, text normalization, translation, and so on.

However, while some of these libraries do support Marathi, they have a very limited range of NLP tasks and pre-trained language models. Figure 2 illustrates the provisional comparison between mahaNLP and the existing libraries for the language Marathi. The iNLTK library is not actively supported, it is based on very limited datasets and LSTM-based ULMFiT models. Marathi models for features like sentiment and NER are unavailable in SpaCy. MahaNLP attempts to address these issues by offering a broader set of pre-trained language models, including BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), and ALBERT (Lan et al., 2019), as well as a broader set of NLP tasks such as sentiment analysis, hate speech analysis, NER Tagger, and MLM-based modules. The NER model supported by Stanza is based on our L3Cube-MahaNER corpus (Litake et al., 2022)¹³.

3 System Design and Architecture

The *ease-of-access* is an important characteristic of mahaNLP and the system is designed from the user’s perspective. These perspectives can be mainly defined as - a *Standard Flow* and a *Model Flow*.

3.1 Standard Flow

The *Standard Flow* can be used by a basic programmer who has the least knowledge of the machine learning domain. In this flow, the complex model arguments are isolated from the users. They can use a feature without knowledge of the models used in the background. This flow has constructs similar to standard NLP libraries. The intuitive nature of this flow makes mahaNLP more user-friendly and easily accessible. The pre-processing,

⁹<https://github.com/l3cube-pune/MarathiNLP>

¹⁰<https://huggingface.co/l3cube-pune>

¹¹<https://youtu.be/KxExcwCrT00>

¹²Example colab

¹³https://stanfordnlp.github.io/stanza/ner_models.html

tokenizer, datasets, and ML-based models are part of this user flow.

3.1.1 Pre-process Module

An initial step in any NLP task is the preprocessing of data. The transformer models in NLP perform much better on cleaned data than raw, unprocessed corpus. The preprocess module helps to provide functions such as the removal of URLs, stopwords, and non-Devanagari words for cleaning Marathi textual data.

3.1.2 Tokenizer Module

A very important step in many NLP tasks is the tokenization of text. The mahaNLP Tokenizer module provides functionalities for sentence-level tokenization (splitting into multiple sentences) and word-level tokenization (splitting into multiple words).

3.1.3 Datasets Module

The mahaNLP library currently supports 3 datasets - MahaHate (Patil et al., 2022) for hate speech detection, MahaSent (Kulkarni et al., 2021) (Pingle et al., 2023) for sentiment analysis, and MahaNER (Litake et al., 2022) for named entity recognition. These datasets can mainly be used for finetuning tasks. These datasets are given as-is in our library and have an official train-test-valid split. Each of these corpora can be separately loaded in the pandas data frame. The datasets are cached locally to avoid repeated downloads.

Feature name	Labels / Outputs	Function name(s)
Sentiment	positive, negative, neutral	get_polarity_score
Hate	hate, non-hate	get_hate_score
Tagger	Person (NEP), Location (NEL), Organization (NEO), Measure (NEM), Time (NETI), Date (NED), Designation (ED)	get_token_labels, get_tokens
Auto Complete	next (N) words	next_word, complete_sentence
Mask Fill	masked word	predict_mask
Similarity	similarity score (0-1)	embed_sentences, get_similarity_score

Table 1: Standard Flow ML features

3.1.4 Machine learning-based modules

A set of modules utilizes machine learning models in the background to provide the desired functionality. Currently, these features or modules are implemented using state-of-the-art Marathi Transformer models and are described below. The basic syntax to use them is:

```
module_name = Module()
result = module_name.functionality_name(text)
```

Table 1 shows the features supported, output type, and the corresponding functions.

3.2 Model Flow

The *Model Flow* provides advanced functionality intended for use by ML practitioners with knowledge in the NLP domain. It offers flexibility for programmers to select background models and adjust their parameters. The *model_repo* module defines the model flow. Table 2 presents the association between standard flow machine learning-based modules and *model_repo* submodules. It also reports the accuracy/F1 scores for pertinent features on the benchmark test sets.

4 System Usage

The mahaNLP library is hosted on the official PyPI repository. It can be installed using the pip command:

```
pip install mahaNLP
```

Once installed, we can then import the required features using the python import statement and start utilizing various functionalities. Currently, the library has been tested on x64-bit Windows 10 OS and Google Colab platform (posix-linux).

4.1 Dataset Loading

A snippet of code demonstrating the loading *mahaSent* dataset provided by mahaNLP is given below:

```
# import
from mahaNLP.datasets import *

# loadDatasets: This loads the dataset into our variable
# parameter:
# name of the dataset

dataset = load_datasets('mahaSent')
```

Standard Flow Module	model_repo submodule	Huggingface model	Score(Acc/F1)
sentiment	SentimentModel	marathi-sentiment-md (Pingle et al., 2023)	84.90
hate	HateModel	mahahate-bert (Patil et al., 2022)	90.9
tagger	NERModel	marathi-ner (Litake et al., 2022)	86.8
autocomplete	GPTModel	marathi-gpt (Joshi, 2022a)	-
mask_fill	MaskFillModel	marathi-bert-v2 (Joshi, 2022a)	-
similarity	SimilarityModel	marathi-sentence-similarity-sbert (Joshi et al., 2022)	83

Table 2: Table shows which *model_repo* submodules are inherited by the Standard Flow modules and the default huggingface model internally used. The huggingface models can be listed and selected in the Model Flow. The score column indicates the accuracy of the model on standard test sets as reported in original works. The scoring metric of the sentiment and hate model is accuracy while that of the NER and similarity model is F1-score and embedding similarity score respectively.

4.2 Usage via Standard Flow

In *Standard Flow*, the user can simply import the feature they want to use (e.g. autocomplete, sentiment, tagger, etc.) and define the object to initialize that particular model. Here, the user can optionally pass the *model_name* as an argument during model initialization.

A simple snippet of generic flow usage for *SentimentAnalyzer* class object is given below:

```
# import
from mahaNLP.sentiment import SentimentAnalyzer

# create a model object
model = SentimentAnalyzer()

# function - getPolarityScore
# It gives the sentiment score in the range of 0 to 1
# for 3 poles: Neutral, Positive, Negative

# example:
text = 'दिवाळीच्या सणादरम्यान सगळे आनंदी असतात.'
# English Translation:
# 'Everyone is happy during Diwali festival.'

polarityscore = model.getPolarityScore(text)
print(polarityscore)

label    score
0 Positive 0.995338
```

The *get_polarity_score* function returns *float* value representing the confidence score for a predicted sentiment class.

4.3 Usage via Model Flow

In *Model Flow*, the user has to import the specific model (e.g. mahaHate, mahaNER, etc.) using `import mahanlp.model_repo.modelname`. Then, the user can define the model object and also can optionally pass the *model_name* as an argument. Refer the following code for loading and usage of the *MaskFillModel* class object via *Model Flow*.

```
# import
from mahaNLP.model_repo import MaskFillModel

# create a model object
# parameter:
# model name
model = MaskFillModel('marathi-bert-v2')

# function - predictMask
# It predicts the masked token

text = 'मी महाराष्ट्रत [MASK].'
# English Translation:
# 'I in Maharashtra [MASK]'

# parameters:
# text, details of prediction, print in dict

model.predict_mask(text, details = "all", as_dict = False)
```

The predicted token string, sequence, and score can be returned as:

	score	token	token_str	sequence
0	0.465607	1155	आहे	मी महाराष्ट्रात आहे.
1	0.079688	92222	राहणार	मी महाराष्ट्रात राहणार.
2	0.074000	1826	नाही	मी महाराष्ट्रात नाही.
3	0.050423	1617	##च	मी महाराष्ट्रातच.
4	0.043737	62560	राहतो	मी महाराष्ट्रात राहतो.

The library also supports various standard hardware devices. The *gpu_enabled = True* option allows users to utilize the gpu for model usage or inference.

Table 3 demonstrates all the modules of MahaNLP along with inputs and their expected outputs. The detailed description and usage of all the functionalities in mahaNLP are available at the [mahaNLP PyPI project](#).

5 Conclusion and Future Work

We have built mahaNLP - a simple, easy-to-use and extensible toolkit for Marathi language pro-

Functionality	Models Used	Input	Output
Sentiment Analysis	marathi-sentiment-md	दिवाळीच्या सणादरम्यान सगळे आनंदी असतात. (Everyone is happy during Diwali festival.)	Label: Positive Score: 0.995
Hate Analysis	mahahate-bert mahahate-multi-roberta	ती मूर्ख आहे. मला ती आवडत नाही. (She is stupid. I don't like her.)	Label: Hate Score: 0.966
Masked Token Prediction	marathi-bert-v2 marathi-roberta marathi-albert	मी महाराष्ट्रात [MASK].	मी महाराष्ट्रात राहतो . (I live in Maharashtra)
Autocomplete	marathi-gpt	आज खूप पाऊस. (A lot of rain today)	आज खूप पाऊस पडला होता. यामुळे ही समस्या गंभीर बनली आहे. (It rained a lot today. That's why this problem has gotten serious)
Sentence Similarity	marathi-sentence-similarity-sbert marathi-sentence-bert-nli	Source Text: त्याला पाकिटात हजार रुपये मिळाले. (He found 1000 rupees in his wallet.) Target Text: त्याला पाकिटात काही पैसे सापडले. (He found some cash in his wallet.)	Similarity: 0.781
Entity Recognition	marathi-ner	१७ तारखेला आदित्य पुण्यात आला. (Aditya came to Pune on 17th.)	Date Other Person Location Other

Table 3: Sample examples of features provided by mahaNLP

cessing and development of robust NLP models. The mahaNLP is majorly built on the top of the huggingface transformers¹⁴ library and utilizes pyTorch¹⁵ framework under the hood. It also utilizes the resources developed by L3Cube as a part of MahaNLP initiative. It primarily provides wrapper classes to perform downstream tasks on supervised datasets, transformer models, and other tools for natural language processing in Marathi. The paper demonstrates the usage of various modules in the mahaNLP library.

We are working on expanding our project scope in the following manner:

- Expanding the project to support more low-resource Indic languages.
- Extending the existing corpora provided by the library and supporting Indo-Aryan language datasets and other regional language datasets like Konkani and Dongari which are very prevalent in Maharashtra state.

¹⁴<https://huggingface.co/l3cube-pune>

¹⁵<https://pytorch.org/>

- Addition of pretrained models for advanced downstream tasks like machine translation, natural language inference etc.
- Creation of web-based user interactive tools or APIs which can be used by developers at the production level.
- Further improvements in model accuracy for predictions and reduction in overall model size and computational cost.

Limitations

- Pre-processing can be made more efficient by expanding the dataset for stopwords to a more exhaustive list.
- The state-of-art huggingface transformer models have not been tested on a generic domain, but only on specific domains such as the news or social media domain. More generic models built as a part of a broader initiative will be integrated into the library.

- The current language models provided by ma-haNLP are built on transformers that have high runtime on CPU. In the future, we plan to build and integrate more compact models.

These issues are planned to be addressed in the upcoming versions of the library.

References

- Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. [Flair: An easy-to-use framework for state-of-the-art nlp](#). In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics (demonstrations)*, pages 54–59.
- Gaurav Arora. 2020. [inltk: Natural language toolkit for indic languages](#). *arXiv preprint arXiv:2009.12534*.
- Steven Bird. 2006. [Nltk: the natural language toolkit](#). In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 69–72.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. [Bert: Pre-training of deep bidirectional transformers for language understanding](#). *arXiv preprint arXiv:1810.04805*.
- Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. [Allennlp: A deep semantic natural language processing platform](#). *arXiv preprint arXiv:1803.07640*.
- Matthew Honnibal and Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7(1):411–420.
- Ananya Joshi, Aditi Kajale, Janhavi Gadre, Samruddhi Deode, and Raviraj Joshi. 2022. L3cube-mahasbert and hindsbert: Sentence bert models and benchmarking bert sentence representations for hindi and marathi. *arXiv preprint arXiv:2211.11187*.
- 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.
- Raviraj Joshi. 2022a. [L3Cube-MahaCorpus and MahaBERT: Marathi monolingual corpus, Marathi BERT language models, and resources](#). In *Proceedings of the WILDRE-6 Workshop within the 13th Language Resources and Evaluation Conference*, pages 97–101, Marseille, France. European Language Resources Association.
- Raviraj Joshi. 2022b. [L3cube-mahanlp: Marathi natural language processing datasets, models, and library](#). *arXiv preprint arXiv:2205.14728*.
- Atharva Kulkarni, Meet Mandhane, Manali Likhitkar, Gayatri Kshirsagar, and Raviraj Joshi. 2021. L3cubemahasent: A marathi tweet-based sentiment analysis dataset. In *Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 213–220.
- Anoop Kunchukuttan. 2020. The IndicNLP Library. https://github.com/anoopkunchukuttan/indic_nlp_library/blob/master/docs/indicnlp.pdf.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. [Albert: A lite bert for self-supervised learning of language representations](#). *arXiv preprint arXiv:1909.11942*.
- Onkar Litake, Maithili Ravindra Sabane, Parth Sachin Patil, Aparna Abhijeet Ranade, and Raviraj Joshi. 2022. L3cube-mahaner: A marathi named entity recognition dataset and bert models. In *Proceedings of The WILDRE-6 Workshop within the 13th Language Resources and Evaluation Conference*, pages 29–34, Marseille, France. European Language Resources Association.
- 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](#). *arXiv preprint arXiv:1907.11692*.
- Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. [The stanford corenlp natural language processing toolkit](#). In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.
- Hrushikesh Patil, Abhishek Velankar, and Raviraj Joshi. 2022. L3cube-mahahate: A tweet-based marathi hate speech detection dataset and bert models. In *Proceedings of the Third Workshop on Threat, Aggression and Cyberbullying (TRAC 2022)*, pages 1–9.
- Aabha Pingle, Aditya Vyawahare, Isha Joshi, Rahul Tangsali, and Raviraj Joshi. 2023. L3cube-mahasent-md: A multi-domain marathi sentiment analysis dataset and transformer models. *arXiv preprint arXiv:2306.13888*.
- Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020. [Stanza: A python natural language processing toolkit for many human languages](#). *arXiv preprint arXiv:2003.07082*.

Sebastian Ruder, Noah Constant, Jan Botha, Aditya Siddhant, Orhan Firat, Jinlan Fu, Pengfei Liu, Junjie Hu, Dan Garrette, Graham Neubig, et al. 2021. Xtreme-r: Towards more challenging and nuanced multilingual evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10215–10245.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. [Huggingface’s transformers: State-of-the-art natural language processing](#). *arXiv preprint arXiv:1910.03771*.