iJRASET
International Journal For Research in Applied Science and Engineering Technology

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 13 **Issue:** XII **Month of publication:** December 2025

**DOI:** https://doi.org/10.22214/ijraset.2025.76109

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 13 Issue XII Dec 2025- Available at www.ijraset.com*

# BEGINNLP-Beginner Friendly Natural Language Processing Library for Python

Vighnesh K.Gupta[1], Yash Wadhwani[2], Manasvi Vinnu[3], Vihan Saraf[4], Srujal Vispute[5], Aditya Wagh[6], Prof. Geeta Zaware[7]

[1, 2, 3, 4, 5, 6] *Student of Department of Engineering, Sciences and Humanities (DESH), Vishwakarma Institute of Technology Pune, India*

[7]*Project Guide, Professor, VIT Pune*

***Abstract:*** ***Natural Language Processing is the most useful tool in the current dynamics where Artificial Intelligence is becoming an integral part of everything we do today. With beginnlp we are making an effort to make this world of NLP more accessible to beginner level tech enthusiasts to begin with NLP with prebuilt easy to you functions like Text Summarization, Keyword Extraction, Translation, Name Entity Recognition, Grammar and Spelling correction, Text Preprocessing and more.***

***Keywords: Natural Language Processing, NLP, Python Library, Text Preprocessing, Artificial Intelligence.***

## I. INTRODUCTION

In the current world dynamics where Artificial Intelligence has become an important part in all our lives and fields and is being integrated in almost every software we use including our social media platforms, it's important for the future developers to have a good grasp on Natural Language Processing.

NLP is the branch of Artificial Intelligence which enables us to understand, interpret and use the natural day to day language used by us in our conversations. From understanding the prompt given by the user to giving a simple understandable answer to the queries, NLP plays a vital role in all the steps involved in Artificial Intelligence and how they interact with people. It is important for future developers to have a good grasp on NLP as the current trends shows that AI will be the most demanded and developing field in the upcoming decade but as a beginner it is hard for people to understand the complex working and mechanisms of NLP because of the mainstream libraries and modules which are harder for new developers to understand and use. The purpose of our library beginnlp is to provide simple to use basic features such as text summarization, translation, grammar and spell check etc and enable everyone to use them.

The current mainstream libraries are all spread out and often the user has to import two or more libraries to perform these basic tasks but in beginnlp we provide these features built in just a single library which are easy to use and understand. These basic functions are what are used in almost every NLP project and is useful to perform these basic tasks.

## II. LITERARY REVIEW

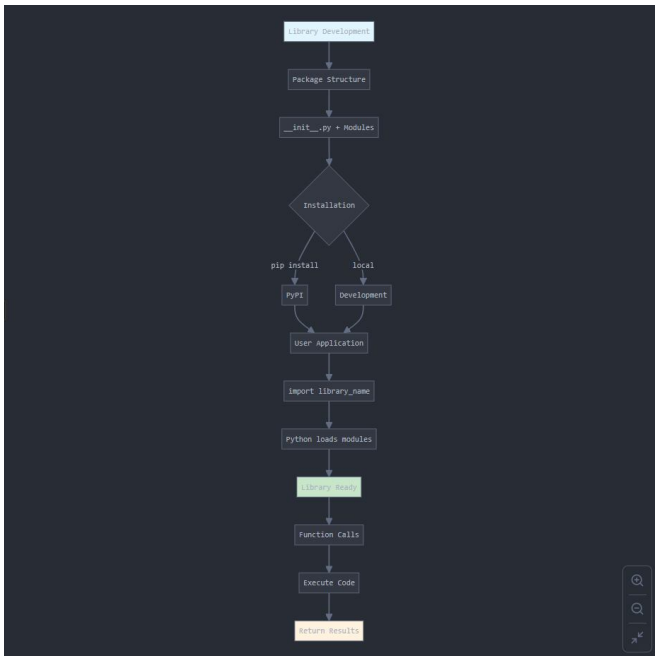
In the Stanza Python NLP library by Yuhao Zhang et al [1] the authors' primary goal was to build a Name Entity Recognition that goes beyond the standard Stanford CoreNLP and spaCy as they lacked customized model for language processing in the field of medical studies as even the scispaCy extended model of spaCy wasn't well suited for it and they did that by building their own pipeline through the Stanza NLP library and combining multiple datasets for the Name Entity Recognition dataset.

For lambeq by Dimitri Kartsaklis, et el [2] as presented by the authors to be the first high level Python library which has specially been designed for Quantum Natural Language Processing. From the previous researches we can see that there is a possibility of running simple NLP tasks on the Quantum hardware but lambeq is supposedly the very first attempt to make a practical implementation for this. The research primarily focuses on the quantum architecture and modularity of lambeq.

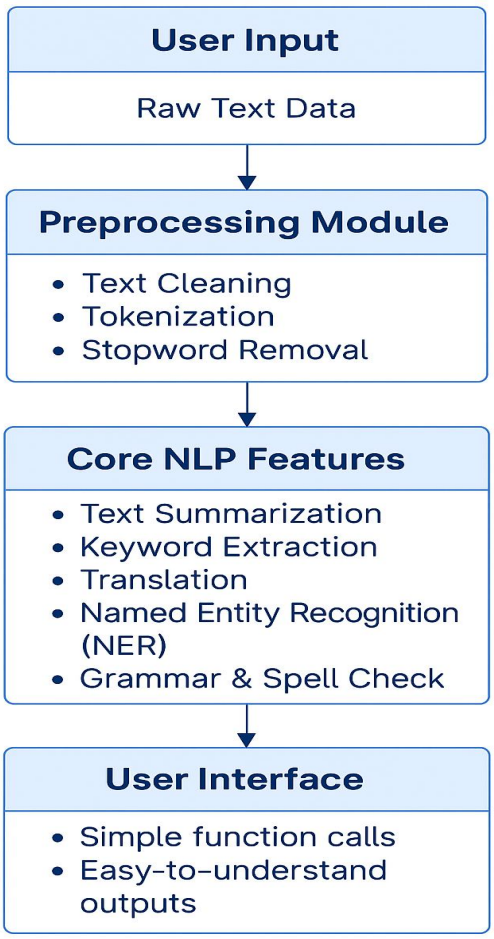
Similar to our library, TextCL by Alina Petukhova, et el [3] focuses on pre processing tasks which help to make long lines of codes to just a few lines by giving multiple preprocessing module. The primary focus of TextCL is to make easy to make module and user friendly library that is easy to use for text data preprocessing in just a few lines.

*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 13 Issue XII Dec 2025- Available at www.ijraset.com*




## III. METHODOLOGY



This project involves the use of several features that are listed as follows:

*1)* Text Summarization**:** For abstractive summarization we are using Hugging face transformer with T5-base models for summarization and tokenizer. This returns a shorter version of the original string which is a completely new string and holds the very same meaning. It works more like in a way humans summarize the text by reading the string, understanding it and then explaining it in their words. Model like T5 include an encoder that understands the context and meaning behind a string and creates a shorter smoother string. This is much more natural and human like by also keeping the grammar intact. The only disadvantage is that it requires internet connection to run and is slower. For extractive summarization we use spaCy library to tokenize keywords and give them all a score by seeing their frequency and then creating the summary of the original string on the basis of the highest scored keywords. In this we first tokenize the sentences by breaking them down to words then removing the stop words and keeping only the keywords. Then on the basis of the frequency of easy keyword we give it a score and then sentences with highest scores are stitched together. This method does not summarize in a very natural or human way and ignores grammar.

*2)* Grammar and Spell Check: Here we use the language_tool_python library to get a list of the errors in grammar and spellings in a string and return the correct words of the string. First we create a LanguageTool object and perform the .check() operation on the object which return a match for each error in the sentence and each match consists of position of the error, the error and the suggested fix. Then we fix these errors and return a corrected sentence.

*3)* Text Pre-processing: This function performs various steps using spaCy and other libraries in the order, lowercase the text, remove punctuation, remove digits and remove white spaces and lemmatize if lemmatization is on. It finally joins together these keywords to form a pre-processed string.

*4)* Name Entity Recognition: We use spaCy to collect all the named entities from a string and return a tuple containing the name of the entity and the category such are ORG, MON etc. We are loading a pretrained language model of en_core_web_sm. This model has vocabulary, pretrained statistical weights and a pipeline with components for tokenizer, tagger, parser and NER. We make a nlp object and tokenize the given sentence then the NER searches for the named entities present in the string and later we print these elements with their tags using a loop in form of tuples.

*5)* Keyword Extraction: In this first we take the input from the user and give them context related keywords. Here we're using the keybert library that is used for keyword extraction as well as put embeddings to word. A defined function that extracts keyword firstly. We have put keywords with their score and basically on their score we are comparing it to the context of the overall text and keeping top_n keywords with their score in a list keywords_with_score. Then we have made another list keywords where we have saved only the keyword part of the upper variable. We return the keywords because we only want that then we have taken input user text and also taken that how many related keywords they want. In the end in a variable we have called the function that we already made that returns a list containing keywords and then print them as well users text.

*6)* Translation: For language translation we have used an online free API (mymemory) that takes three inputs first the text to be translated , second the source language code(detected language of the input string) , third the required language and then provides the translated text so for the second input that is language detection we have used fasttext which is a pre - trained model used for language detection which can detect up to 174 languages accurately

*7)* Speech-To-Text: For this we're using the Whisper model which is a STT model from OpenAI. It can take audio from any source like a microphone or video and this audio is convert usually in waveform or spectrogram form which the model can understand and returns simple text as an output. It is designed to understand different languages and outputs.

*8)* Text-To-Speech: Here we're using the Coqui TTS tool which is a deep learning model. It first converts the text into a spectrogram which is like the visual form of an audio and this spectrogram is later used by a vocoder to return a human like speech.

## IV. RESULTS AND DISCUSSION

Each module of the library has been tested in various forms of conversations ranging from casual conversations to academic texts and it has performed well in all those standards. The most effective part of the library is that it's very modular and easy to use and reduces the lines of code needed to perform minor to major tasks in just a few lines of code. For example the STT model was able to return for text data for audio forms accurately. Similarly, sentiment analysis was able to analyse the emotions of a text and so on. As of now the library lacks quantitative benchmarking.

## V. CONCLUSION

In conclusion we can see that through the made easy functions it Is easier for beginner developers to smoothen out the learning curve for Natural Language Processing. It provides the users with easily accessible feature which are used in day to day Natural Language Processing like text preprocessing, keyword extraction etc. For future we can work on fixing the limitations of the project.

## VI. LIMITATIONS AND FUTURE SCOPE

There are still many limitations to our library. Like

*1)* Though for most of the uses we need to import only our library but the user still needs to download multiple libraries so we can work on using lesser libraries.
*2)* Models like Whisper for STT are accurate but the processing time is comparatively higher than many models so we can try faster models which are also accurate.
*3)* Some modules/functions like translation and abstractive text summarization require an internet connection. We can make these functions available offline.
*4)* We can try more methods for doing functions like keyword extraction and extractive text summarization.
*5)* The dataset used for features like Name Entity Recognition and Sentiment analysis can be increased in size to get better results.

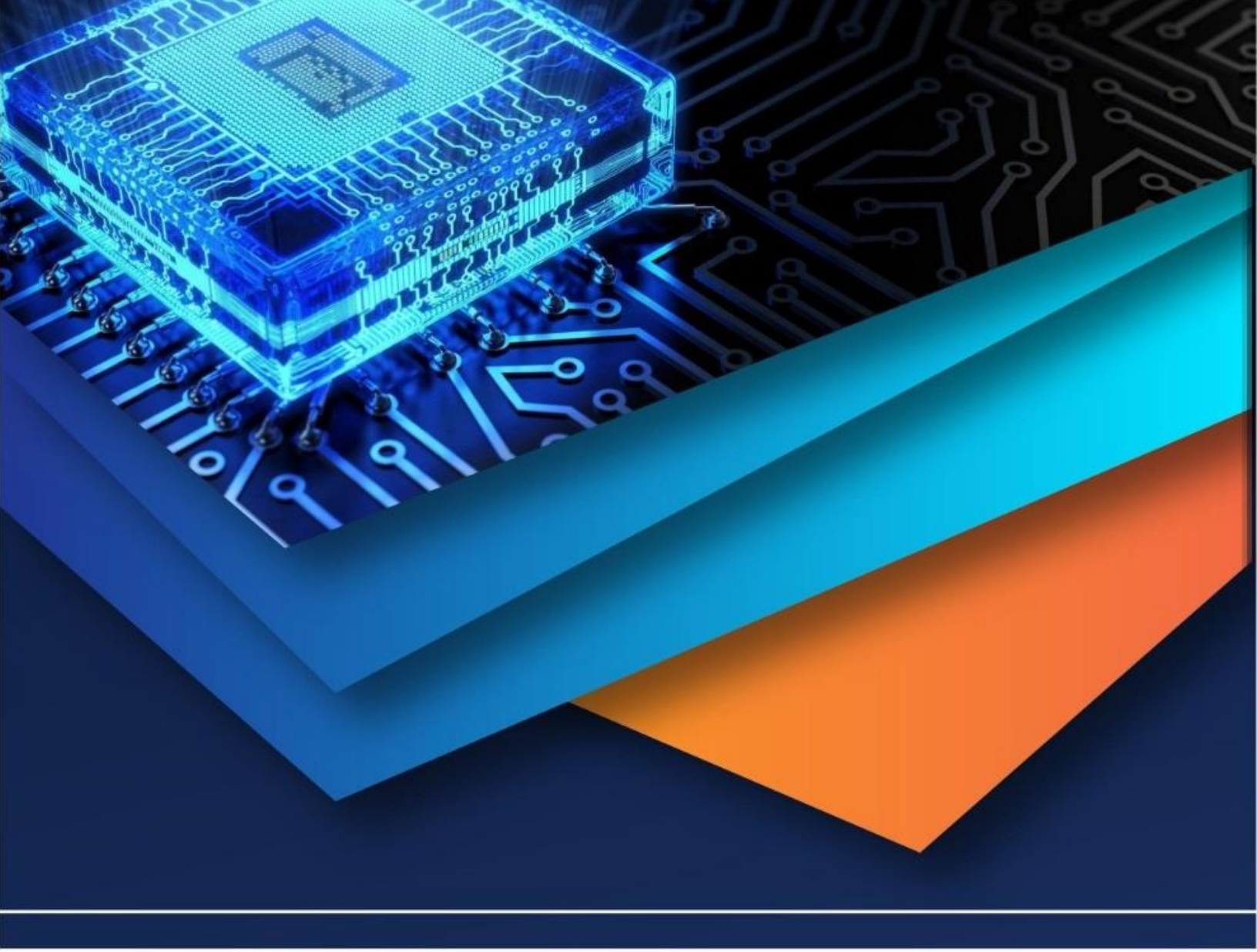
## VII. ACKNOWLEDGEMENT

This project and the research work behind it haven't been possible without the exceptional guidance of our project guide, Prof. Geeta Zaware. Her knowledge and enthusiasm kept us updated, and on tack by inspiring us with her work in research from the first day of work. And our Head of Department Prof. Chandrashekhar Mahajan who gave us an opportunity to work on this project. We like to express our gratitude towards the faculty of Vishwakarma Institutes of Technology who helped us, especially the librarian who time to time help us in finding precious research paper for our reference, allow us to use technologies useful for our work.
We are also grateful to all the researchers whose work help us in reaching to a dedicated work as it was because of their work in this field and their team and many of those who work in this field, also we are grateful to our project leader for his patience and trust on us, have always keep us motivated.
At-last we heartily appreciate our family members for their patience, financial and emotional support in our work.

## REFERENCES

[1] Y. Zhang, Y. Zhang, P. Qi, C. D. Manning, and C. P. Langlotz, “Biomedical and clinical English model packages for the Stanza Python NLP library,” Journal of the American Medical Informatics Association, vol. 28, no. 9, pp. 1892–1899, Sep. 2021, doi: https://doi.org/10.1093/jamia/ocab090
[2] D. Kartsaklis et al., “lambeq: An Efficient High-Level Python Library for Quantum NLP,” arXiv:2110.04236 [quant-ph], Oct. 2021, Available: https://arxiv.org/abs/2110.04236
[3] A. Petukhova and N. Fachada, “TextCL: A Python package for NLP preprocessing tasks,” SoftwareX, vol. 19, p. 101122, Jul. 2022, doi: https://doi.org/10.1016/j.softx.2022.101122.
[4] J. R. Jim, M. Apon, P. Malakar, M. M. Kabir, K. Nur, and M. F. Mridha, “Recent advancements and challenges of NLP-based sentiment analysis: A state-of-the-art review,” Natural Language Processing Journal, vol. 6, pp. 100059–100059, Feb. 2024, doi: https://doi.org/10.1016/j.nlp.2024.100059.
[5] L. Qin et al., “Large Language Models Meet NLP: A Survey,” arXiv.org, 2024. https://arxiv.org/abs/2405.12819
[6] Teja Reddy Gatla, “A Groundbreaking Research in Breaking Language Barriers: NLP And Linguistics Development,” International Journal of Advanced Research and Interdisciplinary Scientific Endeavours, vol. 1, no. 1, pp. 1–7, 2024, doi: https://doi.org/10.61359/11.2206-2401.
[7] S. Dai et al., “AI-based NLP section discusses the application and effect of bag-of-words models and TF-IDF in NLP tasks,” Deleted Journal, vol. 5, no. 1, pp. 13–21, Jun. 2024, doi: https://doi.org/10.60087/jaigs.v5i1.149.
[8] C. Thomson, E. Reiter, and A. Belz, “Common Flaws in Running Human Evaluation Experiments in NLP,” Computational Linguistics, vol. 50, no. 2, pp. 795–805, 2024, doi: https://doi.org/10.1162/coli_a_00508.
[9] C. Si, D. Yang, and T. Hashimoto, “Can LLMs Generate Novel Research Ideas? A Large-Scale Human Study with 100+ NLP Researchers,” arXiv.org, 2024. https://arxiv.org/abs/2409.04109
[10] F. M. Plaza-del-Arco, A. Curry, A. C. Curry, and D. Hovy, “Emotion Analysis in NLP: Trends, Gaps and Roadmap for Future Directions,” arXiv.org, 2024. https://arxiv.org/abs/2403.01222



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY