International Journal of Scientific Research in Computer Science, Engineering and Information Technology

ISSN : 2456-3307

OPEN ACCESS

Available Online at : www.ijsrcseit.com

doi : https://doi.org/10.32628/CSEIT2410242

The Comparative study of Python Libraries for Natural Language Processing (NLP)

Dr. Dhara Ashish Darji*, Dr. Sachinkumar Anandpal Goswami

Faculty of Computer Applications, Ganpat University, Ganpat Vidyanagar Mehsana-Gozaria, Highway, Kherva, Gujarat, India

ARTICLEINFO

Article History:

Accepted: 25 March 2024

Published: 16 April 2024

Publication Issue

Volume 10, Issue 2

March-April-2024

Page Number

499-512

ABSTRACT

Natural Language Processing (NLP) has seen significant advancements in recent years, driven largely by the availability of powerful Python libraries. This comparative study aims to analyze and compare the performance, language support, community support and ease of use of many popular Python libraries for NLP like NLTK (Natural Language Toolkit), spaCy, TextBlob, Flair, Jina, Gensim etc. The study evaluates these libraries across various NLP tasks such as tokenization, part-of-speech tagging, named entity recognition, sentiment analysis, and text summarization. Additionally, the paper discusses the strengths and weaknesses of each library, providing insights into their suitability for different NLP applications. Through detailed experimentation and analysis, this study aims to guide researchers and practitioners in selecting the most appropriate library for their NLP projects.

Keywords : NLP, Libraries, NLU, NLG, NLTK

I. INTRODUCTION

Understanding how computerized systems perceive and handle texts, spoken language, and related materials is known as natural language processing, or NLP. "The application of computational techniques for the analysis and generation of natural language and speech" is how the Oxford Dictionary defines natural language processing. These programs try to make language processing work like a person's for a variety of tasks by using computer methods to look at the texts they create. These apps help to make large-scale textual data analysis more controllable and provide the possibility of thorough linguistic studies. Although natural language processing (NLP) techniques were originally used to preserve endangered languages and save them from extinction, in recent years they have been used in many research projects to organize and understand large amounts of data. Working in a variety of industries these days, such as marketing, information validation, and information retrieval or visualization, without the assistance of natural language processing (NLP), has become much more difficult and time-consuming. [20] This study's main aims are to shed light on the most important stages of NLP applications and to do a full bibliometric analysis of the NLP literature in the field of library and information science (LIS). In this particular context, its

goals are to discern future trends and propose prospective topic areas that might profit from a variety of natural language processing techniques. The following are the main research questions: What are the main uses of NLP that the LIS literature demonstrates, and how have NLP subjects evolved historically? Which scholarly publications, and in what proportions across NLP subtopics, are suggested reading for inexperienced NLP researchers? Which articles in the NLP area are noteworthy for LIS? As such, the first part of this paper presents the basic elements of NLP applications. It then gathers and assesses the literature to reveal the fundamentals of this area.[21]

II. BACKGROUND STUDY OF NLP

Natural language processing (NLP) may be traced back to the years immediately after the conclusion of World War II, namely the 1940s. During this time period, there was a rising awareness of the relevance of the role that automation may play in the translation of languages from one language to another. The creation of a computer that could carry out translations of this kind on its own automatically was the goal. On the other hand, it quickly became clear that the difficulty level of this work was far higher than had been previously anticipated. [9]

By the year 1958, a number of researchers had already identified significant obstacles that impeded the development of NLP. One of these scholars, Noam Chomsky, in particular, noticed a puzzling characteristic of language models. He considered it troublesome that these models viewed sentences that were grammatically correct but nonsensical as equally irrelevant as phrases that were neither grammatically correct nor sensible. He also found it problematic that these models treated sentences that were neither grammatically correct nor sensible as equally relevant. For example, the phrase "Colorless green ideas sleep furiously" was regarded as having the same level of plausibility as the line "Furiously sleep ideas green colorless." These models did not discriminate between them, despite the fact that the first option is accurate according to grammar standards and the second option is incorrect. Chomsky proposed that computers should be able to distinguish between linguistic categories in the same way that humans can. [17]

During the same period of time, between the years 1957 and 1970, researchers working in the field of natural language processing split off into two primary camps: symbolic and stochastic. Researchers who worked with symbolic systems, often known as rule-based systems, focused their attention on formal languages and the production of syntax. This group, which included a significant number of computer scientists and linguists, is generally considered to be the origin of research in artificial intelligence. [16]

After 1970, researchers in natural language processing (NLP) began to broaden the scope of their interests and methods in response to the rapidly developing technology and increasing body of knowledge. Logic-based paradigms, which focused on storing language rules in mathematical logics, emerged as an innovative new field during this time period. After some time, this aspect of NLP research ended up being essential in the creation of the computer language known as Prolog. [7]

Empiricism and probabilistic models came to play an increasingly important role in natural language processing (NLP) research between the years 1983 and 1993. During the 1950s and 1960s, researchers started to question the validity of the arguments that Chomsky and others had presented. As a result of this research, it became clear that a significant number of the arguments that were presented in theoretical books seemed compelling; nonetheless, they did not correlate with actual facts. [18]

As a consequence of this, by the year 1993, probabilistic and statistical approaches had become the preferred models for tackling issues associated with natural language processing. Due to the large amounts of data that are accessible on the Internet, Natural Language Processing (NLP) has also experienced a shift towards information extraction and production during the last decade. [29] Additionally, as personal computers have become more accessible to the general public, applications for natural language processing (NLP) have grown more widespread, which has stimulated additional study in the subject. [8]

III.OVERVIEW OF NLP

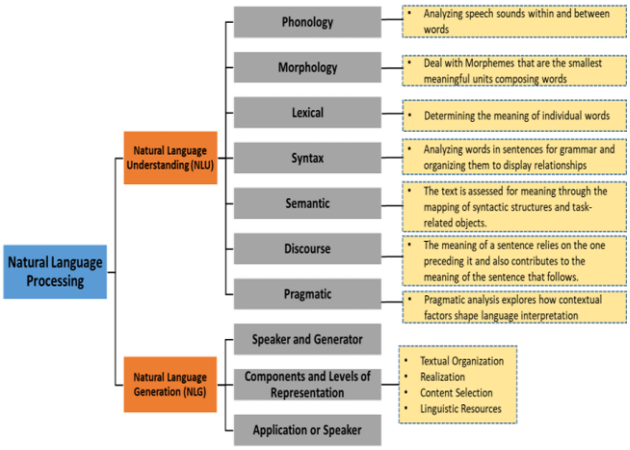
"Natural Language Processing (NLP) is an area of computing technology, specifically within the domain of artificial cognition (AI), that is devoted to providing computers with the capacity to grasp written text and spoken language in a manner that is comparable to how human beings do so. Natural language processing (NLP) is built on the basic rules of linguistic computation, which include modeling human language based on rules. Statistical, machine learning, and deep learning models are also used. The combination of these technologies gives computers the ability to interpret human language, whether it is in the form of written text or spoken words, and to completely comprehend its meaning, which includes the goals and emotions of the person who is speaking or writing it. NLP is the driving force behind a variety of computer systems that can translate text from one language to another, react to voice requests, and quickly summarize huge amounts of text, frequently in real time. [22] It is quite possible that you have interacted with natural language processing (NLP) in some form or another when using common applications such as digital assistants, voice-activated GPS systems, speech-to-text transcription tools, customer care chatbots, and other user-friendly features. In addition, natural language processing is rapidly being used in workplace solutions that aim to improve operational effectiveness, raise staff productivity, and simplify essential business procedures." [4]

IV. COMPONENTS OF NLP

Natural Language Processing (NLP) may be broken down into its two basic components, Natural Language Understanding (NLU) and Natural Language Generation (NLG), which respectively refer to the comprehension and production of written material. [23]The broad categories that NLP falls under are shown in Figure 1. Both Natural Language Understanding (Linguistic) (NLU) and Natural Language Generation (NLG) are topics that will be investigated in the next section. [3]

FIGURE- I

BROAD CLASSIFICATION OF NLP



A. NLU

The term "natural language understanding" (NLU) refers to the ability of computers to understand and interpret human language. This is accomplished by isolating a number of components, including ideas, entities, feelings, and keywords, among others. Practical uses of NLU may be seen in customer service solutions, where it helps with the interpretation of client concerns, whether they are communicated vocally or in written form. Linguistics is a scientific field that investigates the many meanings of language, the context in which language is used, and the various

structural components of language. Because of this, it is very necessary to have a solid understanding of the fundamental concepts related to natural language processing (NLP) as well as the several levels that fall within NLP. In the following sections, we will look into some of the vocabulary that is often used within the many levels that make up NLP. [2]

B. NLG

The term "natural language generation" (also known as "NLG") refers to the process of producing meaningful phrases, sentences, and paragraphs based on an internal representation. It is an essential part of natural language processing and is carried out through a four-step process. The first step involves determining the goals to be accomplished; the second step involves formulating a plan for how these goals can be accomplished by analyzing the current environment and the sources of communication that are available; and the third and final step involves translating these plans into text that is comprehensible. Figure 2 illustrates this process. In contrast to the method of comprehending language, NLG refers to the practice of using non-linguistic means. [2]

V. LIBRARIES FOR NLP

A. Hugging Face Transformers

Transformers provide users with access to a vast library of pre-trained models that are meant to perform a variety of tasks across a variety of modalities, such as text, visual material, and audio. the proper number. They are fluent in more than one hundred different languages. Transformers perform very well in image-related jobs like picture categorization, object identification, and segmentation. These are examples of visual tasks. They demonstrate their talents in the field of audio via activities such as voice recognition and audio categorization. Transformer models are notable for their versatility since they are able to tackle jobs that entail numerous modalities at the same time. These tasks include answering questions linked to visual material as well as table questions, optical character recognition, information extraction from scanned documents, video categorization, and more.

B. SpaCy

Natural language processing is the primary emphasis of the open-source SpaCy library, which is written in both Python and Cython. It is completely free to use and incorporates the latest academic research and technological advances into its design. SpaCy was designed from the very beginning to be used in production settings that are representative of the real world. This library comes complete with pre-trained pipelines and provides support for tokenization and training in more than 60 different languages at the present time. It can do things like tagging, parsing, identifying named entities, text categorization, and learning to do more than one thing at once using pre-trained models like BERT and combining the newest speed and neural network models. SpaCy delivers a potent training system that is ready for use in production, as well as making the process of packaging, deploying, and maintaining models and workflows easier. It is essential to point out that SpaCy is a piece of open-source software that may be used for commercial purposes and is released under the MIT license. [7]

C. Fairseq

As a framework for sequence modeling, Fairseq gives researchers and developers the ability to build bespoke models for a variety of text-generating initiatives, including translation, summarization, language modeling, and many more. These models may be used for tasks such as translation, summarization, and modeling. In addition to this, it provides examples of real-world applications that may be used as implementations for a variety of sequence modeling research articles. [5]

D. Jina

Jina is a state-of-the-art framework for neural search that enables the rapid construction of high-performance and scalable neural search applications. Using Jina, it is simple to develop solutions for problems such as indexing, querying, and understanding multi-modal or cross-modal data. These problems might include the processing of data in a variety of forms, including video, photos, text, audio, source code, and PDFs. [24]

E. Gensim

Gensim is a Python package that was developed for activities such as the modeling of topics, the indexing of documents, and the retrieval of similarities from large text collections. Professionals in the domains of natural language processing (NLP) and information retrieval (IR) are the target audience for this publication. In particular, Gensim stands out due to the efficacy of its parallel processing, which provides implementations of well-known algorithms that are both efficient and effective. [31] Some of the methods in this group are online Latent Semantic Analysis (LSA/LSI/SVD), Latent Dirichlet Allocation (LDA), Random Projections (RP), Hierarchical Dirichlet Process (HDP), and word2vec, which is a deep learning method.

F. Flair

Your text data may be analyzed using cutting-edge NLP models thanks to Flair, which is a powerful and sophisticated natural language processing (NLP) library. It includes a broad variety of skills, such as named entity recognition (NER), part-of-speech tagging (PoS), specific features for managing biological information, sense disambiguation, and classification. [32] These are only a few examples of the capabilities it entails. In addition to this, it provides support for a multitude of languages, and this number is always growing. The user-friendliness of Flair's interfaces, which make it easier to employ a variety of word and document embeddings in conjunction with one another, is one of the software's defining characteristics. The Flair embedding, the BERT embedding, and the ELMo embedding are some examples of these. PyTorch is incorporated easily within the framework, which makes the process of training your own custom models and experimenting with creative methods employing Flair embeddings and related classes much simpler. [26]

G. Allen NLP

Allen NLP is a natural language processing (NLP) research library that was built using PyTorch and is distributed under the Apache 2.0 license. Its goal is to make it easier to construct cutting-edge, deep learning models that can be used for a wide variety of language activities. You will discover a wide variety of already-implemented models scattered throughout its structure. These models are already in use. These implementations are not only rigorously constructed but also painstakingly documented, providing a solid platform for other research initiatives to build upon. The high-level configuration language that Allen NLP uses makes it easier to put a variety of common NLP techniques into practice. This is one of its most notable characteristics. [34] This includes the capability to carry out studies with transformers, participate in multitasking training, handle activities requiring both visual and language, solve concerns about fairness and interpretability, and a great deal more. Researchers are able to experiment with a diverse range of tasks thanks to the use of this configuration language. This enables them to concentrate their efforts on the key issues pertaining to their study rather than the complexities of putting that research into practice.

H. NLTK

The Natural Language Toolkit, sometimes known as NLTK for its abbreviated form, is a collection of open-source Python modules, datasets, and instructional materials that make it easier to conduct research and make improvements in the field of natural language processing. This all-inclusive toolkit provides friendly

user interfaces for accessing a vast variety of resources, such as more than 50 lexical databases and corpora, such as WordNet. In addition to these resources, NLTK offers a collection of text processing libraries that may be used for a variety of tasks, including classification, tokenization, stemming, tagging, parsing, and semantic analysis. In addition to this, it provides APIs that may be used to combine sophisticated NLP libraries that are commonly used in the industry. [24]

I. Core NLP

The Stanford Core NLP project provides users with a selection of Java-based tools for the examination of natural language. It can read raw human-written text and give important details like word roots and the parts of speech they belong to, identifying entities like company or person names, standardizing and understanding dates, times, and numbers, putting together sentences using phrases and the relationships between words, and finding noun phrases that refer to the same entity. [19]

J. Pattern

Web mining is the purpose for which Pattern was developed as a Python package. You will have access to a variety of data mining tools if you use them. Some of these tools include a web crawler, an HTML DOM parser, and access to online services such as Google, Twitter, and Wikipedia. In addition to this, Pattern integrates a variety of models that are associated with natural language processing. These models include part-of-speech tagging, n-gram search, sentiment analysis, and integration with WordNet. Pattern is able to construct machine learning models in addition to natural language processing. Some examples of these models are the vector space model, clustering, and classification via the use of algorithms such as KNN, SVM, and Perceptron. In addition to this, it provides tools for the measurement and display of graph centrality, which extends the value of the product to network analysis. [1]

K. Text Blob

TextBlob is a Python module that has been developed specifically for the processing of data that is based on text. It offers a straightforward user interface that makes it easier to participate in common natural language processing activities, such as part-of-speech tagging, the extraction of noun phrases, sentiment analysis, classification, translation, and a variety of other jobs that are quite comparable. This program, TextBlob, makes good use of what NLTK and Pattern can do. It builds on the many resources that both libraries provide and works well with them all together. [6]

L. Hugging Face Tokenizers

This library offers an implementation of widely utilized tokenizers in the present day, prioritizing both efficiency and adaptability. [14]

M. Haystack

Haystack is a complete framework that enables the building of strong pipelines that are ready for deployment for a variety of search applications. Haystack offers you cutting-edge NLP models to enable novel search experiences, allowing users to express inquiries in a natural language manner. Whether your objective is to perform question-answering or to dig into semantic document retrieval, Haystack can help you achieve either of these goals. Notably, Haystack is organized in a modular fashion, which makes it easier to include leading-edge technologies derived from other open-source projects like Huggingface's Transformers, Elasticsearch, and Milvus. [25]

N. Snips NLU

Python's Snips NLU is a library that was developed specifically for the purpose of extracting structured data from sentences written in natural language. It is crucial to translate the user's words into a representation of their intended message that the machine can understand when a user interacts with an

AI system using natural language. This part of Snips called Natural Language Understanding (NLU) works by first figuring out the user's goal, which is called the "intent," and then getting the query's parameters, which are called "slots." [15]

O. NLP Architect

In Natural Language Processing and Natural Language Understanding, NLP Architect is an open-source Python library that has been specifically designed for research into cutting-edge deep learning architectures and techniques. Its goal is to improve the overall performance of neural networks in these areas. This flexible library was designed on purpose to provide flexibility and simplicity of growth, which enables the simple and speedy incorporation of NLP models into applications. Additionally, it acts as a platform for demonstrating models that have been optimized. [10]

P. PyTorch NLP

PyTorch-NLP is a set of essential tools that have been built for natural language processing using the PyTorch framework. It extends the capabilities of PyTorch by providing fundamental functionality for the processing of text data.

Q. Polyglot

Polyglot is a natural language processing technology that makes it possible to develop large applications in several languages. Over 196 languages can be detected, 165 languages can be tokenized, 40 languages can recognize named entities, 16 languages can tag parts of speech, 136 languages can do sentiment analysis, 137 languages can do word embeddings, 135 languages can do morphological analysis, and 69 languages can be transliterated. These are just some of the capabilities that are included.

R. Text Attack

TextAttack is a Python framework that was developed specifically for natural language processing (NLP) activities. It includes adversarial assaults, data augmentation, and model training. [11]

S. Word Forms

Word forms are able to generate an accurate representation of the whole range of possible variants for an English word. Conjugating verbs and changing single nouns into their plural forms are also included in this step. In addition to this, it is able to create links between other elements of speech, such as connecting nouns and adjectives, linking adjectives and adverbs, and linking nouns and verbs, among other possible connections. [27]

T. Rosetta

Rosetta is a framework that protects users' privacy and is built on top of TensorFlow. It interacts with standard technologies for protecting privacy while computing, including encryption, federated learning, and trusted execution environments. Rosetta takes advantage of TensorFlow's application programming interfaces (APIs) and enables the transfer of typical TensorFlow scripts in a way that protects privacy with just a small number of modifications required. [13]

U. Pandas

Python's Pandas module offers data structures that are quick, flexible, and elegant. These data structures are designed to make the management of "relational" or "labeled" data more straightforward. Its major goal is to fulfill the role of the central, high-level component for the purpose of carrying out pragmatic, real-world data analysis tasks in Python. In addition to this, its overarching goal is to become the open-source tool that is the most powerful, flexible, and adaptable in terms of data manipulation and analysis, regardless of language.

V. Scikit-learn

Scikit-learn is an open-source machine learning package for the Python programming language. It is also often referred to as sklearn. There are many techniques in this group that can be used for classification, regression, and clustering. These include k-means, DBSCAN, support vector machines, random

forests, gradient boosting, and communication, and these algorithms are used to analyze data in order to classify it, predict it, or cluster it. In addition to this, it is designed to work in concert with the Python numerical and scientific libraries known as NumPy and SciPy in an effortless manner. [12]

VI. COMPARATIVE STUDY OF VARIOUS NLP LIBRARIES

TABLE II

COMPARISON OF NLP LIBRARIES

Library	Strengths	Weaknesses	Performance	Language Support	Ease of Use	Community Support	Typical Applications
Hugging Face Transformers	Wide range of models, easy integration, pre-trained models	Complex API for beginners	High performance	Multilingual support	Moderate to Advanced	Large and Active Community	Various NLP tasks - text generation, translation, summarization
SpaCy	High performance, fast, production-ready, API ease	Limited language support	Very good	Multilingual support	Beginner to Intermediate	Active Community	NER, POS tagging, parsing, text classification
Fairseq	Emphasis on fairness, bias mitigation, seq2seq models	Limited high-level APIs	Depends on models used	Multilingual support	Moderate to Advanced	Moderate Community	Translation, sequence-to-sequence learning
Jina	Distributed indexing, cloud-native search framework	Complex for simple tasks	Scalable	Multilingual support	Intermediate to Advanced	Growing Community	Large-scale document search, distributed querying
Gensim	Topic modeling, efficient implementations of algorithms	Focuses mainly on specific NLP tasks	Depends on task	Multilingual support	Beginner to Intermediate	Moderate Community	Topic modeling, word vectorization
Flair	State-of-the-art embeddings, sequence labeling	Resource-intensive for large datasets	Moderate to good	Multilingual support	Intermediate	Active Community	Text classification, named entity recognition
AllenNLP	Built on PyTorch, deep learning for NLP tasks	Steeper learning curve	Depends on models used	Multilingual support	Intermediate to Advanced	Large and Active Community	Deep learning for NLP tasks

NLTK	Comprehensive toolkit, variety of tasks	Processing speed for large datasets	Varies with tasks	Multilingual support	Beginner to Intermediate	Active Community	Tokenization, stemming, parsing, sentiment analysis
CoreNLP	Various NLP tasks, named entity recognition	Requires Java runtime	Depends on tasks	Multilingual support	Intermediate	Moderate Community	NLP tasks - POS tagging, named entity recognition
Pattern	Web mining, data extraction from websites	Limited update and support	Varies with tasks	Limited language support	Beginner to Intermediate	Limited Community	Web mining, data extraction
TextBlob	Simplified interface, sentiment analysis	Limited robustness	Varies with tasks	Limited language support	Beginner	Moderate Community	Text processing, sentiment analysis
Hugging Face Tokenizers	Efficient tokenization library for	Limited standalone function	High performance	Multilingual support	Beginner to Intermediate	Large and Active Community	Tokenization for model inte
	NLP tasks	alities					gration
Haystack	End-to-end framework for large-scale search engines	Learning curve for complex tasks	Depends on tasks	Multilingual support	Intermediate to Advanced	Growing Community	Large-scale document search, question-answering
Snips NLU	Intent recognition, slot filling	Limited features	Varies with tasks	Multilingual support	Beginner to Intermediate	Limited Community	Intent recognition, slot filling
NLP Architect	Framework for complex NLP applications, end-to-end	Limited community and resources	Depends on tasks	Multilingual support	Intermediate to Advanced	Growing Community	Various NLP applications
PyTorch-NLP	PyTorch-based, high-level NLP toolkit	Limited resources and updates	Depends on models used	Multilingual support	Intermediate to Advanced	Moderate Community	Various NLP tasks - classification, tokenization, tagging
Polyglot	Multilingual supp	Limited to speci	Varies with tasks	Extensive lang	Intermediate	Moderate Com	Multilingual text

	ort, word vectors, morphological analysis	fic tasks		uage support		munity	processing, word vectors
TextAttack	Adversarial attacks, model evaluation for NLP models	Focused on specific tasks	Depends on tasks	Multilingual support	Intermediate	Growing Community	Model evaluation, adversarial attacks
Word Forms	Word form generation, morphological analysis	Limited in broader NLP tasks	Varies with tasks	Limited language support	Beginner to Intermediate	Limited Community	Word form generation, morphological analysis
Rosetta	Cross-lingual applications, language translation	Limited documentation, resources	Varies with tasks	Multilingual support	Intermediate	Limited Community	Cross-lingual applications, language translation
Pandas	Data manipulation, prepr	Not specialized for NLP tasks	Varies with tasks	English	Beginner to Intermediate	Large and Active Com	Data manipulation, preproce
	ocessing					munity	ssing for NLP tasks
Scikit-learn	General-purpose ML library for various tasks	NLP-specific features are limited	Depends on tasks	English	Beginner to Intermediate	Large and Active Community	ML tasks including basic NLP tasks

VII. CONCLUSION

Finally, the in-depth study of the classification, uses, and comparison of Python libraries created for Natural Language Processing (NLP) helps us understand how NLP tools and frameworks are constantly changing and becoming more unique. The Natural Language Processing (NLP) Library Taxonomy Project carried out the research. The following are some of the significant discoveries that have emerged from this research: Diverse Taxonomy for Improved Comprehension The taxonomy that was presented in this research acts as a structured framework that can be used to classify and understand the large variety of NLP libraries that are available inside the Python environment. It provides NLP specialists with assistance in navigating the complex environment. The remarkable versatility of Python Python's pre-eminence in the area of natural language processing (NLP) may be due to its remarkable versatility, user-friendliness, and the enormous support it garners from the developer community. Another reason for Python's prominence in the field of NLP is that it is widely used. It has earned its place as the most popular option for use in NLP applications, and justifiably so. Ubiquitous NLP applications: NLP libraries demonstrate a vast spectrum of applications that cover multiple domains, including capabilities such as text preparation, language modeling, sentiment analysis,

machine translation, and a great deal more. [28] This broad spectrum of applications may be found in NLP libraries. In order to successfully carry out activities such as the creation of chatbots, the retrieval of information, and the condensation of material, these technologies are very necessary. The Natural Language Toolkit (NLTK) continues to be an invaluable instructional resource for those who are just starting out in the field of natural language processing as well as teachers. It provides access to a complete collection of tools and resources, which makes it easier to learn about NLP principles and experiment with those concepts. The Industrial Capabilities of SpaCy: SpaCy stands out owing to its exceptional efficiency and speedy processing capabilities, which make it a perfect solution for large-scale NLP applications in industrial settings. Productivity is increased because of the inclusion of pre-trained models and support for a number of different languages. TensorFlow and PyTorch have become powerful players in the field of natural language processing (NLP), particularly for deep learning-related tasks. TensorFlow and PyTorch have become powerful players in the field of natural language processing (NLP). [30] Because of their malleability and dependable performance, they are the prime candidates for the development of complex NLP models. Exploring Word Embeddings and Topic Modeling with Gensim: Gensim continues to be the ideal library for word embeddings and topic modeling. It encompasses applications ranging from word2vec to doc2vec, and as a result, it provides vital insights into the representation of textual material. The Introduction of Transformer-Based Models Used in a New Era for Natural Language Processing The introduction of transformer-based models, which are freely available via libraries such as Hugging Face Transformers, ushered in a new era that was much more advanced for NLP. These models have repeatedly accomplished state-of-the-art outcomes across a wide variety of natural language processing (NLP) tasks. The selection of an NLP library should be in accordance with the project objectives. The selection of an NLP library should be in accordance with both the particular objectives of the NLP project and its requirements. Researchers, developers, and organizations all have a responsibility to carefully consider a number of issues, including operational efficiency, scalability, and the availability of community assistance. [33] Adapting to the Continuous Development of NLP Natural language processing (NLP) is a discipline that is always expanding and becoming more sophisticated as new libraries, models, and methods are developed. It is very necessary for practitioners to keep up with the most recent advancements in order to take advantage of the most cutting-edge tools and models, which will ultimately result in an improvement in the efficiency of natural language processing. In a nutshell, the richness and adaptability of Python's natural language processing (NLP) environment are two of its defining characteristics. [35] The decision about which NLP library to use should be made thoughtfully and in accordance with the particular goals of the project, regardless of whether those goals are focused on research and development or production. Due to the ever-changing nature of NLP, it is essential to maintain a state of continuous learning and flexibility in order to make use of the most recent tools and models for effective natural language processing.

VIII. ACKNOWLEDGMENTS

We are grateful to the academic community and researchers whose work laid the foundation for the study. The wealth of knowledge and insights from existing literature has been indispensable in framing the taxonomy and understanding the diverse applications of Python libraries in NLP. We would also like to acknowledge the developers and contributors of the Python libraries examined in this study. Their dedication to advancing NLP technology has provided researchers and practitioners with powerful tools and resources. This research would not have been possible without the collective efforts and support of these

individuals and communities. Thank you all for being an integral part of this academic journey.

REFERENCES

[1] A. Dunn, D. Inkpen and R. Andonie, "Context-Sensitive Visualization of Deep Learning Natural Language Processing Models," 2021 25th International Conference Information Visualisation (IV), Sydney, Australia, 2021, pp. 170-175, doi: 10.1109/IV53921.2021.00035.

[2] A. Ferrari, L. Zhao and W. Alhoshan, "NLP for Requirements Engineering: Tasks, Techniques, Tools, and Technologies," 2021 IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion), Madrid, ES, 2021, pp. 322-323, doi: 10.1109/ICSE-Companion52605.2021.00137.

[3] A. M. P. Braşoveanu and R. Andonie, "Visualizing Transformers for NLP: A Brief Survey," 2020 24th International Conference Information Visualisation (IV), Melbourne, Australia, 2020, pp. 270-279, doi: 10.1109/IV51561.2020.00051.

[4] Ahmed Banafa, "3 Natural Language Processing (NLP)," in Transformative AI: Responsible, Transparent, and Trustworthy AI Systems , River Publishers, 2024, pp.17-22.

[5] B. D. Bašić and M. P. di Buono, "An Analysis of Early Use of Deep Learning Terms in Natural Language Processing," 2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO), Opatija, Croatia, 2020, pp. 1125-1129, doi: 10.23919/MIPRO48935.2020.9245375.

[6] B. D. Shivahare, S. Ranjan, A. M. Rao, J. Balaji, D. Dattattrey and M. Arham, "Survey Paper: Study of Sentiment Analysis and Machine Translation using Natural Language Processing and its Applications," 2022 3rd International Conference on Intelligent Engineering and Management (ICIEM), London, United Kingdom, 2022, pp. 652-656, doi: 10.1109/ICIEM54221.2022.9853044.

[7] B. K. AlSaidi, S. K. AlMamari and F. H. Mohideen, "A survey on supervised and unsupervised NLP algorithms for mental health detection applications," 6th Smart Cities Symposium (SCS 2022), Hybrid Conference, Bahrain, 2022, pp. 163-167, doi: 10.1049/icp.2023.0390.

[8] B. K. AlSaidi, S. K. AlMamari and F. Hajamohideen, "A survey on mental health based on NLP," 6th Smart Cities Symposium (SCS 2022), Hybrid Conference, Bahrain, 2022, pp. 210-215, doi: 10.1049/icp.2023.0406.

[9] B. Rawat, A. S. Bist, U. Rahardja, Q. Aini and Y. P. Ayu Sanjaya, "Recent Deep Learning Based NLP Techniques for Chatbot Development: An Exhaustive Survey," 2022 10th International Conference on Cyber and IT Service Management (CITSM), Yogyakarta, Indonesia, 2022, pp. 1-4, doi: 10.1109/CITSM56380.2022.9935858.

[10] C. Anilkumar, A. Karrothu, N. S. Mouli and C. B. Tej, "Recognition and Processing of phishing Emails Using NLP: A Survey," 2023 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2023, pp. 1-4, doi: 10.1109/ICCCI56745.2023.10128481.

[11] D. W. Otter, J. R. Medina and J. K. Kalita, "A Survey of the Usages of Deep Learning for Natural Language Processing," in IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 2, pp. 604-624, Feb. 2021, doi: 10.1109/TNNLS.2020.2979670.

[12] Daniel Minoli; Benedict Occhiogrosso, "Current and Evolving Applications to Natural Language Processing," in AI Applications to Communications and Information Technologies: The Role of Ultra Deep Neural Networks , IEEE, 2024, pp.65-116, doi: 10.1002/9781394190034.ch2.

[13] E. Ceh-Varela and E. Imhmed, "Uncovering Water Research with Natural Language Processing," 2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC), Torino, Italy, 2023, pp. 983-984, doi: 10.1109/COMPSAC57700.2023.00138.

[14] F. B. Rodrigues, W. F. Giozza, R. de Oliveira Albuquerque and L. J. García Villalba, "Natural Language Processing Applied to Forensics Information Extraction With Transformers and Graph Visualization," in IEEE Transactions on Computational Social Systems, doi: 10.1109/TCSS.2022.3159677.

[15] I. J. Dristy, A. M. Saad and A. A. Rasel, "Mental Health Status Prediction Using ML Classifiers with NLP-Based Approaches," 2022 International Conference on Recent Progresses in Science, Engineering and Technology (ICRPSET), Rajshahi, Bangladesh, 2022, pp. 1-6, doi: 10.1109/ICRPSET57982.2022.10188544.

[16] K. Hood and P. K. Kuiper, "Improving Student Surveys with Natural Language Processing," 2018 Second IEEE International Conference on Robotic Computing (IRC), Laguna Hills, CA, USA, 2018, pp. 383-386, doi: 10.1109/IRC.2018.00079.

[17] K. Kanhaiya, Naveen, A. K. Sharma, K. Gautam and P. S. Rathore, "AI Enabled- Information Retrival Engine (AI-IRE) in Legal Services: An Expert-Annotated NLP for Legal Judgements," 2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS), Trichy, India, 2023, pp. 206-210, doi: 10.1109/ICAISS58487.2023.10250733.

[18] K. S, S. R, S. R and T. S V, "Survey on Automatic Text Summarization using NLP and Deep Learning," 2023 International Conference on Advances in Electronics, Communication, Computing and Intelligent Information Systems (ICAECIS), Bangalore, India, 2023, pp. 523-527, doi: 10.1109/ICAECIS58353.2023.10170660.

[19] Khurana, D., Koli, A., Khatter, K. et al. Natural language processing: state of the art, current trends and challenges. Multimed Tools Appl 82, 3713–3744 (2023). https://doi.org/10.1007/s11042-022-13428-4

[20] Kunal Sawarkar, Deep Learning with PyTorch Lightning: Swiftly build high-performance Artificial Intelligence (AI) models using Python , Packt Publishing, 2022.

[21] M. Ramprasath, K. Dhanasekaran, T. Karthick, R. Velumani and P. Sudhakaran, "An Extensive Study on Pretrained Models for Natural Language Processing Based on Transformers," 2022 International Conference on Electronics and Renewable Systems (ICEARS), Tuticorin, India, 2022, pp. 382-389, doi: 10.1109/ICEARS53579.2022.9752241.

[22] N. E. Houda Ouamane and H. Belhadef, "Deep Reinforcement Learning Applied to NLP: A Brief Survey," 2022 2nd International Conference on New Technologies of Information and Communication (NTIC), Mila, Algeria, 2022, pp. 1-5, doi: 10.1109/NTIC55069.2022.10100477.

[23] N. Zhang and J. Kim, "A Survey on Attention mechanism in NLP," 2023 International Conference on Electronics, Information, and Communication (ICEIC), Singapore, 2023, pp. 1-4, doi: 10.1109/ICEIC57457.2023.10049971.

[24] P. R. Kshirsagar, D. H. Reddy, M. Dhingra, D. Dhabliya and A. Gupta, "A Review on Application of Deep Learning in Natural Language Processing," 2022 5th International Conference on Contemporary Computing and Informatics (IC3I), Uttar Pradesh, India, 2022, pp. 1834-1840, doi: 10.1109/IC3I56241.2022.10073309.

[25] Pais, S., Cordeiro, J. & Jamil, M.L. NLP-based platform as a service: a brief review. J Big Data 9, 54 (2022). https://doi.org/10.1186/s40537-022-00603-5

[26] R. Boorugu and G. Ramesh, "A Survey on NLP based Text Summarization for Summarizing

Product Reviews," 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2020, pp. 352-356, doi: 10.1109/ICIRCA48905.2020.9183355.

[27] R. Guo and F. Ren, "Towards the relationship between Semantic Web and NLP," 2009 International Conference on Natural Language Processing and Knowledge Engineering, Dalian, China, 2009, pp. 1-8, doi: 10.1109/NLPKE.2009.5313806.

[28] R. Patil, S. Boit, V. Gudivada and J. Nandigam, "A Survey of Text Representation and Embedding Techniques in NLP," in IEEE Access, vol. 11, pp. 36120-36146, 2023, doi: 10.1109/ACCESS.2023.3266377.

[29] S. Cascianelli, G. Costante, A. Devo, T. A. Ciarfuglia, P. Valigi and M. L. Fravolini, "The Role of the Input in Natural Language Video Description," in IEEE Transactions on Multimedia, vol. 22, no. 1, pp. 271-283, Jan. 2020, doi: 10.1109/TMM.2019.2924598.

[30] S. T and S. S, "Survey On Next Word Prediction Techniques In Natural Languages," 2023 International Conference on Innovations in Engineering and Technology (ICIET), Muvattupuzha, India, 2023, pp. 1-6, doi: 10.1109/ICIET57285.2023.10220846.

[31] S. Yang, Z. Ning and Y. Wu, "NLP Based on Twitter Information: A Survey Report," 2020 2nd International Conference on Information Technology and Computer Application (ITCA), Guangzhou, China, 2020, pp. 620-625, doi: 10.1109/ITCA52113.2020.00135.

[32] Stančin, I., & Jović, A. (2019, May). An overview and comparison of free Python libraries for data mining and big data analysis. In 2019 42nd International convention on information and communication technology, electronics and microelectronics (MIPRO) (pp. 977-982). IEEE.

[33] T. S. N. Ayutthaya and K. Pasupa, "Thai Sentiment Analysis via Bidirectional LSTM-CNN Model with Embedding Vectors and Sentic Features," 2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP), Pattaya, Thailand, 2018, pp. 1-6, doi: 10.1109/iSAI-NLP.2018.8692836.

[34] Verspoor, K., Cohen, K.B. (2013). Natural Language Processing. In: Dubitzky, W., Wolkenhauer, O., Cho, KH., Yokota, H. (eds) Encyclopedia of Systems Biology. Springer, New York, NY. https://doi.org/10.1007/978-1-4419-9863-7_158

[35] Z. Shahbazi and Y. -C. Byun, "Blockchain-Based Event Detection and Trust Verification Using Natural Language Processing and Machine Learning," in IEEE Access, vol. 10, pp. 5790-5800, 2022, doi: 10.1109/ACCESS.2021.3139586.