





Article

Natural Language Processing in Knowledge-Based Support for Operator Assistance

Fatemeh Besharati Moghaddam ^{1,2,*} , Angel J. Lopez ^{1,2} , Stijn De Vuyst ^{1,2}  and Sidharta Gautama ^{1,2} 

¹ Department of Industrial Systems Engineering and Product Design, Ghent University, 9000 Ghent, Belgium; angel.lopez@ugent.be (A.J.L.); stijn.devuyt@ugent.be (S.D.V.); sidharta.gautama@ugent.be (S.G.)

² FlandersMake@UGent, Corelab ISyE, 9000 Ghent, Belgium

* Correspondence: fatemeh.besharatimoghaddam@ugent.be

Abstract: Manufacturing industry faces increasing complexity in the performance of assembly tasks due to escalating demand for complex products with a greater number of variations. Operators require robust assistance systems to enhance productivity, efficiency, and safety. However, existing support services often fall short when operators encounter unstructured open questions and incomplete sentences due to primarily relying on procedural digital work instructions. This draws attention to the need for practical application of natural language processing (NLP) techniques. This study addresses these challenges by introducing a domain-specific dataset tailored to assembly tasks, capturing unique language patterns and linguistic characteristics. We explore strategies to process declarative and imperative sentences, including incomplete ones, effectively. Thorough evaluation of three pre-trained NLP libraries—NLTK, SPACY, and Stanford—is performed to assess their effectiveness in handling assembly-related concepts and ability to address the domain's distinctive challenges. Our findings demonstrate the efficient performance of these open-source NLP libraries in accurately handling assembly-related concepts. By providing valuable insights, our research contributes to developing intelligent operator assistance systems, bridging the gap between NLP techniques and the assembly domain within manufacturing industry.

Keywords: natural language processing; NLP; part of speech; POS tagging; closed domain; operator support; assembly instructions; NLTK; SPACY; Stanford; benchmark



Citation: Besharati Moghaddam, F.; Lopez, A.J.; De Vuyst, S.; Gautama, S. Natural Language Processing in Knowledge-Based Support for Operator Assistance. *Appl. Sci.* **2024**, *14*, 2766. <https://doi.org/10.3390/app14072766>

Academic Editor: José Machado

Received: 30 January 2024

Revised: 26 February 2024

Accepted: 20 March 2024

Published: 26 March 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, manufacturing companies have faced escalating demand for complex products with a greater number of variations [1]. This surge in product complexity has been driven by the advent of Industry 4.0 technologies, which encompass a range of advancements, such as the Internet of Things (IoT), artificial intelligence (AI), robotics, and data analytics [2]. These technologies have revolutionized manufacturing processes, creating highly customized products and flexible production lines. Consequently, assembly operators have had to adapt their skills to accommodate a wider range of tasks, often involving intricate assembly procedures and rapid product changes [3].

The development of Industry 4.0 technologies has also resulted in operators needing to learn new skills to cope with the complexity of different products [4]. Automation and digitization have transformed traditional assembly tasks, requiring operators to interact with advanced machinery, robotic systems, and digital interfaces. As a result, the role of assembly operators has evolved from executing manual tasks to managing sophisticated production systems. In this context, the significance of operator assistance and smart assistance technologies has been steadily increasing [5]. These technologies are designed to enhance worker productivity, efficiency, and safety by minimizing errors, automating repetitive tasks, and offering real-time feedback and guidance [6]. The integration of smart technologies into assembly processes is essential for addressing the evolving needs of modern manufacturing. Industry 4.0 principles emphasize the interconnectedness of physical

systems with digital technologies, enabling seamless communication and collaboration between humans and machines. Smart assistance systems leverage these principles to provide operators real-time support, context-aware guidance, and predictive maintenance insights [7]. These systems optimize production workflows, reduce downtime, and improve operational efficiency by harnessing data analytics and AI. Therefore, addressing the challenges in operator assistance systems is crucial for the manufacturing industry to leverage the full potential of Industry 4.0 and to remain competitive in the global marketplace.

While existing support services for assembly operators provide procedural digital work instructions, there are notable limitations. Operators must often rely on their own knowledge or seek help from external experts when faced with unexpected challenges in the assembly process. For example, if a unique problem arises during an assembly task that deviates from the standard instructions, the current systems may not provide adequate guidance. This reliance on human expertise can lead to delays and reduced efficiency on the factory floor. The digitization of information presents an opportunity to address these limitations by gathering data from IT systems and providing more comprehensive support to operators [8]. To bridge this gap, adaptive assistance systems have emerged as a promising solution in manufacturing contexts. These systems encompass a range of technologies, including head-mounted displays [9], augmented reality [10,11], tangible user interfaces [12], and motion recognition [13]. They aim to provide real-time, context-aware support to operators, adapting to the specific needs of each assembly task and helping operators overcome unexpected challenges more effectively.

Most existing support systems in the assembly domain are designed around well-defined, pre-established questions and answers. These systems are effective when there is a direct mapping between a question posed by an operator and a known answer, and this mapping has been pre-defined within the system. However, a significant challenge arises when operators ask open-ended questions for which no predefined semantic links or specific answers exist. Moreover, these open questions are often formulated incompletely, making it even more challenging to interpret their correct semantic meaning [14]. Consider a scenario where an operator asks a question like, "How do I fix this?". The system may not have a predefined response because it lacks a direct match with any known question in its database. Furthermore, the question's vagueness and incompleteness pose additional hurdles to understanding the operator's intent. In these situations, an advanced solution is required. This is where natural language processing (NLP) technology comes into play [15]. NLP empowers assistance systems to comprehend and respond to open questions posed by operators by translating text strings into formal, semantic representations. NLP, in essence, offers a computerized approach to process and understand human language. It serves as a bridge between human language and computer systems, enabling the effective analysis of text or speech [16]. NLP research covers both open and closed domains [17]. NLP projects in both domains are areas of active research and have been extensively studied [18]. Open-domain NLP has received significant attention over the years, focusing on understanding and processing general human language, without specific constraints or limitations to a particular domain. Closed domains are more specialized in different industries and domains where the language and concepts are more specialized. The assembly domain in the assembly environment can be categorized as the closed domain in NLP, which poses a challenge for several reasons. First, to the best of our knowledge, there is limited existing research on the application of NLP techniques in the manufacturing industry, particularly in the area of assembly [19,20]. It means that there is a lack of publicly available documents and specific corpora related to NLP in the assembly.

Secondly, informal writing and poor grammar in procedures like quality reports further add to the challenge. While ChatGPT [21] and other large language models demonstrate proficiency at generating human-like text across diverse fields, their suitability for manufacturing assembly line support requires careful assessment. While these models can offer detailed responses to a wide range of questions, the critical demand for precision in manufacturing means that a single precise answer is often more valuable than extensive

information. When the operator interacts with the assistance system, the initial request can be in the form of either textual input in a conversational assistance system or spoken dialogue processed by a speech-to-text system. However, incomplete sentences may occur due to issues such as grammar, spelling, and speech variations, including different accents, dialects, and speaking styles. The assistance system must be capable of handling such incomplete requests effectively.

In this research, our objective is to address the challenges associated with NLP in the assembly domain. To tackle the lack of specific corpora related to NLP in assembly, as the first challenge, we aim to design and develop a domain-specific dataset with the dedicated target language ‘English’ that serves as a benchmark for close-domain analysis. This dataset will capture the unique language patterns, vocabulary, and linguistic characteristics specific to assembly-related tasks. By creating such a dataset, our aim is to provide a valuable resource for future research and contribute to the expansion of NLP applications in the assembly domain. Moreover, we recognize the need for handling incomplete sentences as the second challenge. To address this challenge, our research involves exploring two types of sentences in our designed dataset: declarative and imperative. We consider how both complete and incomplete sentences can be effectively processed within the realm of NLP, employing techniques such as part-of-speech analysis. This exploration is fundamental to bridging the gap between NLP techniques and the assembly domain within manufacturing industry. Furthermore, as part of our work, we have rigorously evaluated three prominent pre-trained NLP libraries: NLTK, SPACY, and Stanford. We have undertaken this evaluation as a means to assess and validate the effectiveness of these libraries in handling assembly-related concepts and addressing the unique challenges presented by the assembly domain. This validation, in turn, informs our broader contribution to the development of intelligent and efficient assembly processes.

2. Literature Review

Benchmarks are used to determine the top-performing system in a specific domain. According to Bowman [22], a suitable benchmark should meet certain requirements, including reliable assessment of language aspects, consistent annotated data, statistical power, and discouragement of biased models. A good NLP benchmark should have diverse tasks that reflect language understanding, high-quality labeled data, replicability, promotion of robust and generalizable models, regular updates, accessibility, and wide adoption [23–25].

In the context of building a benchmark, tasks specifically refer to the linguistic challenges that NLP models are designed to address and evaluate. These tasks encompass various aspects of language understanding, including, but not limited to, text classification, question answering, sentiment analysis, and machine translation [15]. Each of these tasks comes with its dedicated dataset and evaluation metrics that serve to measure the performance of NLP models across a wide range of language understanding challenges. These metrics can include accuracy, precision, recall, F1-score, or other task-specific evaluation criteria, providing a standardized way to compare and rank different models based on their performance [26]. To clarify, a task in this context represents a specific NLP problem that a benchmark aims to assess, such as classifying the sentiment of a text or translating one language to another.

State of the Art

There are two main categories of NLP benchmarks: open-domain and closed-domain. Open-domain benchmarks refer to benchmarks that cover a wide range of topics and domains and are designed to evaluate the general language understanding abilities of NLP models, e.g., GLUE (General Language Understanding Evaluation) [25] and Super-GLUE [27]. For building the benchmark in the open domain, various sources can be used, such as Wikipedia or Google articles [24,28], or the collection of multiple-choice Q/A tasks in different books, articles, and online forums [29].

In the open domain, some benchmarks utilize knowledge bases as a source of information, e.g., QALD [30], WebQuestion [31], and SimpleQuestion [32]. The goal of a knowledge-based benchmark is to evaluate a model's ability to reason and utilize structured knowledge. This type of benchmark typically involves tasks that require a model to answer questions or complete tasks based on a given knowledge base or graph [33]. The aim is to assess the model's ability to retrieve, understand, and utilize information from the knowledge base. Others, e.g., GLUE [25], SuperGLUE [27], and SQUAD [24], do not necessarily require knowledge bases. The goal of a non-knowledge-based benchmark is to evaluate a model's ability to understand language and perform general language tasks without relying on external knowledge sources. This type of benchmark typically involves tasks such as text classification, sentiment analysis, and language modeling. The aim is to assess the model's ability to generalize and make sense of natural language text without access to a specific knowledge base.

Closed-domain benchmarks, on the other hand, are designed to evaluate NLP models in specific domains or tasks. These benchmarks typically have a narrower focus than open-domain benchmarks and are designed to evaluate the ability of NLP models to perform specific tasks in particular domains. There are some benchmarks available in the medical domain (MiPACQ clinical QA benchmark [34], MIMIC benchmark [35], i2b2/VA benchmarks [36]), and a bankruptcy dataset in the finance domain [37]. The purpose of creating benchmarks in the close domain of NLP is to foster innovation, improve the understanding of domain-specific language processing, and drive the development of intelligent and efficient solutions that cater to the requirements of specific industries, domains, or specialized fields.

For the training data, in some research, authors have used the manually trained corpus in the open domain [38,39] or the closed domain [40,41] for their investigation. Kumar [42] proposed an approach in part-of-speech (POS) tagging for the open domain, considering their defined corpus with 77,860 tokens for training and 7544 for testing. In a similar study in the open domain, 14,369 tokens in the training set and 5000 tokens in the testing set were considered [43]. Rezai [44] offered a POS tagger corpus with 5,000,000 tokens in the open domain for training and 11,766 tokens in the test set for the Persian language. However, using the manually annotated corpus, the corpus size may not be enough for modeling and efficient evaluation [16,45].

To address the challenge of corpus size, particularly in the closed domain, one solution can be to employ open-source NLP libraries to train their methodologies and evaluate their datasets. Open-source libraries in NLP can be traditional libraries, e.g., Stanford NLP Suite [46], Google SyntaxNet [47], NLTK [48], and SpaCy [49], or the deep learning-based models (transfer-based) libraries, e.g., BERT [50] and DistilBERT [51]. Many types of research analyzing NLP tasks consider both traditional and deep learning pre-trained libraries and compare them together in different aspects [52,53]. The transformer-based technique performs efficiently in entity recognition, information extraction, and semantic analysis [54–57]. However, in the preliminary step of pre-processing, particularly part-of-speech (POS) tagging, traditional NLP techniques demonstrate notable efficiency [58,59]. In [60], the BERT method was used for sentiment analysis while considering information extraction and named entity recognition. However, for the POS tagging, they used the methods of the SpaCy and NLTK libraries to perform the analyses. The reason for choosing these libraries, based on [60], was the ability to make predictions efficiently about which tag or label most likely applied in this context. In the study by Omran [61], four distinct traditional open-source libraries were utilized to train their methodology. Subsequently, they manually annotated 1116 tokens with the correct part-of-speech tags and assessed the tagging accuracy using each of these open-source libraries.

The method for collecting sentences in a benchmark varies depending on the benchmark and the task it aims to evaluate. It should be carefully designed to ensure that the benchmark is both diverse and representative of the task it aims to evaluate. In some cases,

the sentences are collected from existing datasets or corpora, while in other cases, these may be created specifically for the benchmark.

Wang [25] collected the sentences from a variety of sources, including news articles, social media posts, and product reviews, among others, in order to build the GLUE benchmark. The sentences were then labeled according to specific tasks, such as natural language inference, sentiment analysis, and textual entailment. Dietz [62] provided a benchmark named “Wikimarks” for query-specific clustering, entity-linking, and entity-retrieval. Their methods of picking sentences from Wikipedia can cover a wide range of topics of general interest. In the case of the Penn Treebank benchmark [63,64], the sentences were drawn from a large corpus of Wall Street Journal articles, which were manually annotated with part-of-speech tags and syntactic parse trees.

In some cases, the sentences may be collected through crowdsourcing or other methods, such as in the case of the WebQuestions dataset [31], which was created by collecting natural language questions from the Internet and then annotating them with corresponding answers. Another example of a benchmark where the sentences were not collected through corpora or datasets is the Winograd Schema Challenge [65]. The Winograd Schema Challenge consists of a set of multiple-choice questions that are designed to test a machine’s understanding of natural language. The questions are based on a specific type of pronoun resolution problem known as a Winograd Schema, which requires the machine to understand the meaning of a sentence in order to correctly identify the referent of a pronoun. The questions for the challenge were created by the organizers.

Various methods for sentence selection in both open and closed domains are discussed in Table 1, taking into account the number of categories considered and the objective of constructing the benchmark dataset.

Table 1. Techniques and purposes for sentence selection in open and closed domains of NLP benchmark.

Reference	Domain	Sentence Picking Techniques	Number of Classes	Purpose
[28]	Open	Queried Google with the OR-linked keywords	One class related to requirements engineering	Oriented to enable replication of NLP experiments and generalization of results in requirements engineering.
[29]	Open	Building three benchmarks considering rating numbers in AMAZON, extracting sentence pairs from Japanese datasets, and randomly picking from Wikipedia	-	Having a benchmark in the Japanese language considering cultural/social parameters without using the translation methods in order to evaluate NLU ability in the general domain
[66]	Close	In three domains, medicine, technology, and finance, they pick the sentences in Wikipedia based on the defined categories	Depend on the domain in Wikipedia categories	Fill the gap of the lack of pre-trained domain-specific models for languages other than English
[67]	Close	Build a benchmark based on one specific available corpus	14 subcategories based on the definition of events and entity, with not equal amounts for each	Compare two approaches on a newly built benchmark
[68]	Close	The benchmark is built on 6 existing tasks available in the state-of-the-art	6 available subtasks in medical	To support downstream applications in computerized diagnostic decision support and improve the efficiency and accuracy of healthcare providers during patient care.

Two factors, the size and composition of a benchmark, are important for its success. Diverse, representative, and accurately labeled sentences are crucial. The number of sentences needed depends on the task complexity and data variability. In the case of an

open domain benchmark, where the goal is to test a model's ability to perform a wide range of tasks, it may be important to have a large number of sentences to ensure adequate coverage of different domains and topics. The GLUE benchmark [25] consists of a collection of nine different datasets each with its considered task. The number of sentences is between 780 for 755k to this benchmark. The SQuAD benchmark [24] consists of 100,000 question-answer pairs, and [28] build a benchmark in the open domain with 79 documents as 34,268 sentences with 865,551 tokens.

However, in a closed domain benchmark, where the focus is on a specific task within a domain, a smaller number of sentences may be sufficient. It is important to note that while extensive datasets are vital for open-domain benchmarks to encompass diverse domains and topics, the process of creating and curating large datasets, especially in closed domains, can be inherently challenging and resource-intensive. This is exemplified by previous research studies that have successfully utilized limited sets of sentences and tokens for specific domains. For instance, a study examining the cooking domain in a question-answering context employed 2175 questions [69]. In the email domain for spam/thread detection, a dataset consisting of 350 paired questions was utilized for analysis [70]. Similarly, a study in the university field domain employed a corpus of 2903 sentences [71], while another study focusing on the oil industry utilized an 18,000 token dataset [72]. These examples highlight that even with a relatively smaller number of sentences or tokens, a closed domain benchmark can yield valuable insights and enable targeted analysis specific to the domain. Nonetheless, it is crucial for the selected sentences to encompass various aspects of the specific domain to ensure comprehensive coverage during evaluation.

In addressing the challenge of incomplete sentences within the realm of NLP, various methodologies have been proposed to infer missing information and enhance comprehension effectively. One such approach, hole semantics [73], has garnered attention for its ability to interpret incomplete sentences by identifying and filling gaps in the semantic structure. Recent studies have explored the application of techniques such as language modeling, contextual understanding, and syntactic analysis to address this issue [74,75].

To the best of our understanding, there exists a research gap in the field of NLP specifically in the context of assembly operations. This gap pertains to the lack of a benchmark dataset tailored to the assembly domain, which is essential for achieving high-performance NLP models. Furthermore, it remains unclear whether NLP techniques can effectively handle non-complete sentences commonly encountered in the assembly domain. The objective of our study is to bridge this gap. In Section 3, we outline the techniques employed to construct the benchmark dataset for the assembly domain and elaborate on the evaluation methods employed. The experimental findings are presented and analyzed in Section 4. Lastly, Section 5 presents the conclusions drawn from our research and outlines potential avenues for future work.

3. Methodology

In the assembly domain, NLP can be used to improve various aspects of operations, such as quality control, supply chain management, and customer service. Based on the general requirement for building a benchmark in NLP [15,76], a benchmark in the assembly domain for manufacturing would require consideration of the following factors:

- **Domain-specific language:** The manufacturing industry has its own language and terminology that people outside the industry may not be familiar with. A benchmark for NLP in manufacturing would need to use language that is specific to the domain.
- **Task-specific questions:** The benchmark should be designed to evaluate the performance of NLP models on tasks that are relevant to the manufacturing industry, such as defect detection, predictive maintenance, and inventory management. The questions should be carefully crafted to test the ability of NLP models to understand and process information related to these tasks.
- **Data quality:** The quality of the data used to create the benchmark is critical to its success. The data should be representative of the types of language and tasks that are

encountered in the manufacturing industry, and it should be of high quality to ensure that the benchmark results are accurate and reliable.

- **Evaluation metrics:** The benchmark should define clear evaluation metrics that can be used to measure the performance of NLP models on task-specific questions. The metrics should be relevant to the manufacturing industry and should provide a meaningful assessment of the model's ability to perform the task.
- **Ethical considerations:** The manufacturing industry deals with sensitive information, such as product designs and trade secrets. Therefore, the benchmark should take into account ethical considerations, such as intellectual property and confidentiality, to ensure that the data used in the benchmark are handled appropriately.

Overall, building a benchmark for NLP in assembly (as shown in Figure 1) would require careful consideration of the unique language, tasks, and data encountered in the industry, as well as the ethical considerations that come with handling sensitive information. The benchmark should be designed to evaluate the performance of NLP models on tasks that are relevant to the assembly domain in the manufacturing industry and provide a meaningful assessment of the model's ability to perform these tasks.

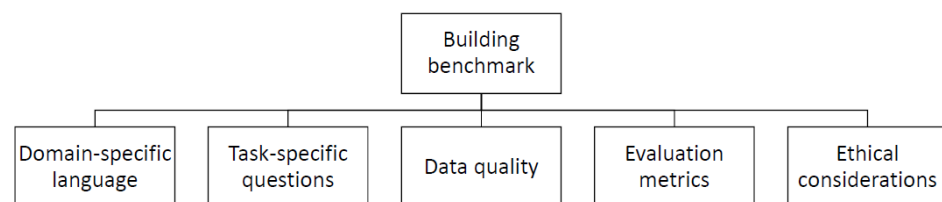


Figure 1. Key factors for building an NLP benchmark in the assembly domain.

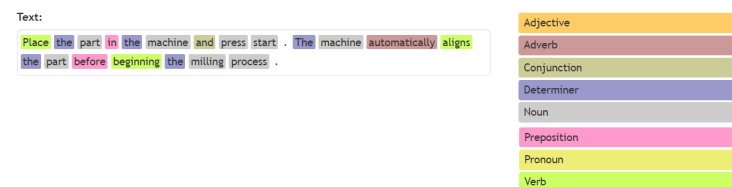
In this research, we adopt a methodology similar to that described in [67], encompassing the definition of the task, the introduction of the dataset within the assembly domain, and the establishment of evaluation metrics. First, we outline the specific task. Following that, we present the proposed dataset designed for the assembly domain. Finally, we introduce the evaluation metrics utilized to assess task performance. In essence, the benchmark's purpose lies in the assessment of task performance by contextualizing it within a specific text domain, furnishing it with a tangible dataset, and employing well-defined evaluation metrics.

3.1. The Task

In this sub-section, we focus on the defined task: part-of-speech (POS) tagging. POS tagging is a crucial step in NLP in the case of improving the performance of a system related to information retrieval and it has many practical applications, such as text-to-speech synthesis, machine translation, and information retrieval [77]. This can be considered as an initial stage of processing aimed at the ultimate objective of enabling computers to comprehend human language [78]. It is the task of labeling or tagging each token in sentences based on the defined rule [79,80]. A token refers to a sequence of characters, such as a word or a punctuation mark, that serves as a unit of input for NLP tasks. Tokenization is the process of breaking a text into individual tokens, allowing for further analysis and processing [77]. POS tagging is useful for a variety of NLP tasks, such as information extraction, entity recognition, and grammatical structure identification. It automatically assigns the parts of speech tags to the tokens considering two main aspects: finding the exact tags for each token and choosing between the possible tags for ambiguous tokens [81–83]. Figure 2 shows an example of the output of a POS tagger in regard to two different sentences considering the eight classes of parts of speech tags [77] as *Nouns*, *Verbs*, *Adjectives*, *Pronouns*, *Determiners*, *Adverbs*, *Prepositions*, and *Conjunctions* (Table 2).

Table 2. Considered classes of part of speech tags.

Class of Tags	Description
<i>Noun</i>	Refer to entities in the world like people, animals, and things
<i>Verb</i>	Describe actions, activities, and states
<i>Adjective</i>	Describe properties of nouns
<i>Pronoun</i>	Used as a substitute for a noun or noun phrase
<i>Determiner</i>	Providing additional information or to specify the reference of the noun
<i>Adverb</i>	Modify verbs, adjectives, providing extra information
<i>Preposition</i>	Prototypically express spatial relationships
<i>Conjunction</i>	Used to connect words, phrases, or clauses within a sentence

**Figure 2.** Visualizing part-of-speech tags for two sentences.

The main goal of developing the POS tagger for any language is to improve the accuracy of tagging and also to consider the different language structures, seeking to remove ambiguity in the tokens [84]. Two relevant factors to improve the performance and accuracy of POS tagging are the number of tokens in the training and testing data and also the corpus or the open-source dictionary being used in POS tagging [77]. In NLP, a corpus refers to a large and structured collection of text or language data that is used for analysis and linguistic research. On the other hand, a dictionary in NLP typically refers to a structured resource that contains words or terms along with their corresponding definitions, translations, and other linguistic information. It serves as a reference for lexical information and can be used for tasks like word lookup and semantic analysis [85]. In this research, we consider eight classes of parts of speech as mentioned in Table 2.

3.2. The Dataset

In our approach to building a benchmark dataset for the assembly domain, we categorize sentences based on two distinct structures commonly found in assembly-related content: imperative and declarative sentences. In NLP, imperative and declarative sentences are two types of sentence structures with distinct characteristics and functions [86]. An imperative sentence is a type of sentence that gives a command, makes a request, or gives an instruction. It is usually written in the present tense and begins with a verb. On the other hand, a declarative sentence is a type of sentence that makes a statement, expresses a fact, or conveys information. It typically has a subject and a predicate.

In a benchmark, a representative portion of the multiple-class category refers to a balanced distribution of the data across all the classes in the classification task. Kwong [70] with 350 questions in their benchmark, considering different classes on sentence structures, categorized 80% of the questions as interrogative, 11% as imperative, 5% as declarative, with 4% categorized as others [87]. In the TV show database benchmark with 25,076 statements, considering two categories as imperative and non-imperative, imperative statements took up about 8.8%.

In assembly benchmarks, both imperative (directive) and declarative (descriptive) sentences can be used. Imperative sentences are often used for instructions and informative sentences, such as “Place the part in the machine and press start”, while declarative sentences are used to convey information or describe a process, such as “The machine automatically aligns the part before beginning the milling process”. Both types of sentences can be used in different tasks of the benchmark, such as text classification, information extraction, and question answering.

However, considering the mentioned categories in the different academic references and also taking into account the different documents used in the assembly line in the assembly domain, in this research, we consider four different categories in the assembly domain as follows: Warnings, Informative texts, Manuals, and Work instructions.

- Manual documents are important in manufacturing as they provide detailed instructions on how to operate, maintain, and repair various types of equipment. (The manual is on the level of equipment.) These documents typically contain textual descriptions and diagrams or illustrations to help users better understand the processes involved. Manuals may also include technical documents used in manufacturing, such as installation manuals, service manuals, and troubleshooting guides. The manual mostly includes declarative sentences.
- Warning documents in manufacturing are documents that provide warnings and safety instructions to users who may operate or come into contact with machinery or equipment. These documents are typically included with products or machinery and provide information on how to properly operate, maintain, and service them to avoid potential hazards. Examples of warning documents in manufacturing include safety manuals, warning labels, caution tags, and safety data sheets. These documents use mostly imperative sentences to convey warnings and instructions to ensure the safety of those interacting with the equipment.
- Informative texts in the assembly domain provide additional information that is relevant to the task at hand, but not necessarily part of the direct instructions or warnings. These types of texts can provide important context and support for workers, helping them to complete their tasks safely and efficiently, and mostly include imperative sentences.
- Work instructions are documents that provide step-by-step guidance to operators, technicians, or assembly line workers on how to perform a specific task or operation. (The work instructions are on the level of process.) These instructions typically include information on the tools or equipment required, safety procedures, quality checks, and other relevant information and mostly contain imperative sentences. Work instructions may be presented in various formats, such as text, diagrams, photographs, or videos. They are essential for ensuring that products are manufactured consistently and to the required quality standards, and for training new employees. Work instructions are often updated and revised based on feedback from workers and changes in the manufacturing process.

In the prepared benchmark, we consider around half of the sentences of the well-written structured manual sentences to be declarative sentences, and the other half of the sentences, comprising three different sub-categories in semi-structured sentences (Warning, Informative texts, and Work instructions), to be imperative sentences. With the imperative sentences, we consider 20% of the sentences to have an incomplete structure. In the previous research, we considered 100 sentences, including 45 declarative sentences and 55 imperative sentences within the four categories in the assembly domain. Here, in order to make the benchmark more diverse and representative of a wider range of languages and also to check the effectiveness of the built benchmark, we add 100 new sentences to the benchmark. So, in the new benchmark, we have 104 sentences that are imperative and 96 sentences that are declarative.

Thus, with this extension, we create a set including sentences picked from four different categories in the assembly domain, Warning, Informative texts, Manuals, and Work instructions. Having 2752 tokens (excluding the punctuation marks), we perform the annotation for all the tokens manually. The introduced eight POS tags based on our reference [77] are assigned to each token in the considered sentences. After performing the manual POS tagging with the help of an expert, we investigate which of the considered open-source libraries achieves the best result in the test set. After manually tagging the considered corpus, we have 523 *verbs*, 68 *pronouns*, 419 *prepositions*, 896 *nouns*, 352 *determiners*, 172 *conjunctions*, 70 *adverbs*, and 251 *adjectives* in our annotated dataset. Table 3, shows the distribution of the

tags in the considered corpus based on the assembly context. In addition, Figure 3 shows the stacked bars of imperative and declarative sentences based on internal partitions of different tags in the considered ground truth.

Table 3. Tag distribution in the considered benchmark dataset based on four assembly categories.

Sentences	Adjective	Adverb	Conjunction	Determiner	Noun	Preposition	Pronoun	Verb
Manual	167	50	75	155	436	211	36	267
Warning	28	10	34	41	127	66	8	84
Informative	23	5	28	37	85	36	4	60
Work instruction	33	5	35	119	248	106	20	112
Declarative	167	50	75	155	436	211	36	267
Imperative	84	20	97	197	460	208	32	256

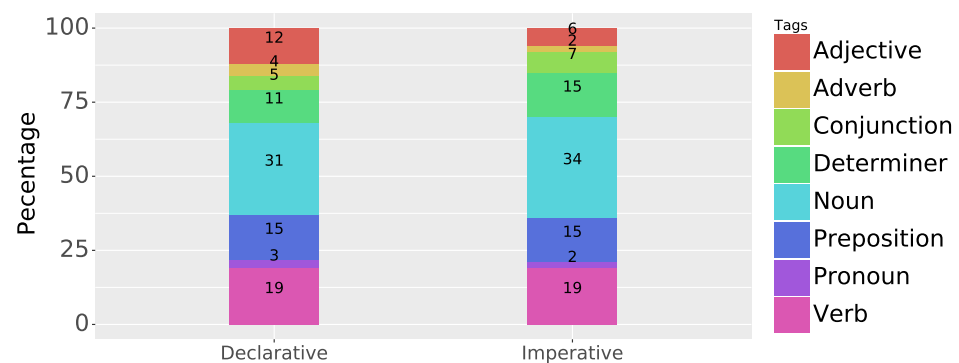


Figure 3. Percentage partitions of tags in both declarative and imperative sentences in the considered ground truth.

To the best of our knowledge, there are no publicly available POS annotated training data on the assembly domain of manufacturing. In order to implement POS tagging, this research leveraged three prominent open-source NLP libraries: Stanford, SpaCy, and NLTK (Natural Language Toolkit), serving as the framework for validating the prepared benchmark. Traditional NLP libraries offer pre-trained models optimized for POS tagging tasks, rendering them adept at efficiently processing large volumes of text. Their interpretability enhances understanding and analysis of POS tagging results—an advantage over deep learning-based models. The Stanford library [46], known for its robust capabilities in NLP, further enriched our analysis. It provides comprehensive tools for tokenization, part-of-speech tagging, and more, enhancing the depth of our investigation. The NLTK (Natural Language Toolkit) library [88], a leading platform for Python-based human language data manipulation, SpaCy [89], a powerful Python and Cython library for advanced NLP tasks, and the newly added Stanford library, collectively formed the cornerstone of our research toolkit. These libraries empower our work in the assembly domain of manufacturing, facilitating tasks such as named entity recognition, sentiment analysis, text classification, and beyond. They also offer a wide range of pre-trained models for multiple languages, augmenting the versatility of our NLP pipelines.

3.3. The Evaluation Metrics

There are several evaluation metrics [26] that can be used to evaluate the performance of a benchmark in NLP, depending on the specific task and the goals of the benchmark. In this research, we consider accuracy, precision, recall, and the F1-score as the evaluation metrics. Brief definitions of the considered metrics are provided below:

- **Accuracy:** Accuracy measures the overall correctness of the model's predictions and is calculated as the ratio of correctly predicted instances to the total instances.

- **Precision:** Precision quantifies the model's ability to make correct positive predictions and is calculated as the ratio of true positives to the sum of true positives and false positives.
- **Recall:** Recall assesses the model's capacity to identify all relevant instances and is calculated as the ratio of true positives to the sum of true positives and false negatives.
- **F1-score:** The F1-score, as a measure that balances precision and recall, calculated as the harmonic mean of the two values, is used as the evaluation metric in many types of research related to benchmark performance comparison [66,68,78].

4. Result and Discussion

To assess the performance of pre-trained taggers libraries, such as NLTK, SPACY, and Stanford, in the assembly domain corpus, and to determine if the newly created benchmark can effectively perform NLP tasks, the study utilized a ground truth consisting of 2752 tokens divided into four categories. The libraries' taggers were applied to the ground truth data to compare the tags produced by the libraries with the manually annotated ones. The tagging accuracy of the NLTK tagger compared to the ground truth was 90%, an improvement from the previous research's 87% accuracy with 100 initial sentences. In addition, recall 90%, precision 91%, and F1-score 90% were estimated for NLTK. The tagging accuracy of the SpaCy tagger on the ground truth was 93%, which is 3% more accurate than the NLTK tagger. This is consistent with the previous research, which achieved a 93% tagging accuracy with 100 initial sentences. In addition, recall, precision, and the F1-score were estimated as 93% for SpaCy. The Stanford tagger, as the newly added library, had 91% accuracy compared to the ground truth, 91% recall, 92% precision, and 91% F1-score.

In the task of token recognition and comparison with the ground truth, the NLTK tagger achieved a perfect match by accurately identifying 2752 tokens, exhibiting a token accuracy rate of 100%. The NLTK tagger demonstrated the ability to correctly handle hyphenated compound words, such as "oil-lubricated", treating them as single tokens. In contrast, the SpaCy tagger successfully identified all tokens in the corpus but segmented hyphenated compound words into separate tokens, resulting in 2792 identified tokens. This discrepancy occurred because the SpaCy tagger treated hyphenated compound words as three distinct tokens, for instance, "oil-lubricated" being recognized as "oil", "-", and "lubricated". The assembly corpus contained a total of 36 hyphenated compound tokens. A similar issue arose for the Stanford library as the tagger identified 2786 tokens, recognizing 6 out of 36 hyphenated compound words.

The accuracy of individual tags obtained by the considered libraries is presented in Figure 4. With regard to analyzing the three key parts of speech, namely *nouns*, *verbs*, and *adjectives*, SPACY achieved a high accuracy of 95% for *verbs*, 92% for *nouns*, and 78% for *adjectives*. NLTK achieved an accuracy of 88% for *verbs*, 91% for *nouns*, and 82% for *adjectives*. The considered accuracy for similar tags in the Stanford library was 94% for *verbs*, 94% for *nouns*, and 78% for *adjectives*. The three libraries demonstrated satisfactory accuracy in recognizing most tags. However, the NLTK and Stanford libraries exhibited the lowest accuracy of 62% specifically for *conjunction* tags when compared to the other tags.

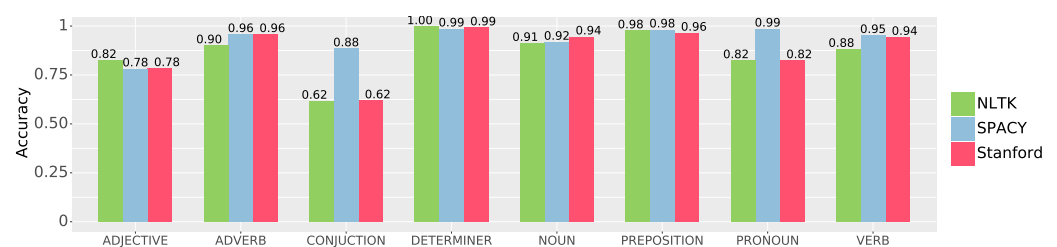


Figure 4. Accuracy for each individual tag on the considered assembly corpus of NLTK, SPACY, and Stanford libraries taggers.

The absolute numbers of mis-annotations for each type of token in NLTK, SPACY, and Stanford are presented in Figure 5. In NLTK, as shown in Table 4, the highest percentage of mis-annotations, at 38%, occurred for *conjunctions*. Specifically, 35 *conjunctions* were mistakenly annotated as *prepositions*, and 24 were annotated as *adverbs*. Among the 251 *adjective* tokens, 18% were mis-annotated, with 30 of them incorrectly labeled as *nouns*. Additionally, 11 out of the 12 mis-annotated tokens pertaining to *pronouns* were erroneously annotated as *determiners*. There were instances of mis-annotations where *nouns* and *verbs* were mistakenly labeled as each other or as *adjectives*. Among the tokens analyzed in the SPACY library, *adjectives* had the highest rate of mis-annotation at 24%. Among these mis-annotations, 28 tokens were mistakenly labeled as *nouns* and 27 as *verbs*. Additionally, 18 tokens originally identified as *conjunctions* were mislabeled as *prepositions* and *pronouns*. Similar to NLTK, there were instances where *nouns* and *verbs* were incorrectly labeled as each other. Furthermore, 22 *nouns* were mistakenly labeled as *pronouns* and 26 as *adjectives*. For Stanford, the highest mis-annotation was 38%, which occurred for *conjunctions*, mostly mis-annotated as *prepositions* and *adverbs* as for NLTK. A total of 30 *adjective* tokens were mistakenly annotated as *nouns* and 24 *adjective* tokens were mistakenly annotated as *verbs*. In addition 24 out of 30 mis-annotated *verbs* were identified as *nouns* in Stanford taggers.

Upon closer examination of mis-annotations, distinct patterns emerged in the labeling behavior of NLTK, SpaCy, and Stanford NLP. Within NLTK, the most prevalent mislabeling occurrences involved *conjunctions* being incorrectly labeled as *prepositions*, *pronouns* mistakenly identified as *determiners*, *conjunctions* labeled as *adverbs*, and *adjectives* mislabeled as *nouns*. In the case of SpaCy, a notable tendency was observed in misclassifying *adjectives* as *nouns* and *adjectives* as *verbs*. Stanford NLP exhibited its own set of common mislabeling instances, including *adjectives* being miscategorized as *nouns* and *verbs*, *conjunctions* mislabeled as *adverbs* and *prepositions*, *nouns* inaccurately tagged as *adjectives* and *verbs*, and *verbs* erroneously identified as *nouns*.

Table 4. Mis-annotations of token types in NLTK, SpaCy, and Stanford libraries.

	NLTK	SPACY	Stanford
Adjective	18.0%	22.0%	22.0%
Adverb	10.0%	4.0%	4.0%
Conjunction	38.0%	12.0%	38.0%
Determiner	0.2%	1.0%	0.8%
Noun	9.0%	8.0%	6.0%
Preposition	2.0%	2.0%	4.0%
Pronoun	18.0%	1.0%	18.0%
Verb	12.0%	5.0%	6.0%

To assess the classification of declarative and imperative sentences, we designated the manual category as declarative, while the warning, informative, and work-instruction categories were categorized as imperative. We evaluated the performance of each library separately for these two sentence types across the eight tags. The NLTK library attained an accuracy of 91% for declarative sentences and 89% for imperative sentences. Similarly, SpaCy exhibited accuracies of 91% for declarative sentences and 95% for imperative sentences. Additionally, Stanford achieved 90% accuracy for declarative sentences and 93% for imperative sentences.

Figure 6 illustrates a comparison between NLTK, SPACY, and Stanford in terms of imperative and declarative sentences. In the case of imperative sentences, NLTK exhibited lower accuracy for *adverbs* and *conjunctions* compared to other tag types. Conversely, SPACY performed well across all tag types for imperative sentences. In the Stanford library, for the imperative sentences, *conjunctions* were less accurately tagged compared to other tags.

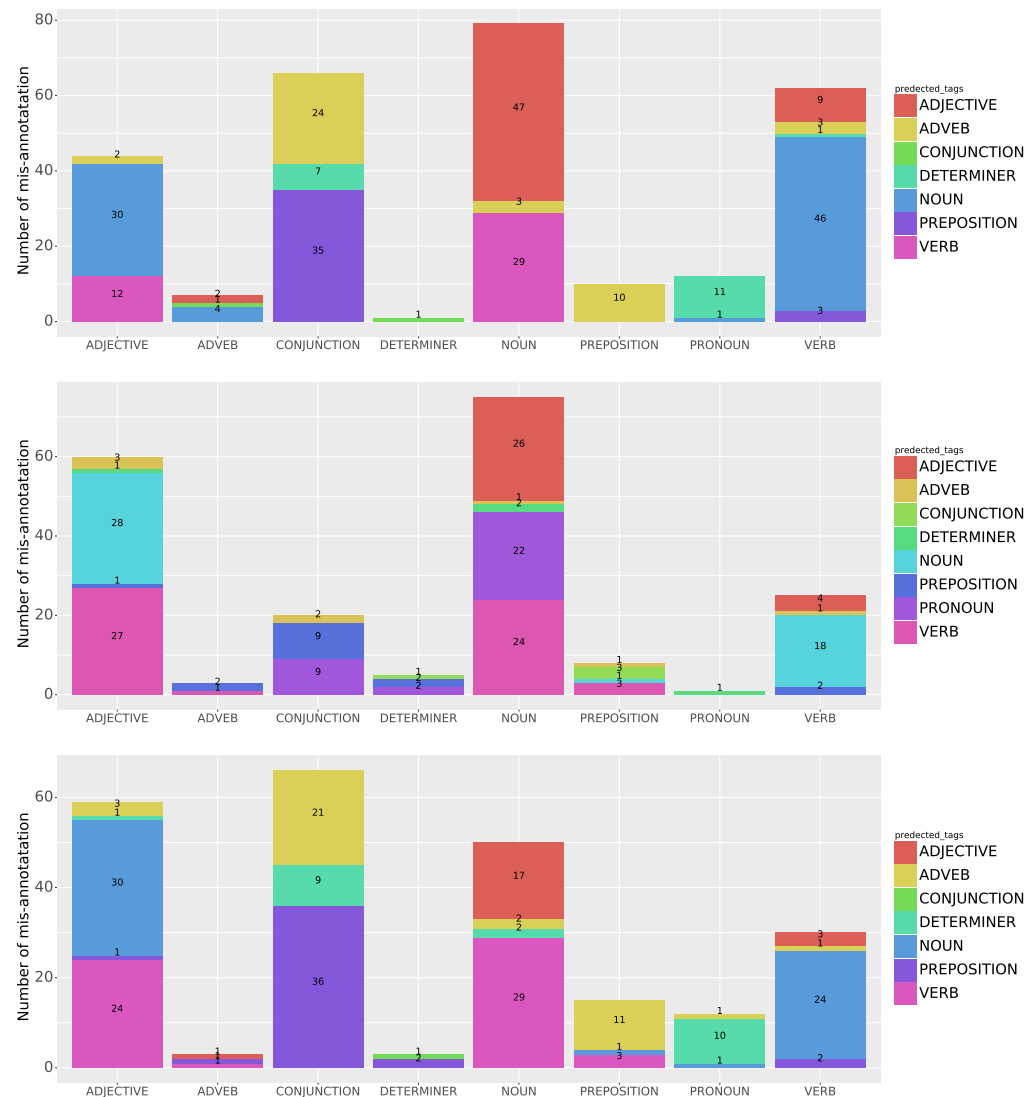


Figure 5. Comparison of mis-annotations in 1-NLTK, 2-SpaCy, and 3-Stanford: Absolute numbers of mislabeling by token type.

Further analysis revealed the percentage of mis-annotations for each tag in imperative and declarative sentences, as depicted in Figure 7 for each respective library. The confusion matrix presented in Figure 7 highlights the predominant types of mis-annotations observed in NLTK for declarative sentences, which include *pronouns* mislabeled as *determiners*, *adjectives* mislabeled as *nouns*, *conjunctions* mislabeled as *adverbs*, and *conjunctions* mislabeled as *prepositions*. Similarly, for imperative sentences in NLTK, the prevalent mislabeled tags encompassed *conjunctions* mislabeled as *prepositions*, *adverbs* mislabeled as *nouns*, *verbs* mislabeled as *nouns*, and *adjectives* mislabeled as *nouns*. In the case of SpaCy, there were no mis-annotations exceeding 10% for imperative sentences, while for declarative sentences, the prominent mislabeled tags consisted of *adjectives* mislabeled as *nouns*, *adjectives* mislabeled as *verbs*, and *conjunctions* mislabeled as *pronouns*. For the Stanford taggers, in declarative sentences, the mislabel happened in *adjectives* to *nouns*, *conjunctions* to *adverbs*, *determiners*, and *prepositions*, and finally, *pronouns* to *determiners*. For the imperative sentences, the only mis-annotation higher than 10% was related to mislabeling *conjunctions* to *prepositions*.

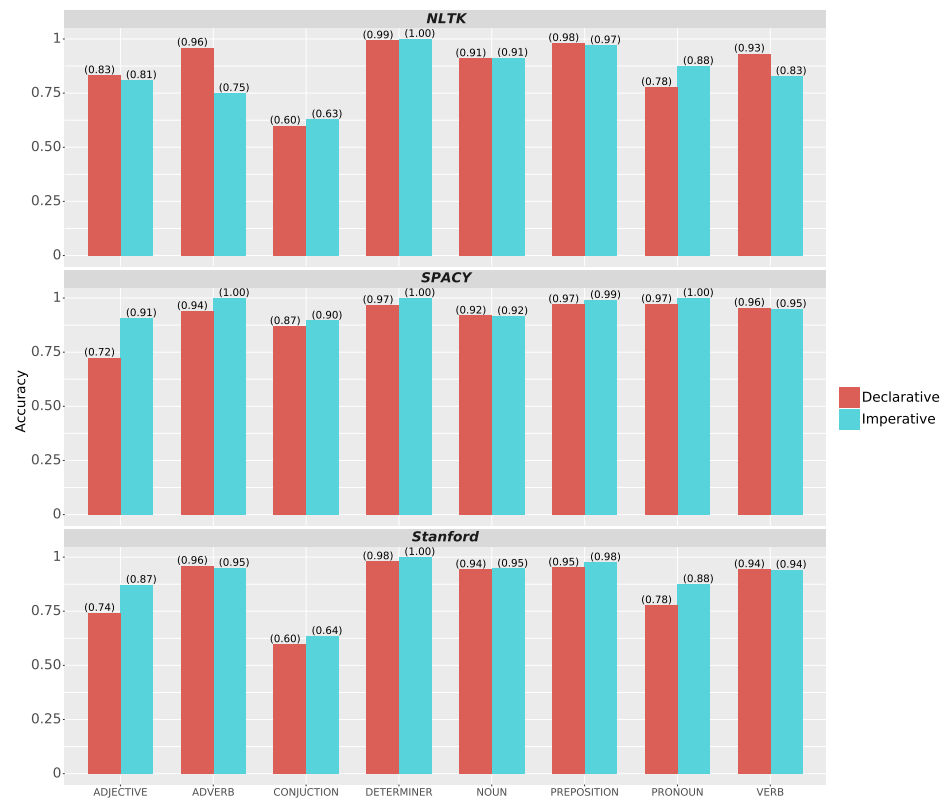


Figure 6. Comparison of NLTK, SpaCy, and Stanford performance in classifying declarative and imperative sentences across multiple tags.

Considering the incompleteness of sentences, which accounted for 20% of the imperative sentences within the benchmark under examination, the results revealed that out of 140 tokens in the incomplete sentences, NLTK annotated 121 tokens correctly. SpaCy, on the other hand, identified 141 tokens within incomplete sentences, of which 138 tokens were annotated accurately. In Stanford, when recognizing 141 tokens, 132 of them were correctly annotated.

Within the diverse landscape of our dataset, where sentence lengths exhibit variations across our four distinctive categories, we embarked on an exploration of whether sentence length plays a pivotal role in the tagging performance of our NLP libraries: NLTK, SpaCy, and Stanford. To investigate this, we conducted a comprehensive correlation analysis, considering the interplay between sentence length and the number of mis-annotations in each library. We meticulously evaluated this relationship, examining each library and category individually. This resulted in 12 correlation analyses, providing a granular view of whether certain libraries exhibited preferences for tagging accuracy in sentences of specific lengths within different categories. Our findings, depicted in Figure 8 and Table 5, shed light on the nuanced interactions between sentence length and tagging performance. These figures offer insights into whether longer or shorter sentences are more prone to mis-annotations for each library and category, enabling a deep understanding of their performance intricacies. According to the figures, 7 out of 12 pairs had a significant positive correlation and 5 out of 12 had no statistically significant correlation. For NLTK, informative text and work instructions, for SpaCy, manual and informative text, and for Stanford, work instructions had no significant positive correlation. Additionally, to assess the impact on imperative and declarative sentences, we conducted similar correlation analyses. In general, NLTK and Stanford had a positive correlation for both imperative and declarative sentences. SpaCy had a positive correlation with imperative sentences. However, there was no statistically significant correlation in declarative sentences in the SpaCy library.

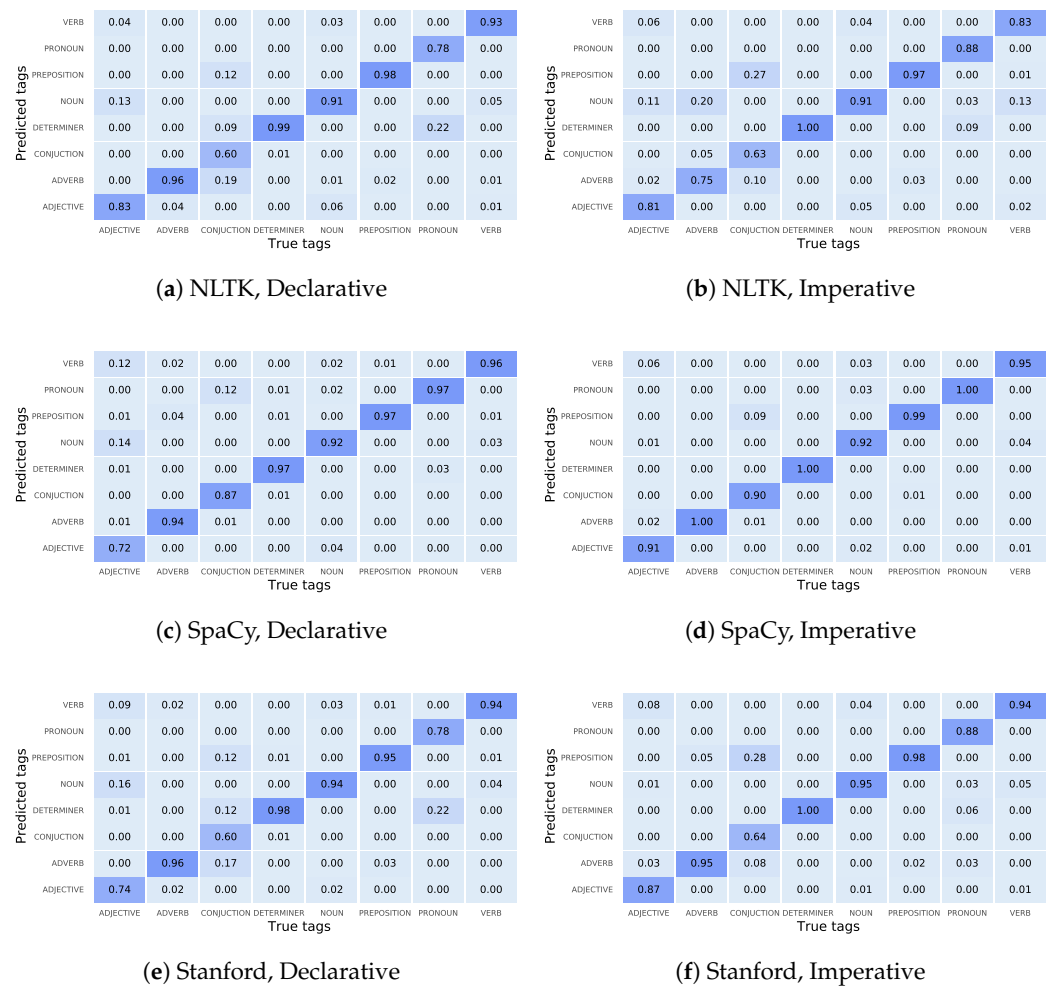


Figure 7. Comparison of tag mis-annotation in NLTK, SpaCy, and Stanford for imperative and declarative sentences: confusion matrix analysis.

Table 5. Correlation analysis: sentence length vs. mis-annotation in NLTK, SpaCy, and Stanford libraries for 4 different categories.

Category	NLTK	SPACY	Stanford
Manual	9×10^{-4}	4.700×10^{-1}	3.30×10^{-2}
Warning	7.20×10^{-4}	8.600×10^{-3}	3.2×10^{-6}
Informative text	1.500×10^{-1}	3.800×10^{-1}	7.5×10^{-3}
Work instruction	8.800×10^{-1}	4.2×10^{-3}	6.30×10^{-2}

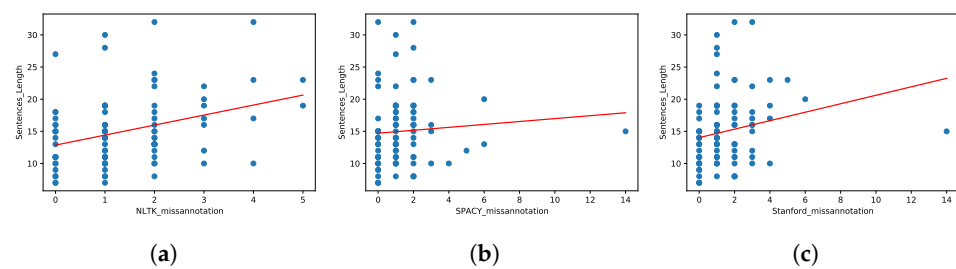


Figure 8. Cont.

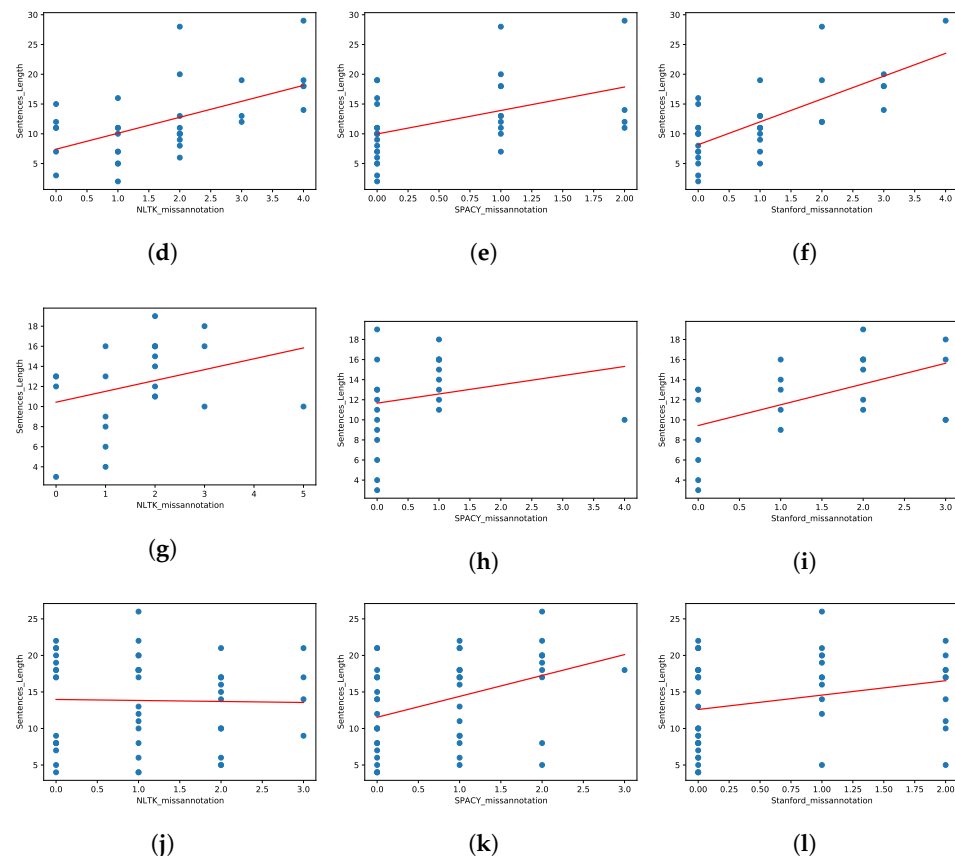


Figure 8. Correlation analysis: sentence length vs. mis-annotation in different categories and libraries. (a) Manual, NLTK, sentence length, (b) Manual, SPACY, sentence length, (c) Manual, Stanford, sentence length, (d) Warning, NLTK, sentence length, (e) Warning, SPACY, sentence length, (f) Warning, Stanford, sentence length, (g) Informative text, NLTK, sentence length, (h) Informative text, SPACY, sentence length, (i) Informative text, Stanford, sentence length, (j) Work instruction, NLTK, sentence length, (k) Work instruction, SPACY, sentence length, (l) Work instruction, Stanford, sentence length.

5. Conclusions

In this study, we investigated the application of natural language processing techniques in the assembly domain of manufacturing industry. By creating a domain-specific dataset and exploring strategies for handling incomplete sentences, we sought to address the challenges posed by unstructured open questions and incomplete sentences in operator assistance systems. The outcomes of this research contribute to the advancement of NLP applications in the assembly domain, providing valuable insights for the development of intelligent and efficient assembly processes.

In light of the limited availability of a comprehensive corpus in the assembly domain, we present a novel dataset as a meticulously constructed benchmark comprising 2752 tokens pertaining to four distinct categories (Manual, Warning, Informative text, and Work instructions) within the assembly. This dataset encompasses both imperative and declarative sentence forms, mirroring the complexity inherent in assembly instructions. The percentage of tokens in this built dataset is 51% for declarative sentences and 49% for imperative sentences. Also, in order to check the incomplete sentences, we consider 20% of the imperative sentences to be incomplete ones.

By manually assigning POS tags to the tokens within the dataset and utilizing the NLP open-source libraries SPACY, NLTK, and Stanford, we assessed the effectiveness of their pre-trained taggers in accurately labeling assembly-related concepts. Our objective was to evaluate the extent to which these libraries can achieve high levels of accuracy in tagging assembly concepts, leveraging their existing capabilities in NLP.

In all aspects of our study, including tagging accuracy, token recognition, part-of-speech annotation, mis-annotations, and sentence classification, the evaluated NLP libraries, NLTK, SpaCy, and Stanford, exhibited robust performance. Each library displayed unique strengths and behaviors, offering valuable insights for the development of intelligent assembly processes and operator assistance systems in the manufacturing domain. While our study achieved significant progress in accurately tagging assembly concepts and improving the performance of NLP libraries, it is important to acknowledge the challenges we encountered. We observed instances of mis-annotation within the dataset in the assembly domain. These mis-annotations, although limited in number, highlight the ongoing complexity of NLTK, especially in specialized hyphenated compound words in the assembly domain.

In conclusion, this research significantly advances the application of NLP in the assembly domain, offering a nuanced understanding of the capabilities and limitations of prominent NLP libraries. The insights gained from this study lay the groundwork for future advancements in intelligent assembly processes, strengthening the foundation for Industry 4.0 transformations in the manufacturing sector.

Author Contributions: All authors contributed to the study conception and design. The first draft of the manuscript was written by F.B.M., and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript. All the authors participated in preparing the paper and read and approved the final manuscript to be sent to the journal for publication.

Funding: This research was funded by the Flanders Make organization under the project OperatorAssist_SBO, project number 2021-0133. Flanders Make is the Flemish strategic research center for the manufacturing industry.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Due to the data being collected from a real case study and based on agreement with that specific production line, the data used in this paper are confidential. The code can be made available for reviewers on request. However, the code would be sent without the data used in the case study.

Conflicts of Interest: The authors have no relevant financial or non-financial interests to disclose.

References

1. Nunes, M.L.; Pereira, A.C.; Alves, A.C. Smart products development approaches for Industry 4.0. *Procedia Manuf.* **2017**, *13*, 1215–1222. [\[CrossRef\]](#)
2. Ghobakhloo, M. Industry 4.0, digitization, and opportunities for sustainability. *J. Clean. Prod.* **2020**, *252*, 119869. [\[CrossRef\]](#)
3. Longo, F.; Nicoletti, L.; Padovano, A. Smart operators in industry 4.0: A human-centered approach to enhance operators' capabilities and competencies within the new smart factory context. *Comput. Ind. Eng.* **2017**, *113*, 144–159. [\[CrossRef\]](#)
4. Bagnasco, A.; Chirico, M.; Parodi, G.; Scapolla, A.M. A model for an open and flexible e-training platform to encourage companies' learning culture and meet employees' learning needs. *J. Educ. Technol. Soc.* **2003**, *6*, 55–63.
5. Moencks, M.; Roth, E.; Bohné, T.; Kristensson, P.O. Human-computer interaction in industry: A systematic review on the applicability and value-added of operator assistance systems. *Found. Trends Hum. Interact.* **2022**, *16*, 65–213. [\[CrossRef\]](#)
6. Urgo, M.; Tarabini, M.; Tolio, T. A human modelling and monitoring approach to support the execution of manufacturing operations. *CIRP Ann.* **2019**, *68*, 5–8. [\[CrossRef\]](#)
7. Mark, B.G.; Gualtieri, L.; Rauch, E.; Rojas, R.; Buakum, D.; Matt, D.T. Analysis of user groups for assistance systems in production 4.0. In Proceedings of the 2019 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Macao, China, 15–18 December 2019; pp. 1260–1264.
8. Yang, X.; Plewe, D.A. Assistance systems in manufacturing: A systematic review. In Proceedings of the Advances in Ergonomics of Manufacturing: Managing the Enterprise of the Future: Proceedings of the AHFE 2016 International Conference on Human Aspects of Advanced Manufacturing, Walt Disney World®, Florida, USA, 27–31 July 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 279–289.
9. Magerkurth, C.; Engelke, T.; Röcker, C. The smart dice cup: A radio controlled sentient interaction device. In Proceedings of the Entertainment Computing-ICEC 2006: 5th International Conference, Cambridge, UK, 20–22 September 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 211–216.
10. Gorecky, D.; Campos, R.; Chakravarthy, H.; Dabelow, R.; Schlick, J.; Zühlke, D. Mastering Mass Customization—a Concept for Advanced, Human-Centered Assembly. *Acad. J. Manuf. Eng.* **2013**, *11*, 62–67.

11. Röcker, C.; Etter, R. Social radio: A music-based approach to emotional awareness mediation. In Proceedings of the 12th International Conference on Intelligent User Interfaces, Honolulu, HI, USA, 28–31 January 2007; pp. 286–289.
12. Röcker, C. Universal access to awareness information: Using smart artefacts to mediate awareness in distributed teams. *Univers. Access Inf. Soc.* **2012**, *11*, 259–271. [[CrossRef](#)]
13. Ukita, N.; Kaulen, D.; Röcker, C. A user-centered design approach to physical motion coaching systems for pervasive health. In *Smart Health: Open Problems and Future Challenges*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 189–208.
14. Mueller, E.T. *Commonsense Reasoning: An Event Calculus Based Approach*; Morgan Kaufmann: Malvern, UK, 2014.
15. Vajjala, S.; Majumder, B.; Gupta, A.; Surana, H. *Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems*; O'Reilly Media: Newton, MA, USA, 2020.
16. Chiche, A.; Yitagesu, B. Part of speech tagging: A systematic review of deep learning and machine learning approaches. *J. Big Data* **2022**, *9*, 1–25. [[CrossRef](#)]
17. Mishra, A.; Jain, S.K. A survey on question answering systems with classification. *J. King Saud Univ.-Comput. Inf. Sci.* **2016**, *28*, 345–361. [[CrossRef](#)]
18. Antoniou, C.; Bassiliades, N. A survey on semantic question answering systems. *Knowl. Eng. Rev.* **2022**, *37*, 345–361. [[CrossRef](#)]
19. Shi, M. Knowledge graph question and answer system for mechanical intelligent manufacturing based on deep learning. *Math. Probl. Eng.* **2021**, *2021*, 6627114. [[CrossRef](#)]
20. Xingguang, L.; Zhenbo, C.; Zhengyuan, S.; Haoxin, Z.; Hangcheng, M.; Xuesong, X.; Gang, X. Building a Question Answering System for the Manufacturing Domain. *IEEE Access* **2022**, *10*, 75816–75824. [[CrossRef](#)]
21. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
22. Bowman, S.R.; Dahl, G.E. What will it take to fix benchmarking in natural language understanding? *arXiv* **2021**, arXiv:2104.02145.
23. v. Kistowski, J.; Arnold, J.A.; Huppler, K.; Lange, K.D.; Henning, J.L.; Cao, P. How to build a benchmark. In Proceedings of the 6th ACM/SPEC International Conference on Performance Engineering, Austin, TX, USA, 31 January–4 February 2015; pp. 333–336.
24. Rajpurkar, P.; Zhang, J.; Lopyrev, K.; Liang, P. Squad: 100,000+ questions for machine comprehension of text. *arXiv* **2016**, arXiv:1606.05250.
25. Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; Bowman, S.R. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv* **2018**, arXiv:1804.07461.
26. Powers, D.M. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv* **2020**, arXiv:2010.16061.
27. Wang, A.; Pruksachatkun, Y.; Nangia, N.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; Bowman, S. Superglue: A stickier benchmark for general-purpose language understanding systems. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 3266–3280.
28. Ferrari, A.; Spagnolo, G.O.; Gnesi, S. Pure: A dataset of public requirements documents. In Proceedings of the 2017 IEEE 25th International Requirements Engineering Conference (RE), Lisbon, Portugal, 4–8 September 2017; pp. 502–505.
29. Kurihara, K.; Kawahara, D.; Shibata, T. JGLUE: Japanese general language understanding evaluation. In Proceedings of the Thirteenth Language Resources and Evaluation Conference, Marseille, France, 22–25 June 2022; pp. 2957–2966.
30. Diefenbach, D.; Lopez, V.; Singh, K.; Maret, P. Core techniques of question answering systems over knowledge bases: A survey. *Knowl. Inf. Syst.* **2018**, *55*, 529–569. [[CrossRef](#)]
31. Berant, J.; Chou, A.; Frostig, R.; Liang, P. Semantic parsing on freebase from question-answer pairs. In Proceedings of the 2013 conference on empirical methods in natural language processing, Seattle, DC, USA, 18–21 October 2013; pp. 1533–1544.
32. Bordes, A.; Usunier, N.; Chopra, S.; Weston, J. Large-scale simple question answering with memory networks. *arXiv* **2015**, arXiv:1506.02075.
33. Pereira, A.; Trifan, A.; Lopes, R.P.; Oliveira, J.L. Systematic review of question answering over knowledge bases. *IET Softw.* **2022**, *16*, 1–13. [[CrossRef](#)]
34. Cairns, B.L.; Nielsen, R.D.; Masanz, J.J.; Martin, J.H.; Palmer, M.S.; Ward, W.H.; Savova, G.K. The MiPACQ clinical question answering system. In Proceedings of the AMIA Annual Symposium Proceedings, Washington DC, USA, 22–26 October 2011; American Medical Informatics Association: Bethesda, MD, USA, 2011; Volume 2011, p. 171.
35. Johnson, A.E.; Pollard, T.J.; Shen, L.; Lehman, L.W.H.; Feng, M.; Ghassemi, M.; Moody, B.; Szolovits, P.; Anthony Celi, L.; Mark, R.G. MIMIC-III, a freely accessible critical care database. *Sci. Data* **2016**, *3*, 1–9. [[CrossRef](#)] [[PubMed](#)]
36. Uzuner, Ö.; South, B.R.; Shen, S.; DuVall, S.L. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. *J. Am. Med. Inform. Assoc.* **2011**, *18*, 552–556. [[CrossRef](#)] [[PubMed](#)]
37. Lombardo, G.; Pellegrino, M.; Adosoglou, G.; Cagnoni, S.; Pardalos, P.M.; Poggi, A. Machine Learning for Bankruptcy Prediction in the American Stock Market: Dataset and Benchmarks. *Future Internet* **2022**, *14*, 244. [[CrossRef](#)]
38. Akhil, K.; Rajimol, R.; Anoop, V. Parts-of-Speech tagging for Malayalam using deep learning techniques. *Int. J. Inf. Technol.* **2020**, *12*, 741–748. [[CrossRef](#)]
39. Anastasyev, D.; Gusev, I.; Indenbom, E. Improving part-of-speech tagging via multi-task learning and character-level word representations. *arXiv* **2018**, arXiv:1807.00818.
40. Mutabazi, E.; Ni, J.; Tang, G.; Cao, W. A review on medical textual question answering systems based on deep learning approaches. *Appl. Sci.* **2021**, *11*, 5456. [[CrossRef](#)]

41. Yitagesu, S.; Zhang, X.; Feng, Z.; Li, X.; Xing, Z. Automatic part-of-speech tagging for security vulnerability descriptions. In Proceedings of the 2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR), Madrid, Spain, 17–19 May 2021; pp. 29–40.
42. Kumar, S.; Kumar, M.A.; Soman, K. Deep learning based part-of-speech tagging for Malayalam Twitter data (Special issue: Deep learning techniques for natural language processing). *J. Intell. Syst.* **2019**, *28*, 423–435. [CrossRef]
43. Mohammed, S. Using machine learning to build POS tagger for under-resourced language: The case of Somali. *Int. J. Inf. Technol.* **2020**, *12*, 717–729. [CrossRef]
44. Rezai, M.J.; Mosavi Miangah, T. FarsiTag: A part-of-speech tagging system for Persian. *Digit. Scholarsh. Humanit.* **2017**, *32*, 632–642. [CrossRef]
45. Patoary, A.H.; Kibria, M.J.B.; Kaium, A. Implementation of automated Bengali parts of speech tagger: An approach using deep learning algorithm. In Proceedings of the 2020 IEEE Region 10 Symposium (TENSYP), Dhaka, Bangladesh, 5–7 June 2020; pp. 308–311.
46. Manning, C.; Surdeanu, M.; Bauer, J.; Finkel, J.; Bethard, S.; McClosky, D. Stanford CoreNLP a Suite of Core NLP Tools. Available online: <http://stanfordnlp.github.io/CoreNLP/> (accessed on 17 November 2020).
47. Petrov, S. Announcing Syntaxnet: The World's Most Accurate Parser Goes Open Source. *Google Res. Blog* **2016**, *12*, 42. Available online: <https://blog.research.google/2016/05/announcing-syntaxnet-worlds-most.html> (accessed on 1 February 2021).
48. Loper, E.; Bird, S. Nltk: The natural language toolkit. *arXiv* **2002**, arXiv:cs/0205028.
49. Explosion, A. Spacy-Industrial-Strength Natural Language Processing in Python. Available online: <https://spacy.io> (accessed on 1 February 2021).
50. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
51. Sanh, V.; Debut, L.; Chaumond, J.; Wolf, T. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv* **2019**, arXiv:1910.01108.
52. Lin, T.J.; Abhishek, N.V. Personal Identity Information Detection using Synthetic Dataset. In Proceedings of the 2023 6th International Conference on Applied Computational Intelligence in Information Systems (ACIIS), Darussalam, Brunei, 23–25 October 2023; pp. 1–5.
53. Bawa, V.; Baroud, I.; Schaffer, S. “HalloBzar”: A German chatbot for accessing the regional digital marketplace. In *INFORMATIK 2023—Designing Futures: Zukünfte Gestalten*; Gesellschaft für Informatik e.V.: Bonn, Germany, 2023; pp. 1607–1614. [CrossRef]
54. Danenas, P.; Skersys, T. Exploring Natural Language Processing in Model-To-Model Transformations. *IEEE Access* **2022**, *10*, 116942–116958. [CrossRef]
55. Phan, T.H.; Do, P. NER2QUES: Combining named entity recognition and sequence to sequence to automatically generating Vietnamese questions. *Neural Comput. Appl.* **2022**, *34*, 1593–1612. [CrossRef]
56. Forth, K.; Abualdenien, J.; Borrmann, A. Calculation of embodied GHG emissions in early building design stages using BIM and NLP-based semantic model healing. *Energy Build.* **2023**, *284*, 112837. [CrossRef]
57. Chantrapornchai, C.; Tunsakul, A. Information extraction on tourism domain using SpaCy and BERT. *ECTI Trans. Comput. Inf. Technol.* **2021**, *15*, 108–122.
58. Das, S.; Deb, N.; Cortesi, A.; Chaki, N. Extracting goal models from natural language requirement specifications. *J. Syst. Softw.* **2024**, *211*, 111981. [CrossRef]
59. Schmitt, X.; Kubler, S.; Robert, J.; Papadakis, M.; LeTraon, Y. A replicable comparison study of NER software: StanfordNLP, NLTK, OpenNLP, SpaCy, Gate. In Proceedings of the 2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS), Granada, Spain, 22–25 October 2019; pp. 338–343.
60. Nemes, L.; Kiss, A. Information extraction and named entity recognition supported social media sentiment analysis during the COVID-19 pandemic. *Appl. Sci.* **2021**, *11*, 11017. [CrossRef]
61. Al Omran, F.N.A.; Treude, C. Choosing an NLP library for analyzing software documentation: A systematic literature review and a series of experiments. In Proceedings of the 2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR), Buenos, Argentina, 20–21 May 2017; pp. 187–197.
62. Dietz, L.; Chatterjee, S.; Lennox, C.; Kashyapi, S.; Oza, P.; Gamari, B. Wikimarks: Harvesting Relevance Benchmarks from Wikipedia. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, 11–15 July 2022; pp. 3003–3012.
63. Marcus, M.P.; Santorini, B.; Marcinkiewicz, M.A. Building a Large Annotated Corpus of English: The Penn Treebank. *Comput. Linguist.* **1993**, *19*, 313–330.
64. Marcus, M.; Kim, G.; Marcinkiewicz, M.A.; MacIntyre, R.; Bies, A.; Ferguson, M.; Katz, K.; Schasberger, B. The Penn treebank: Annotating predicate argument structure. In Proceedings of the Human Language Technology: Proceedings of a Workshop, Plainsboro, NJ, USA, 8–11 March 1994.
65. Levesque, H.; Davis, E.; Morgenstern, L. The winograd schema challenge. In Proceedings of the Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning, Rome, Italy, 10–14 June 2012.
66. Cagliero, L.; La Quatra, M. Inferring multilingual domain-specific word embeddings from large document corpora. *IEEE Access* **2021**, *9*, 137309–137321. [CrossRef]

67. Kierszbaum, S.; Klein, T.; Lapasset, L. ASRS-CMFS vs. RoBERTa: Comparing Two Pre-Trained Language Models to Predict Anomalies in Aviation Occurrence Reports with a Low Volume of In-Domain Data Available. *Aerospace* **2022**, *9*, 591. [\[CrossRef\]](#)
68. Gao, Y.; Dligach, D.; Miller, T.; Caskey, J.; Sharma, B.; Churpek, M.M.; Afshar, M. DR. BENCH: Diagnostic Reasoning Benchmark for Clinical Natural Language Processing. *J. Biomed. Inform.* **2023**, *138*, 104286. [\[CrossRef\]](#)
69. Manna, R.; Das, D.; Gelbukh, A. Question-answering and recommendation system on cooking recipes. *Comput. Y Sist.* **2021**, *25*, 223–235. [\[CrossRef\]](#)
70. Kwong, H.; Yorke-Smith, N. Detection of imperative and declarative question-answer pairs in email conversations. *AI Commun.* **2012**, *25*, 271–283. [\[CrossRef\]](#)
71. Chandra, Y.W.; Suyanto, S. Indonesian chatbot of university admission using a question answering system based on sequence-to-sequence model. *Procedia Comput. Sci.* **2019**, *157*, 367–374. [\[CrossRef\]](#)
72. Khabiri, E.; Gifford, W.M.; Vinzamuri, B.; Patel, D.; Mazzoleni, P. Industry specific word embedding and its application in log classification. In Proceedings of the 28th Acm International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 2713–2721.
73. Giachos, I.; Papakitsos, E.C.; Antonopoulos, I.; Laskaris, N. Systemic and hole semantics in human-machine language interfaces. In Proceedings of the 2023 17th International Conference on Engineering of Modern Electric Systems (EMES), Oradea, Romania, 9–10 June 2023; pp. 1–4.
74. Heng, F.N.R.; Deris, M.M.; Basir, N. A Similarity Precision for Selecting Ontology Component in an Incomplete Sentence. In Proceedings of the Recent Advances on Soft Computing and Data Mining: Proceedings of the Third International Conference on Soft Computing and Data Mining (SCDM 2018), Johor, Malaysia, 6–7 February 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 95–104.
75. Shin, D.; Kam, H.J.; Jeon, M.S.; Kim, H.Y. Automatic classification of thyroid findings using static and contextualized ensemble natural language processing systems: Development study. *JMIR Med. Inform.* **2021**, *9*, e30223. [\[CrossRef\]](#)
76. Quan, T.T. N/A Modern Approaches in Natural Language Processing. *VNU J. Sci. Comput. Sci. Commun. Eng.* **2022**, *39*. [\[CrossRef\]](#)
77. Manning, C.; Schütze, H. *Foundations of Statistical Natural Language Processing*; MIT Press: Cambridge, MA, USA, 1999.
78. Liao, Z.; Zeng, Q.; Wang, Q. Chinese Word POS Tagging with Markov Logic. In Proceedings of the Intelligence and Security Informatics: Pacific Asia Workshop, PAISI 2015, Ho Chi Minh City, Vietnam, 15 May 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 91–101.
79. Kumawat, D.; Jain, V. POS tagging approaches: A comparison. *Int. J. Comput. Appl.* **2015**, *118*, 62510340. [\[CrossRef\]](#)
80. Chungku, C.; Rabgay, J.; Faaß, G. Building NLP resources for Dzongkha: A tagset and a tagged corpus. In Proceedings of the Eighth Workshop on Asian Language Resources, Beijing, China, 21–22 August 2010; pp. 103–110.
81. Lv, C.; Liu, H.; Dong, Y.; Chen, Y. Corpus based part-of-speech tagging. *Int. J. Speech Technol.* **2016**, *19*, 647–654. [\[CrossRef\]](#)
82. Singh, J.; Joshi, N.; Mathur, I. Part of speech tagging of Marathi text using trigram method. *arXiv* **2013**, arXiv:1307.4299.
83. Das, B.R.; Sahoo, S.; Panda, C.S.; Patnaik, S. Part of speech tagging in odia using support vector machine. *Procedia Comput. Sci.* **2015**, *48*, 507–512. [\[CrossRef\]](#)
84. Cing, D.L.; Soe, K.M. Improving accuracy of part-of-speech (POS) tagging using hidden markov model and morphological analysis for Myanmar Language. *Int. J. Electr. Comput. Eng.* **2020**, *10*, 2023. [\[CrossRef\]](#)
85. McEnery, T.; Hardie, A. *Corpus Linguistics: Method, Theory and Practice*; Cambridge University Press: Cambridge, UK, 2011.
86. Jurafsky, D. *Speech & Language Processing*; Pearson Education India: Bangalore, India, 2000.
87. Xiao, Y.; Slaton, Z.Y.; Xiao, L. TV-AfD: An Imperative-Annotated Corpus from The Big Bang Theory and Wikipedia’s Articles for Deletion Discussions. In Proceedings of the Twelfth Language Resources and Evaluation Conference, Marseille, France, 11–16 May 2020; pp. 6542–6548.
88. Bird, S. NLTK: The natural language toolkit. In Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions, Sydney, Australia, 17–18 July 2006; pp. 69–72.
89. Altinok, D. *Mastering spaCy: An End-to-End Practical Guide to Implementing NLP Applications Using the Python Ecosystem*; Packt Publishing Ltd.: Birmingham, UK, 2021.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.