

Article

A Semantic Annotation Pipeline towards the Generation of Knowledge Graphs in Tribology

Patricia Kügler ^{1,*}, Max Marian ², Rene Dorsch ¹, Benjamin Schleich ¹ and Sandro Wartzack ¹

¹ Engineering Design, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Martensstr. 9, 91058 Erlangen, Germany; rene.dorsch@fau.de (R.D.); schleich@mfk.fau.de (B.S.); wartzack@mfk.fau.de (S.W.)

² Department of Mechanical and Metallurgical Engineering, School of Engineering, Pontificia Universidad Católica de Chile, Macul, Santiago 6904411, Chile; max.marian@ing.puc.cl

* Correspondence: kuegler@mfk.fau.de

Abstract: Within the domain of tribology, enterprises and research institutions are constantly working on new concepts, materials, lubricants, or surface technologies for a wide range of applications. This is also reflected in the continuously growing number of publications, which in turn serve as guidance and benchmark for researchers and developers. Due to the lack of suited data and knowledge bases, knowledge acquisition and aggregation is still a manual process involving the time-consuming review of literature. Therefore, semantic annotation and natural language processing (NLP) techniques can decrease this manual effort by providing a semi-automatic support in knowledge acquisition. The generation of knowledge graphs as a structured information format from textual sources promises improved reuse and retrieval of information acquired from scientific literature. Motivated by this, the contribution introduces a novel semantic annotation pipeline for generating knowledge in the domain of tribology. The pipeline is built on Bidirectional Encoder Representations from Transformers (BERT)—a state-of-the-art language model—and involves classic NLP tasks like information extraction, named entity recognition and question answering. Within this contribution, the three modules of the pipeline for document extraction, annotation, and analysis are introduced. Based on a comparison with a manual annotation of publications on tribological model testing, satisfactory performance is verified.

Keywords: tribo-testing; tribo-informatics; machine learning; artificial intelligence; natural language processing; tribAIn; BERT



Citation: Kügler, P.; Marian, M.; Dorsch, R.; Schleich, B.; Wartzack, S. A Semantic Annotation Pipeline towards the Generation of Knowledge Graphs in Tribology. *Lubricants* **2022**, *10*, 18. <https://doi.org/10.3390/lubricants10020018>

Received: 14 December 2021

Accepted: 23 January 2022

Published: 25 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 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

The emergence of efficient and sustainable technologies represents a major challenge for the 21st century. While renewable energy sources are increasingly replacing fossil fuels in order to reduce CO₂ emissions, the influence of friction and wear on the energy efficiency of a wide range of technical processes has hardly reached public awareness. However, these offer considerable potential for saving CO₂ and resources. Holmberg and Erdemir [1] estimated that roughly 23% of the global primary energy is consumed to overcome friction and to repair/replace worn components in tribo-technical systems. The authors predicted that these energy losses could be reduced by up to 40% through tribological advances. Accordingly, companies and research institutions are focusing on new concepts, materials, lubricants, or surface technologies in a wide range of applications. This is also reflected in the continuously growing number of publications related to the domain of tribology, which in turn serve as inspiration, guidance, and benchmark for researchers and developers, but which are almost impossible to keep up with due to their vast quantity and the associated complexity and diversity. Thereby, profound data bases in combination with machine learning (ML) and artificial intelligence (AI) approaches can support sorting through the complexity of patterns and identifying trends [2]. Therefore, they are more

and more employed in the analysis, design, optimization, or monitoring of tribological systems in various fields [3], ranging from composite materials [4], drive technology [5,6], manufacturing [7], surface engineering [8,9], or lubricant formulation [10,11]. As pointed out by Marian and Tremmel [12], novel findings and additional value in the domain of tribology can especially be created by extracting knowledge from available literature and drawing higher-level conclusions. For example, Kurt and Oduncuoglu [13] trained artificial neural networks (ANNs) with data from literature to study the influence of normal load, sliding speed as well as the type and weight fraction of various reinforcement phases within a polyethylene matrix on the resulting friction and wear behavior. Similarly, Vinoth and Datta [14] utilized 153 data sets from literature to predict mechanical properties of carbon nanotube or graphene reinforced polyethylene in dependency of composition, particle size, and bulk properties by means of an ANN. Subsequently, multi-objective optimization by genetic algorithms and corresponding experimental validation actually demonstrated improved tribological properties compared to the references. Using 80 data sets from four-ball-tests and 120 data sets from pin-on-disk experiments with varying base oils and friction modifiers as reported in literature as well as an ANN and a genetic algorithm, Bhaumik et al. [15] optimized the lubricant formulation and experimentally validated their results. The aforementioned studies indicate the potential through leveraging knowledge from the available literature. However, the data acquisition and processing still are very manual in the field of tribology, involving the review of publications and the extraction of relevant (most frequently textually/descriptive) information, which limits the generation of sophisticated and broad databases and thus the further use of ML/AI [12].

High manual efforts to acquire and curate information and knowledge for further processing are not limited to the domain of tribology and is known as “knowledge acquisition bottleneck” [16]. Although the latter has been discussed since the rise of expert systems in the 1980s [17], for instance with the purpose of tribological design decisions [18] or failure diagnosis [19] to mention two examples from the tribological domain, knowledge acquisition and thus knowledge engineering are still quite manual and time-consuming tasks. Studer et al. [20] argue that knowledge engineering is a modeling activity, which goes beyond the simple transfer of directly accessible knowledge into an appropriate computer representation towards a model construction process [21]. In consequence, knowledge structuring and modeling plays an important role in the knowledge acquisition process. Hoekstra [22] therefore refers to a “knowledge reengineering bottleneck”, which highlights the general difficulty of continuously reusing existing generic and assertional knowledge. The latter refers to data-level or object knowledge, while generic knowledge concerns schema-level describing conceptual knowledge and is represented as a domain theory to structure the respective domain. This includes the decision on used vocabulary to describe the domain and a representation form to formalize the model. Chandrasegaran et al. [23], as well as Verhagen et al. [24], emphasized the importance of semantic interoperability for knowledge reuse and sharing, which is frequently dealt with ontological models represented in formal logics. According to Gruber [25], an ontology is an “explicit specification of a conceptualization”. This means that an ontology can be used to explicitly define a domain model for sharing and reusing structured knowledge by humans and machines. In other domains, e.g., bioinformatics, ontologies are widely used for knowledge structuring, data integration and decision support systems [26]. One successful example is the Gene Ontology (GO) [27], which provides broadly accepted vocabulary for annotating gene product data from different databases and sources. Exploiting ontologies for accessing and reusing experimental knowledge has also been pursued in the domain of tribology. One example is the “OntoCommons” project (<https://ontocommons.eu/industrial-domain-ontologies>, accessed on 14 December 2021), where a tribological use case aimed at reducing efforts in tribological experiments by reusing existing knowledge. Thereby, Esnaola-Gonzalez and Fernandez [28] argue, that semantic technologies, and more specifically ontologies propose a suited representation for the vaguely documented results of experiments. Within the domain of materials science, the “European Materials Modelling Ontology” (EMMO,

<https://emmc.info/emmo-info/>, accessed on 14 December 2021) provides a representational ontology based on materials modelling and characterization knowledge. Furthermore, we recently introduced the tribAIn ontology [29] for reusing knowledge from tribological experiments. The domain ontology was built for the purpose of providing a common and machine-readable schema for structuring tribological experiments intending to improve reuse and shareability of testing results from different sources. Since this contribution relies on the tribAIn ontology, more detailed information is provided in Section 2.2. In addition to schema-level generic knowledge, assertional knowledge refers to specific knowledge objects, e.g., results from individual experiments. As mentioned before, assertional knowledge from experiments in the domain of tribology is usually published in natural language, thus publications are a well-suited knowledge source for acquiring the current state of tribological findings. Dealing with natural language sources is usually problematic since it is ambiguous and unstructured. Moreover, textual descriptions may be incomplete in the sense of formal models. Due to the time-consuming process of acquiring and structuring knowledge from textual sources in systematic literature studies or manual database construction, those knowledge bases are not suited for long-term reuse and continuous extension. A successful example for generating structured information from textual sources is the DBpedia project [30], which extracts structured data from Wikipedia content using templates and pattern matching techniques. The structured format then allows querying the vast content in a sophisticated way instead of searching articles by keywords and processing the information manually. In terms of the results from tribological experiments, publications—similar to Wikipedia—contain structured (e.g., operational parameters, wear rate, coefficient of friction, etc.) and unstructured knowledge (for example interpretive description and discussion of results). By extracting the information from text in a structured way, the knowledge can be queried, processed, and compared. Thus, one could query for tribological experiments on desired materials and testing conditions, for example dry-running pin-on-disk model tests with various reinforcement phases within composites or deposited coatings on the specimen surfaces.

A large-scale employment of aforementioned knowledge extraction approaches, however, strongly demands for strategies for (semi-)automatically streamlining data acquisition. Therefore, this contribution aims at the introduction of a semantic annotation pipeline based upon natural language processing (NLP) methods in order to overcome the “knowledge reengineering bottleneck” in the domain of tribology. The motivation behind this contribution is mainly inspired by the current practice in biomedical research, where a massive growth in published research articles led to increasing attention for automated information extraction methods to support human researchers [31,32]. Regarding similar challenges, like sharing research outcomes via natural language publications, semantic ambiguity and interdisciplinary nature of the domain, this contribution is a first attempt to apply (semi-)automatic knowledge acquisition techniques within the domain of tribology. Therefore, while the methods used within this contribution have already shown potential in similar knowledge acquisition and structuring issues within the biomedical domain, this paper aims at the effective use of these methods in tribology. The contribution is structured as follows: First, the applied methods for the acquisition pipeline are introduced, containing a description of the underlying domain theory of tribological test methods as a generic schema as well as the relevant semantic web and NLP techniques, especially named entity recognition and question answering under the use of the BERT language model [33]. The semantic annotation pipeline and packages used for implementation are summarized in Section 3. Subsequently, the access-level and performance of the pipeline are demonstrated in Section 4, including a description of the Web-User-Interface and a technical evaluation of the single modules of the pipeline. Finally, we discuss the potentials and limitations of the pipeline, as well as connections and outlooks to further approaches in Section 5.

2. Theory and Methods

2.1. Domain Theory from Tribology

As mentioned before, generic knowledge builds a domain theory, which can be represented as a formal ontology. In terms of semantic annotation, the domain theory is used as structured metadata, the unstructured resource is enriched with. Therefore, relevant concepts and relations from established methodologies of tribological testing are used to build the schema for the semantic annotation pipeline. Generally, a tribological system can be described by its system structure, input and output variables and their functional conversion within the open or closed system boundary [34] (Figure 1). The system structure consists of the relatively moving body and counter-body, which are rubbing against each other and may be completely or partially separated by an intermediate medium (liquid or gaseous). Operational input variables, such as loads, kinematics, duration, and temperatures, can be summarized in the stress collective. Depending on the latter, as well as any disturbance variables and the system structure, the body and counter-body physically and chemically interact at temporally and spatially varying locations. On the one hand, this results in loss variables such as friction and wear, which cause changes in the surface, loss of material and energy dissipation. On the other hand, this results in the actual functional variables of the tribological system. The mechanisms and applications of tribology extend over several size scales. This ranges from processes on the nano- or micro-level in the field of physics, chemistry, and material sciences, such as the formation of boundary layers or the shearing of nanoparticle layers and ends with machine elements and assemblies as well as multiple tribological contacts in the engineering sciences in the micrometer to meter range, for example in rolling bearings or gears. Accordingly, tribometry, i.e., tribological measuring and testing technology, covers all dimensional ranges of tribology determining friction and wear parameters of tribological systems. The significance of various quantifiable measured variables, e.g., a friction coefficient averaged over time or a wear coefficient, usually depends on the underlying mechanisms, the measurement method, and the objective of the study. Given the function and structure of tribological systems, tribological testing can be divided into six categories according to the simplification of the system structure, the stress collective or the environmental conditions. While original and complete systems are tested under real operating and environmental conditions in field tests (category I), this is carried out under laboratory conditions with merely practical operating conditions in test bench tests (II). In aggregate (III) and component tests (IV), this is further reduced to the investigation of original aggregates or components. Specimen tests (V) are conducted with specimens that are similar to the components and subjected to similar stresses as in the target application. Finally, model tests (VI) involve fundamental analyses of friction and wear processes with simplified specimens under defined loads. Typical representatives of the latter are disk-on-disk, cylinder-on-cylinder, ball-on-disk or pin-on-disk tribometer tests. The advantages of the individual test categories can be combined by a suitable test chain [34].

2.2. The TribAIn Ontology

Kügler et al. [29] introduced the tribAIn ontology as a schema for structuring, reusing and sharing experimental knowledge within the tribological domain. The ontology was modelled highly relying on existing tribological test methods (see Section 2.1 and [34]). The presence of a common and shared methodology as well as terminology are vital assumptions for specifying a formal ontology of a domain, since those build a strong and accepted conceptualization, the formal specification relies on. Furthermore, the ontology is based on the EXPO ontology (ontology of scientific experiments) introduced by Soldatova and King [37], which is a generic formal description of experiments. Since tribAIn shares the same purpose of efficient analysis, annotation and sharing of results from scientific experiments, EXPO concepts were reused and further specified for the domain of tribology. The tribAIn ontology is formalized in OWL (Web Ontology Language) [38], which is a common ontology language based on description logics (DLs) [39]. Knowledge formalized with an

ontology language is expressed in form of triples: <subject> <predicate> <object>, which means the ontology can be visualized as a directed graph with named relations between two classes (concepts). In the following, we will use Turtle Syntax [40] for streamlining triples of tribAIn. Since every object within an OWL ontology has a unique identifier, the prefix *tAI* is used for the tribAIn IRI (Internationalized Resource Identifier), thus concepts and relations of the tribAIn namespace can be identified by this prefix. The ontology provides concepts to describe the three main working areas “Experimental Design”, “Procedure” and “Experimental Results” (Figure 2). The concepts from these areas structure the information about a specific experiment, with the tribological system (*tAI:Tribosystem*) investigated, pre-processing procedures (*tAI:IndustrialProcess* and subclasses) as well as the test procedure (*tAI:TribologicalTesting*) itself and links that information with the outcome of the investigation (*tAI:OutputParameter* and subclasses). Due to the close relation to the underlying methodology (cf. Section 2.1), the concepts refer to common terms within the domain of tribology. Parameters or variables, for instance loads or temperatures, are described using a pattern containing the two triples: *Parameter hasValue xsd:float* and *Parameter hasUnit Unit*. The first triple links a value of the datatype float to an instance of the class (or some subclass of) *Parameter*, while the other triple links a unit to the same instance. In this manner, measurement series are generated in a consistent fashion, which can be compared and analyzed.

Due to the design of the tribAIn ontology, a knowledge base (KB) which uses the ontology as schema, can be queried in terms of the following example questions (cf. [29]):

- Which tribological systems were investigated under dry-running conditions using a solid lubricant coating?
- Which variables were tested regarding their influence on the behavior of a material pairing?
- Which wear rate was calculated of sample XY?

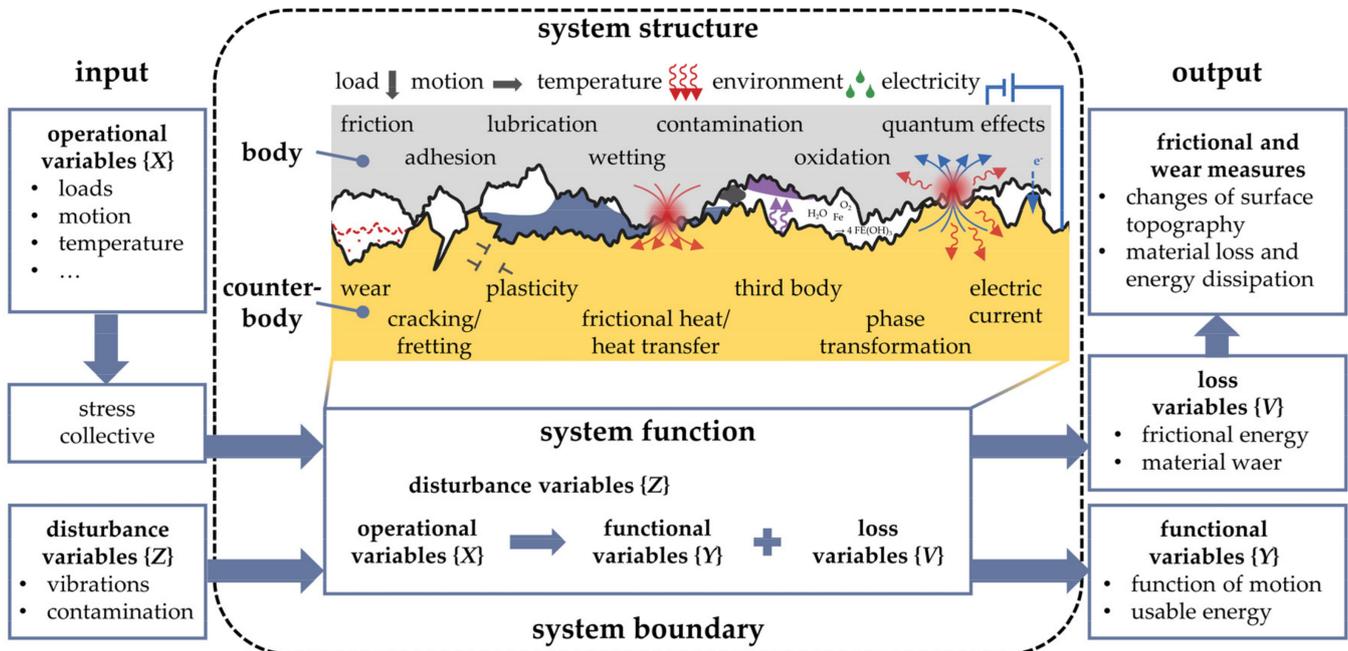


Figure 1. Overall representation of a tribological system, its target function, and interactions in tribological contacts. Redrawn and adapted from [29,34–36].

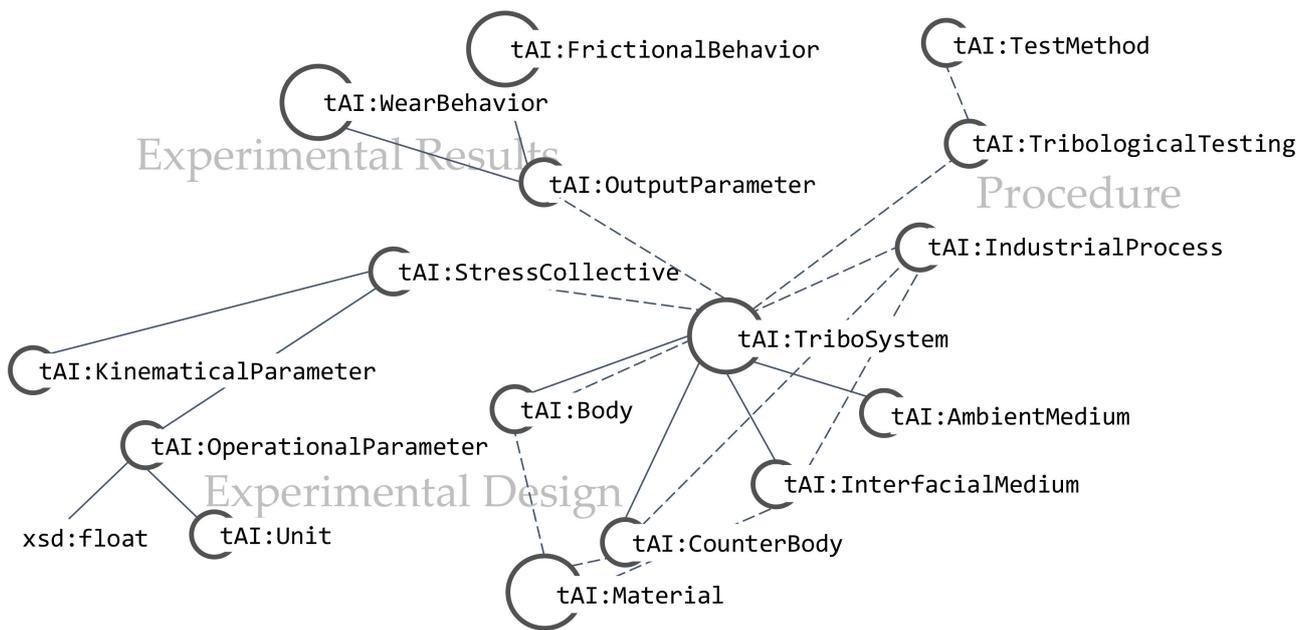


Figure 2. Excerpt of relevant tribAI concepts. Redrawn and adapted from [29].

2.3. Ontologies, Knowledge Graphs and Semantic Annotation

As Gruber [41] states, within ontologies, definitions associate names of the entities within a universe of discourse. Therefore, the schema provided by an ontology can be shared among different knowledge graphs, which hold the actual data. This is referred to as ontological commitment and is a guarantee for consistency, even for incomplete knowledge, since there are agreements to use a shared vocabulary [41]. Those commitments to a specific vocabulary (or terminology) are also implicitly made within natural language communication. Within the domain of tribology, they exist for instance for the description of a tribological system (cf. Figure 1). Since tribological testing should enable reproducible and comparable results, experiments must be built upon a common methodology, which defines the system structure as well as input- and output parameters. Describing experiments as well as results within a scientific publication under the use of a common terminology is a first step of knowledge formalization (Figure 3).

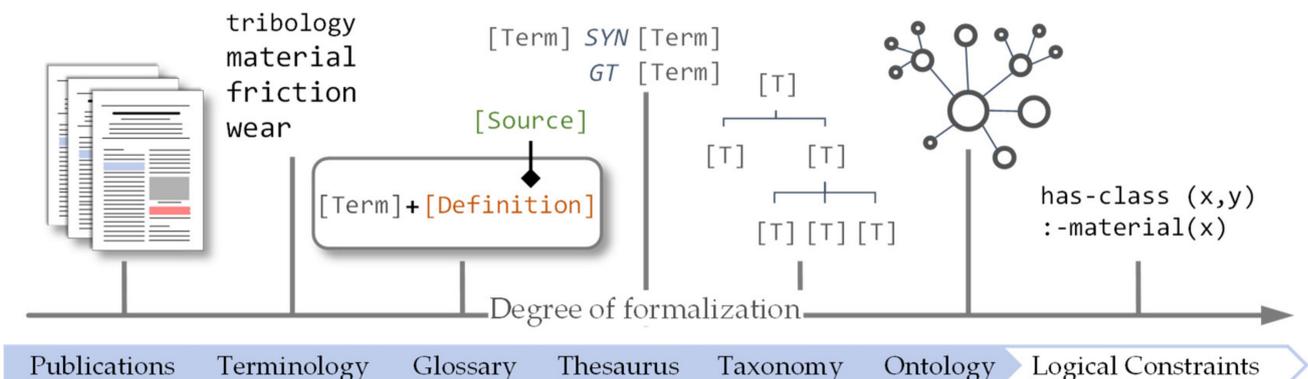


Figure 3. Different degrees of formalization from natural language text to logical constraints. Redrawn and adapted from [42].

Nevertheless, the challenge with sharing knowledge among natural language publications is the vague or even insufficient description, since often knowledge about the domain theory is assumed to be present to the human reader. An example is the following

description from the materials section of an experimental study on $\text{Ti}_3\text{C}_2\text{T}_x$ nanosheets (MXenes) [43–45] investigated as solid lubricant for machine elements [46]:

“Commercially available thrust ball bearings 51201 according to ISO 104 [. . .] consisting of shaft washer, housing washer and ball cage assembly were used as substrates (Figure 1a).”

With background knowledge of the tribological domain, it becomes clear, that the studied tribological system here is a certain thrust ball bearing and with the information given by the referenced figure, the coated parts can be identified by a human reader. However, within the textual description, the link to the underlying methodology is not stated explicitly. Thus, the documentation of the experiment is incomplete and ambiguous from a formalization perspective. In other words, a semantic gap between textual descriptions from publications and the general knowledge models with a higher degree of formalization (Figure 3) prevents machine-supported processing of existing tribological knowledge from publications. In order to bridge such a gap, semantic annotation is the process of joining natural language and formal semantic models (e.g., an ontology) [47]. A semantic annotation of the example cited above associated with ontological concepts from the tribAIIn ontology [29] is shown in Figure 4. In this example, the string “thrust ball bearings 51201” is recognized as referring to the instance *tbb_51201*, which is a tribological system (“*tbb_51201* a *tAI:TriboSystem*” in the triple notation of Figure 4). Furthermore, the components of the thrust ball bearing are referred to the instances *sw_51201* (shaft washer), *hw_51201* (housing washer) and *as_bc_51201* (ball cage assembly) and are annotated as parts of the tribological system within the triple notation. Annotating text snippets semantically to instances of an ontology enriches the natural language text with machine-readable context. For example, the instance “*tbb_51201*” may not only be referred to the experimental testing described in the publication, but also be linked within the knowledge graph to information from the ISO 104 mentioned within the text snippet. Therefore, the semantic annotation process links mentions of entities from different sources to knowledge objects within a knowledge graph, which are further semantically defined by an ontological schema.

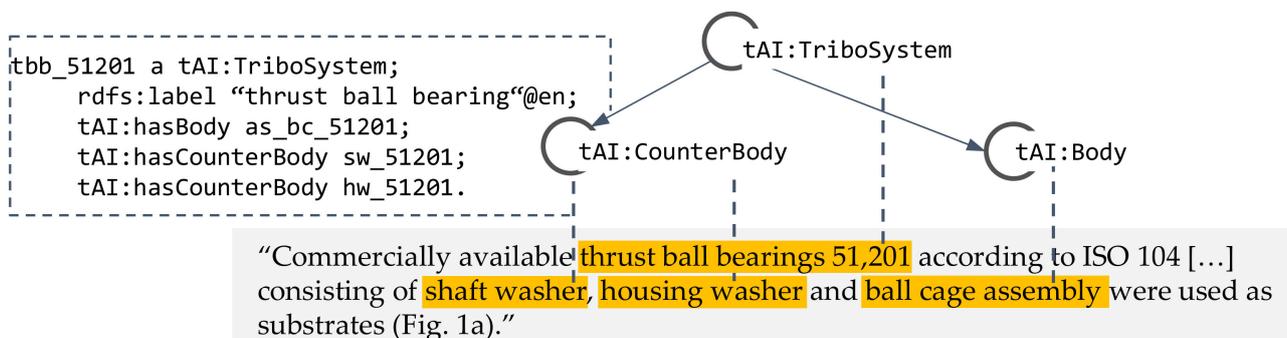


Figure 4. Example of a semantic annotation of a text excerpt from [46] with concepts from the tribAIIn ontology [29] graphically visualized and in triple notation (Turtle format).

Furthermore, some semantic annotation systems perform ontology population, which means not only annotating documents with respect to an existing ontology resulting in semantic documents but creating new instances from the textual source [47]. For example, the ball bearing from the example above is instantiated as a new knowledge object within a knowledge graph. One advantage of building knowledge graphs from textual sources is the direct link between mentions of knowledge objects within a source and the capability of generating structured data from those mentions, even if the facts about a knowledge object origin from different sources. A schematic architecture of a semantic knowledge base, which consists of a domain ontology on schema-level, as well as a knowledge graph that holds the data about knowledge objects, is shown in Figure 5. An example of structured

information is given for the knowledge object “thrust ball bearings”, once as its use in a tribological test setup and once as a rolling bearing with its specification.

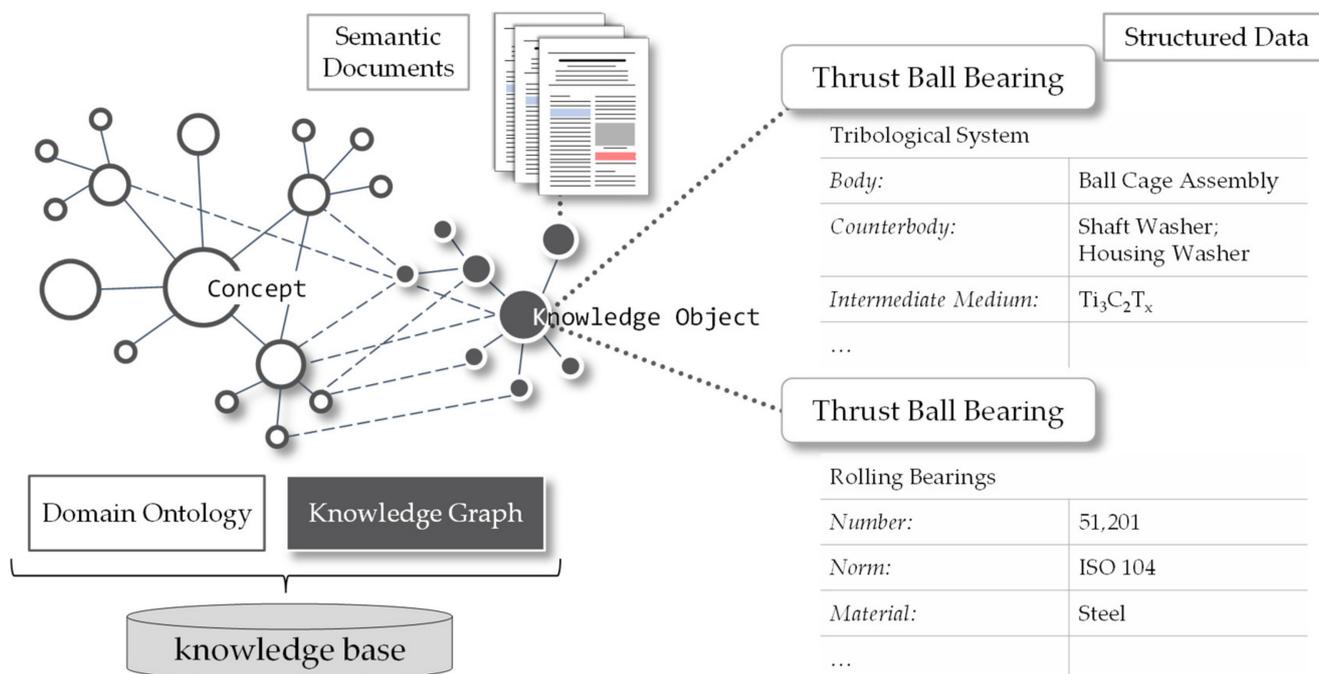


Figure 5. Schematic architecture of a semantic knowledge base, consisting of schema-level ontologies and a knowledge graph containing knowledge objects as structured data and mentions of knowledge objects from semantic documents.

Thus, different information from various sources is linked for an object of the knowledge graph. Moreover, the original textual sources are also linked nodes within the graph. Semantic annotation can be performed manually, semi-automatically, or automatically. Thereby, the semi-automatic approach is preferred since manual annotation is time-consuming and automatic approaches can lead to unreliable information within the resulting knowledge graph [47].

2.4. Natural Language Processing

A semi-automatically semantic annotation process is often conducted by methods from NLP. The main challenge of NLP is the representation of contextual nuances of human language, since the same matter can be described utilizing different wording and the same word can be used for different meanings depending on the context. Therefore, enabling machines to understand and process natural language demands the provision of a machine-readable model of language. However, Goldberg [48] describes a challenging paradox in this context: Humans are excellent in producing and understanding language and are capable to express and interpret strongly elaborated and nuanced meaning of language. In contrast, humans struggle at formally understanding and describing the rules, which govern our language [48]. Rules in this context are not only referred to syntax and grammar, but also to contextual concerns. For example, considering a classic NLP task of document classification into one of the four categories metals, fluids, ceramics, or polymers. Human readers categorize documents relatively easy into those topics guided by the words used within a publication but writing up those implicitly applied rules for categorization is rather challenging [48]. Therefore, machine learning models are trained to learn vectorized text representations from examples, which are suited input formats for NLP downstream tasks (e.g., document classification). The classic preprocessing steps for generating those text representations from a document corpus are summarized in Figure 6.

Almost any analysis of natural language starts with splitting the documents (e.g., plain text, charts, figures), removing noise (e.g., references, punctuation) and normalization of word forms [49]. Subsequently, the plain text is further split into minimal entities of textual representation, the tokens, on word- or character-level. Since ML models assume some kind of numerical representation as input, the tokens are replaced by their corresponding IDs [50]. If a text is split into tokens on word level, the question arises, what counts as a word. To answer this question, morphology deals with word structures and the minimal units a word is built from, such as stems, prefixes and suffixes. Those minimal units are important, if a tokenizer has to deal with unknown words (meaning words, which were not within the training corpus) [48]. Tokenizers like WordPiece [51] represent words as subword vectors [49], e.g., “nanosheets” can be separated in the subwords “nano” and “sheet” and the plural-ending “-s”. However, the tokens are further transferred in so-called embeddings, which are an input representation a ML or deep learning architecture can handle for NLP tasks. An embedding is a representation of the meaning of a word; thus, they are learned under the premise, that a word with the same meaning has a similar vector representation [49]. A distinction is made between static embeddings and contextualized embeddings. One quite popular static word embedding package is Word2Vec [52,53]. A shortcoming of those static embeddings is that polysemantic is not properly handled since one fixed representation is learned for each word in the vocabulary even if a word has a different meaning in different contexts [49,54].

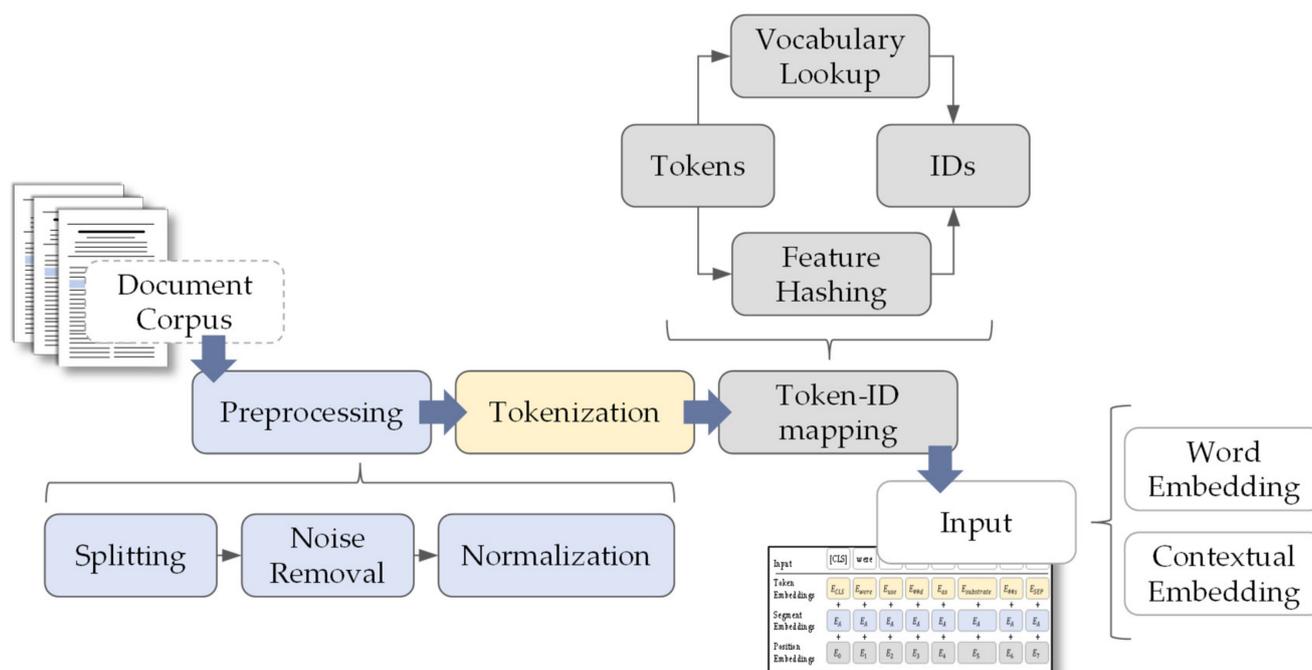


Figure 6. Preprocessing steps to generate embeddings from text as input to NLP downstream task. Redrawn and adapted from [50].

Therefore, contextualized (dynamic) embeddings provide different representations of each word based on other words within the sentence. State of the art representatives are ELMO (Embeddings from Language Models) [55], GPT & GPT2 (Generative Pre-Training) [56] and BERT (Bidirectional Encoder Representations from Transformers) [33], which are also referred as pre-trained language models. BERT is a multi-layer bidirectional transformer encoder [33,57], which is provided in a base version with 12 layers and large version with 24 layers. Most of the recent models for NLP tasks are pre-trained on language modeling (unsupervised) and fine-tuned (supervised) with task-dependent labeled data [58]. Thus, those models are trained to predict the probability of a word occurring in a given context [48]. BERT is pretrained on large amount of general-purpose texts from

BooksCorpus and English Wikipedia, which resulted in a training corpus of about 3300 M words [33]. Devlin et al. [33] differentiate BERTs pre-training from the other mentioned models, consisting of two unsupervised tasks: masked language modeling (LM) and next sentence prediction (NSP) (see also [59] for further information on BERTs pre-training). Fine-tuning BERT for downstream tasks, like Question Answering (QA) or Named Entity Recognition (NER), the same architecture is used apart from the output layer (see Figure 7). The input layer consists of the tokens (Tok 1...Tok n). The special token [CLS] signs the starting point of every input and [SEP] is a special separator token. For instance, question answering pairs can thus be separated within the input [33]. The contextual embeddings ($E_1 \dots E_n$) further result in the final output ($T_1 \dots T_n$), after being computed through every layer resulting in different intermediate representations (T_{rm}). For more information on Transformer architectures, the interested reader is referred to [60]. There are different extensions of the original model of BERT, which are specialized for certain downstream tasks or domain terminologies. The SciBERT model [61] is pre-trained on scientific papers improving the performance of downstream tasks with scientific vocabulary. BioBERT [32] is pre-trained on large-scale biomedical corpora and improves the performance of BERT especially in biomedical NER, relation extraction and QA. Furthermore, SpanBERT [62] is a pre-training approach, which is focused on a representation of text spans instead of single tokens. Both pre-training tasks from the original BERT are adapted for predicting text spans instead of tokens, which is especially useful in relation extraction or QA.

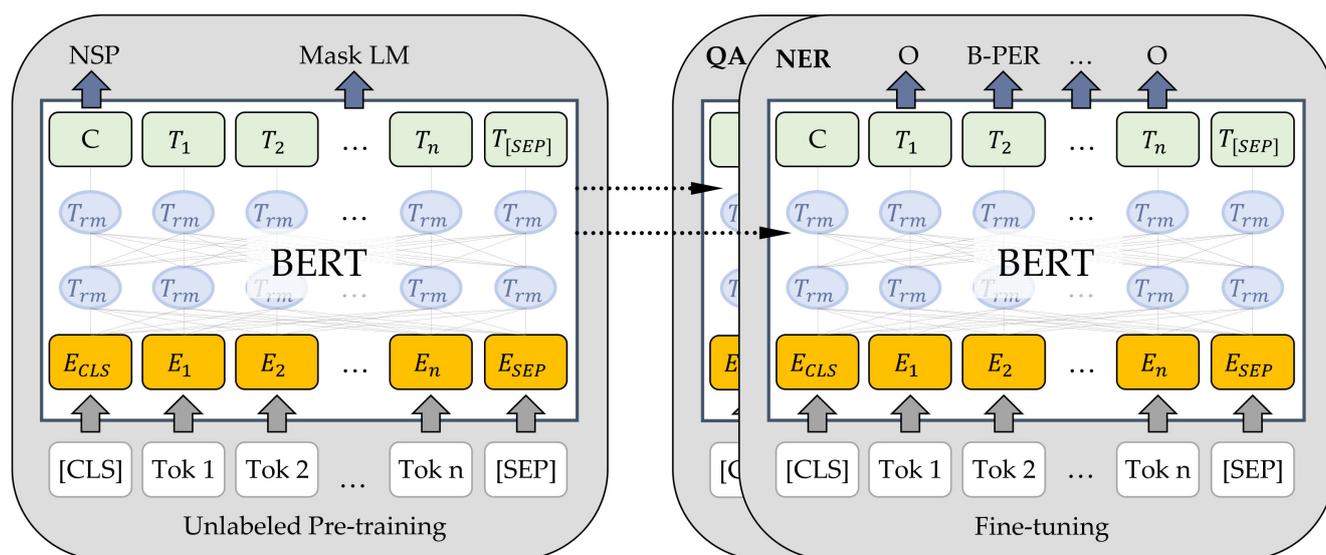


Figure 7. BERT pre-training and fine-tuning procedures using the same architecture for both. Only the output layer differs depending on the downstream task e.g., NER, QA. Redrawn and adapted from [33].

2.5. NLP Downstream Tasks

Information Extraction (IE) is a task of obtaining structured data from unstructured information, e.g., embedded in textual sources, by recognizing and extracting occurrences of concepts and relationships among them [49,63]. IE is often used to build knowledge graphs from textual representations (e.g., DBpedia), since those can be queried and are a common way of presenting information to users [49]. IE and semantic annotation (see Section 2.3) are often combined, since both share the subtask of NER. NER is a sequence-labeling task to recognize and tag words or phrases usually like “Person” (B-PER), “Location” (B-GEO) or “Organization” (B-ORG) within textual data. A named entity can be anything, which has a proper name, thus can be distinguished from other objects [49]. Therefore, NER is often based on a specific domain vocabulary, e.g., in biomedicine [32,64]. Moreover, relation extraction is also a subtask of IE in the context of building knowledge graphs and mainly

deals with the extraction of binary relations like child-of or part-whole relationships used within taxonomies, ontologies and knowledge graphs [49]. IE can be used for template filling, meaning recognizing and filling a pre-defined template of structured data from the unstructured sources (cf. Figure 5) [49]. Question Answering (QA) is a task of information retrieval, but with a query, which is a question in natural language and a response as an actual answer [63]. QA is often used within Chatbots of customer services or within virtual speech assistants (e.g., Amazon Alexa or Apple Siri). The main difference from classic retrieval operations is the form of asking questions in natural language instead of formal database queries and the retrieval of a precise answer to the question instead of document retrieval. Therefore, QA can be exploited for generating structured data and template filling.

3. Semantic Annotation Pipeline

The developed semantic annotation pipeline consists of three separate modules and a graphical user interface (GUI). The modules communicate via REST API. Due to the modular architecture, the single module can be exchanged and thus the pipeline can be adapted to particular applications. The pipeline is trained for annotating and extracting information from tribological publications with the scope of model tests (cf. Section 2.1).

3.1. Document Extraction

The first module (Figure 8) performs the preprocessing step introduced in Section 2.4 (Figure 6). Since the source format (PDF) is not suited for further processing, plain text is extracted. By parsing through the documents, elements like figures, charts or tables are also detected.

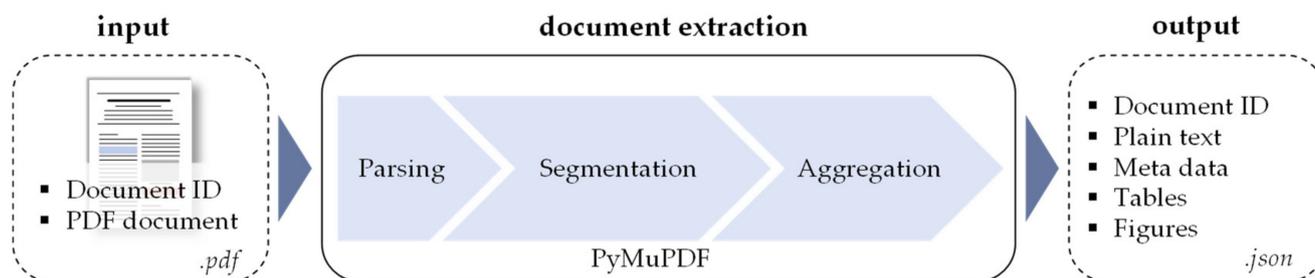


Figure 8. Document extraction module.

Besides detecting non-textual elements, the document is segmented into its paragraphs (e.g., abstract, introduction etc.), since—depending on the IE purpose—the relevant information may be provided mainly in a certain paragraph. For example, the introduction section usually contains information about the aim of the investigation, while the results section further provides a description of the outcome. Parsing is performed based on syntactical rules and pattern matching, e.g., indentations, blank spaces or different fonts, can be used as indicators for the detection process. Besides the content of the publication, meta data about the document (e.g., author, DOI, date, publishing information) is extracted. The last step is the aggregation of the previously segmented elements into JSON (Java Script Object Notation), which is a common and platform independent data sharing format. The document extraction module is implemented using the PyMuPDF Library [65].

3.2. Document Annotation

The document annotation module (Figure 9) performs the actual annotation process. The module expects the files in JSON format from the document extraction module as input and is capable of annotating plain text and table data. The annotation module uses the Flair library [66] and the embeddings of the NER model are trained on the tribological annotation categories displayed in Figure 10. Within a parameter study, embeddings from BERT (Base), SciBERT and SpanBERT were tested against each other. Thereby, SpanBERT

were chosen, since those have shown the best results with an F1 (micro) score of 0.8065 and an F1 (macro) score of 0.8012.

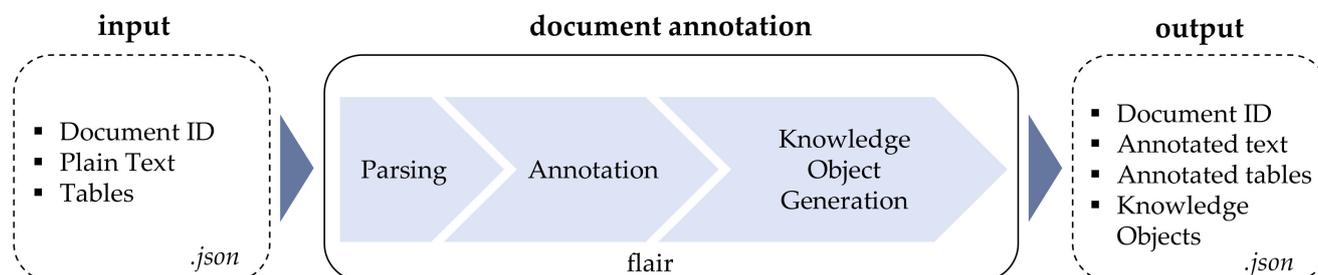


Figure 9. Document annotation module.

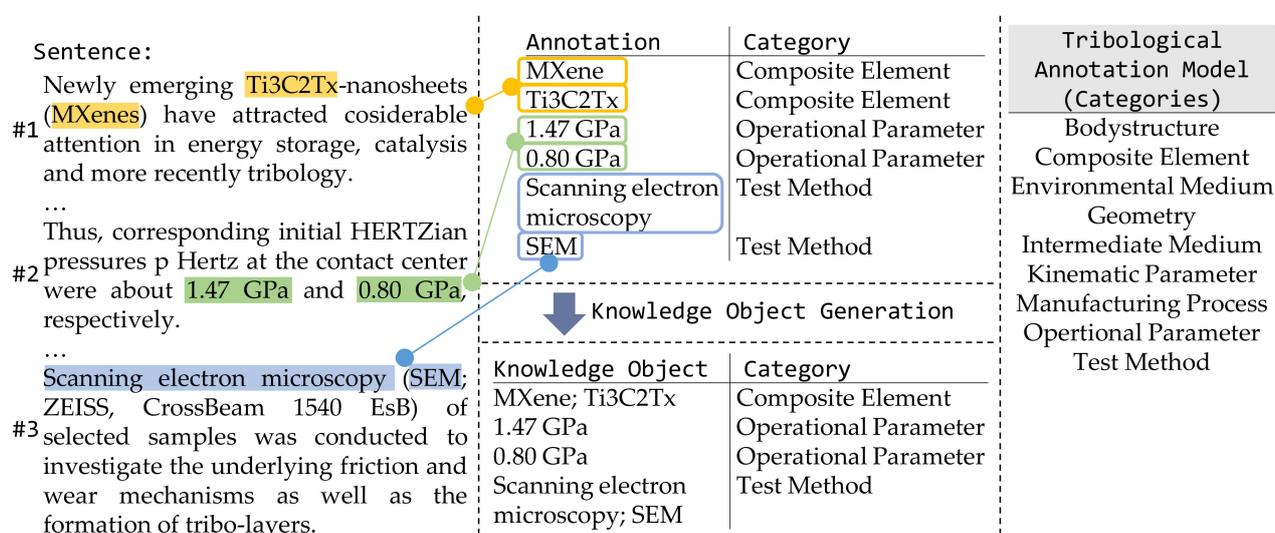


Figure 10. Example of semantic annotation and knowledge object generation within the document annotation module and annotation categories for example sentences from [67].

The annotation step within the module recognizes entities of the tribological categories. An example is shown in Figure 10. The inputs are three different sentences (plain text), which are parsed. Then, entities of the different categories are annotated. In a second step, the annotations are aggregated to knowledge objects, thus for instance the two recognized entities MXene and Ti₃C₂T_x refer to the same knowledge object (Figure 10). Due to the knowledge object generation, different terms used to describe the same entity within a text are aggregated to a single object. The generation of knowledge objects is mainly based on identifying acronyms and synonyms. The identified character strains are then compared. For character strains, which go beyond four, a fuzzy comparison using the Fuzzy-Wuzzy Library (<https://pypi.org/project/fuzzywuzzy/>, accessed on 14 December 2021) is conducted, which calculates the Levenstein distance to compare two-character strains. Output data from the document annotation module is again streamlined in JSON format and contains the annotated text and table data as well as the aggregated knowledge objects.

3.3. Document Analysis

The document analysis module (Figure 11) is a QA system that extracts answers from the text to questions about tribological model tests to create triples from the document. The QA system is built on the PyTorch framework (<https://pytorch.org/>, accessed on 14 December 2021) using a SciBERT-Model from the Hugging Face library (<https://huggingface.co/>, accessed on 14 December 2021).

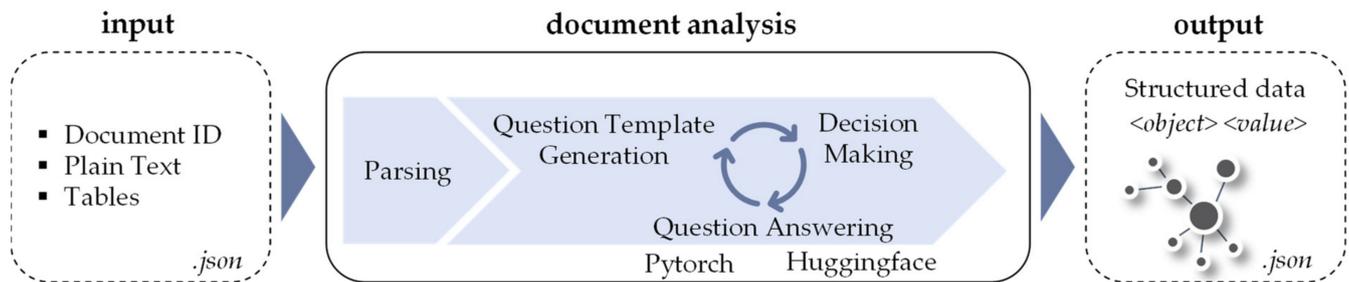


Figure 11. Document analysis module.

The BERT model is fine-tuned by question-answer pairs. Question templates (Figure 12) are generated, which contain the questions for extracting knowledge objects from the text. Those templates determine the structure of the information, which should be extracted from the text. This means, the question templates can be customized depending on the extraction task. The decision maker is an intermediate aggregation step containing multiple redundancies, which ensures higher reliability of an extracted answer. Therefore, the question template contains the same question rephrased several times. Furthermore, the answer space is restricted by using regular expressions (Regex) to define an expected answer pattern and by specifying an entity type (tribological category) of the extracted answer. The final result is an ID of a knowledge object and its textual annotation, for the case a knowledge object can be assigned. Otherwise, the textual passage is extracted as answer to the question.

```

testing_duration":{
  "question":[
    Question(s)    "What was the test duration for the test?",
                  "How many minutes went the test?",
                  "How long did the test go on?"
  ],
  Answer space   ],
  (Regex)        "expectedAnswerSpace":[
                  "([0-9]{0,5}\\.|)[0-9]+ *(min|h|s)"
  ],
  Search space   "preSearchSpace":[
                  " minutes ", "min ", " hour", " second",
                  "sec ", " h ", "time"
  ],
  Entity subcategory "specific_questionType":"TESTING_DURATION",
  Entity category   "broader_questionType":"OPERATIONAL_PARAMETER",
                  ... ]
}
  
```

Figure 12. Question template example for the extraction of the testing duration of an experiment.

4. Implementation and Evaluation

4.1. Web-Application

For testing the capability, the pipeline can be assessed through a web interface (see Data Availability Statement). The three modules communicate via REST API with the GUI. Publications have to be provided in PDF-format and can be delivered by drag and drop to the first module (1). As indicated in Figure 13, a manual control and adaptation step is integrated between the three separate modules. Thus, the acquisition of structure works in a semi-automatic manner with human supervision.

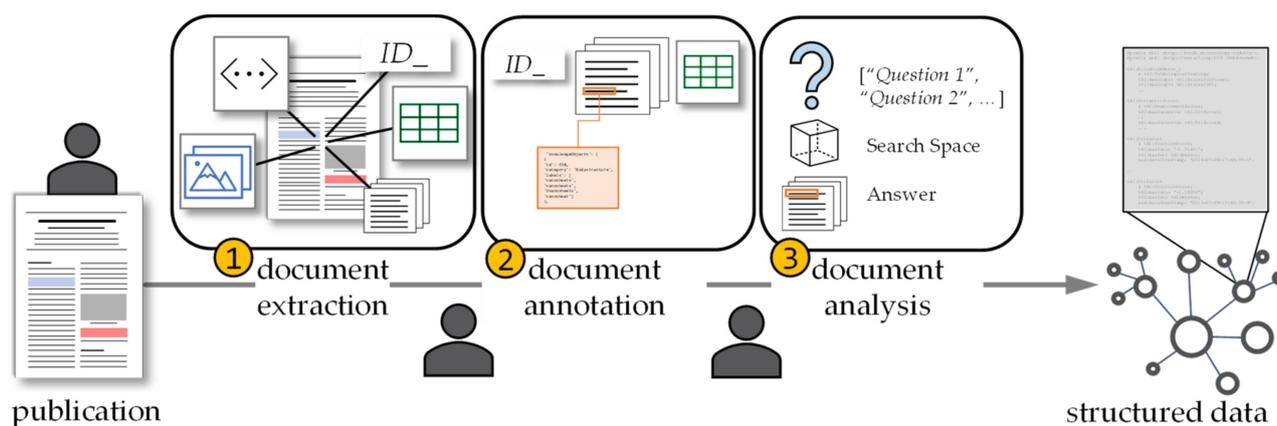


Figure 13. Process of document annotation via the web-interface.

The output from the first module is the extracted text and other entities (e.g., figures, tables, metadata) from the PDF. The text itself is split into chapters, paragraphs and sentences, which serve as input to the next modules. The user check enables adaptation and correction of the automatically generated output from the module (e.g., adaption of the paragraph separation). Subsequently, the pipeline can be continued via the GUI and the JSON-data produced by the first module forwarded to the document annotation module (2), which performs NER and knowledge object generation. The output from the second module can also be checked by the user after the automatic annotations are generated. The NER process thereby performs the annotation part while the knowledge objects are the semantic output of the annotation pipeline (see Figure 5 in Section 2.3 for the role of knowledge objects within a semantic knowledge base). The check especially contains proof of annotations and the aggregation to knowledge objects. Next, the pipeline is continued to the final context analyses via the document analysis module, which performs the QA and generates the structured data as final output.

4.2. Resulting Knowledge Graph

The output from the document annotation module is a linked data structure, containing the aggregated knowledge objects related to the mentions within paragraphs and tables of the publication. Therefore, the output can be visualized as a knowledge graph containing the structured data annotated within the respective publication. This is exemplified in Figure 14 for a representative publication [67]. The generated knowledge graph is a complex network of nodes and relationships. Thereby, the size of the nodes corresponds to the number of mentions of a knowledge object within the publication. This means that a knowledge object with only one mention has the minimum size while the size increases with the number of mentions. In this way, the important knowledge objects within the publication can be easily identified within the graph.

While the output graph from the document annotation module is particularly suited to identify the main topics of the analyzed publication in form of the most often mentioned knowledge objects, the output from the document analysis module is a graph generated from the question templates (Figure 15). The knowledge graph contains the identified and correctly classified answers (triangles) given by the decision maker. Thereby the triples are generated from the schema provided by the tribAIn ontology [29]. Thereby, the excerpts of the knowledge graph in Figure 15 refer to the example questions introduced in Section 2.2 regarding the tested variables (1) and the calculated wear rate (2). The generated linked data combined with an ontology provides a formally and semantically unambiguous representation which can be queried, filtered and further processed.

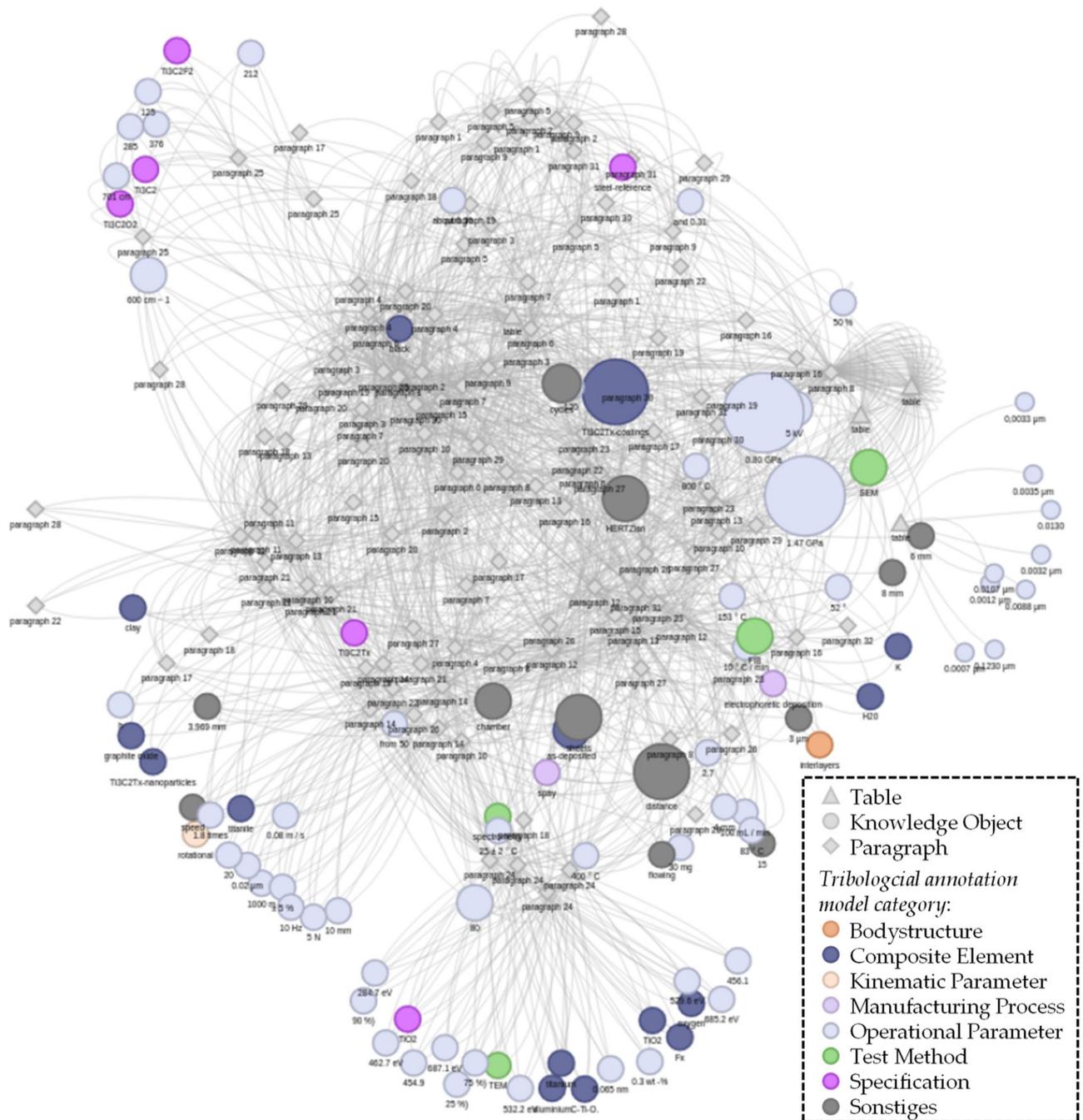


Figure 14. Schematic visualization of the resulting knowledge graph from the document annotation module for the processed representative publication [67].

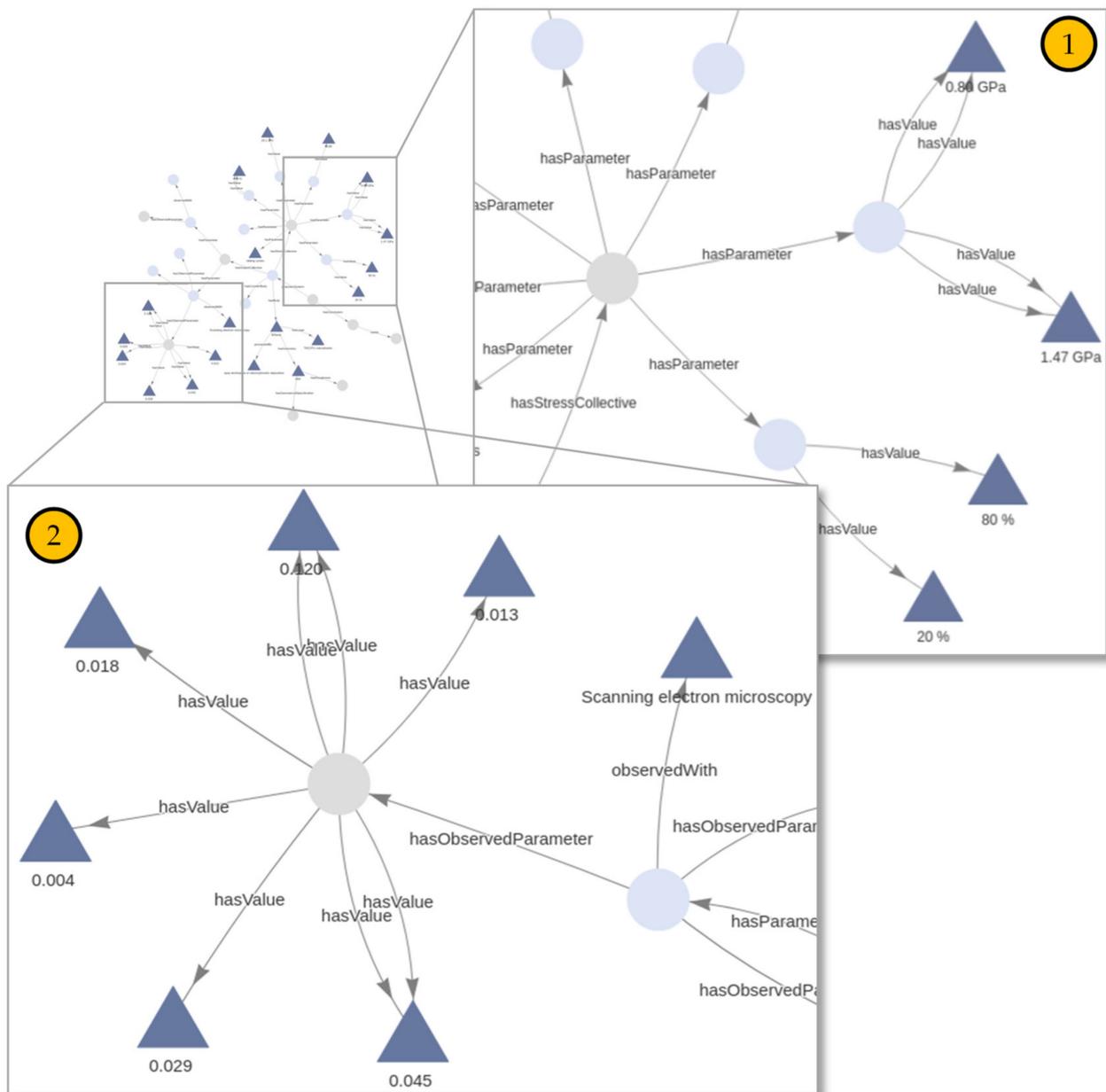


Figure 15. Schematic of the resulting knowledge graph from the document analysis module for the processed representative publication [67] with a detailed view on the varied operational parameters (1) and the outcome measurements of the wear rate (2).

4.3. Evaluation

For evaluation, five documents from a pool of existing publications on model-tests were used for an initial performance test of the pipeline:

- **Doc#1:** Marian et al., 2020 [67]
- **Doc#2:** Mekingwe et al., 2019 [68]
- **Doc#3:** Li & Xu 2018 [69]
- **Doc#4:** Wang et al., 2018 [70]
- **Doc#5:** Byeong-Choon & In-Sik 2017 [71]

The documents differentiate in length and format, for instance number of columns. Furthermore, Doc#5 is kind of defect since the PDF contains invisible text overlaps. Thereby, the three modules were evaluated separately depending on the evaluation aim. Since semantic annotation is not a common task within the domain of tribology, standard test

documents, which are widely accepted for performance measurement of NLP tasks, do not exist. Therefore, the test documents were manually annotated for the special purpose of evaluating the pipeline introduced within this contribution. The document extraction module was analyzed with respect to its quality in extracting and separating text and other elements like figures and tables from the PDF documents. This resulted in a comparison of the data from the extraction module against the ground truth (GT) for text, figures and tables. This shows if the system works as it is intended. The smaller the deviations from GT, the more reliable is the PDF extraction. Within Table 1, the reliability of text extraction is assessed against the criteria, if chapters, paragraphs, sentences, words and chars are correctly detected and separated. The deviations from GT are relatively small for Doc#1 (chapter −12.5%; paragraph −8.6%, sentence −7.3%, word −7.3%, char −9.8%), Doc#2 (chapter 37.5%; paragraph −26.7%, sentence −3.0%, word −10.0%, char −18.2%), Doc#3 (chapter 7.1%; paragraph 11.5%, sentence −2.4%, word −1.3%, char −5.4%) and Doc#4 (chapter 27.3%; paragraph −22.2%, sentence 23.6%, word −5.0%, char −10.9%), while the deviation is substantially higher for Doc#5 (chapter 240%; paragraph 400%, sentence 149%, word 132%, char 118%). Those high deviations can be attributed to the defect PDF, which contains embedded textual and other elements, which overlap the intended content of the document. The results of the figure extraction analysis are shown in Table 2. Almost all figures within the test set were correctly detected. Only one figure was partly incorrect extracted in Doc#2 and two figure areas were incorrectly recognized in Doc#4. However, all figures were correctly extracted within the defect PDF Doc#5. Thereby, 14 additional figures were identified, which is due to the overlaid elements within the PDF. The extraction of tables seems also reliable since the majority of tables are correctly recognized (see Table 3). An exception is within Doc#1, which can be attributed to the table being rotated within the publication. This shows that the first module depends on the quality and regularity of the input files. Since the module provides a manual check, small deviations from the expected output can easily be fixed via the GUI.

Table 1. Quality of text extraction regarding the extracted chapters, paragraphs, sentences, words, chars and if an abstract was detected (true/false). The GT is given in the brackets.

Document	Doc#1	Doc#2	Doc#3	Doc#4	Doc#5
Chapter	14 (16)	11 (8)	15 (14)	14 (11)	31 (9)
Paragraph	32 (35)	11 (15)	29 (26)	14 (18)	40 (18)
Sentence	179 (193)	65 (67)	124 (127)	68 (55)	237 (95)
Word	4213 (4547)	1304 (1449)	2898 (2937)	1205 (1276)	5192 (2241)
Char	26,127 (28,881)	7497 (9160)	17,119 (18,089)	6998 (7759)	30,928 (14,198)
Abstract	true	true	true	true	true

Table 2. Quality of figure extraction regarding detected figures, incorrectly detected figure area and additional extractions. The GT is given in the brackets.

Document	Doc#1	Doc#2	Doc#3	Doc#4	Doc#5
Figure	12 (12)	5 (5)	12 (12)	5 (5)	12 (12)
Incorrect area	0	1	0	2	0
Additional Figure	0	0	0	0	14

Table 3. Quality of tables extraction regarding detected tables, additional extractions and correct number of cells. The GT is given in the brackets.

Document	Doc#1	Doc#2	Doc#3	Doc#4	Doc#5
Table	0 (1)	1 (1)	0 (0)	0 (0)	1 (1)
Additional Table	0	0	0	0	0
Incorrect cells	-	0	-	-	0

The document annotation module was evaluated with respect to the ability of NER and knowledge object generation. Three language models (BERT, SciBERT and SpanBERT) were trained with the hyperparameters shown in Table 4, which were an outcome of a previously conducted parameter study. Thereby, an RNN (recurrent neuronal network) architecture was used with one layer and a hidden size of 128. Dropout [72,73] is a method to reduce overfitting by deactivating a number of neurons randomly from the neural network. The learning rate defines the step size of the optimization and thus controls how quickly the model learns the given problem. The batch size specifies the number simultaneously evaluated examples. Since the used language models have already been pre-trained on large-scale general language data (cf. Section 2.4), the training includes only fine-tuning, which is computationally less expensive. The training of the models took about 20 to 30 min each on a NVIDIA RTX 2070 and 8 GB RAM.

Table 4. Hyperparameters for training language models.

Language Model	RNN Layers	Hidden Size	Dropout Rate	Learning Rate	Mini Batch Size
BERT	1	128	0.0479	0.1	32
SciBERT	1	128	0.0020	0.1	32
SpanBERT	1	128	0.1454	0.15	32

Micro and Macro F1 scores were calculated to select the best of the three models for the recognition task. Therefore, the five documents were manually annotated due to the tribological annotation model categories. For every category the precision, recall and F1 score were calculated three times for each of the trained language models with regard to the manual annotations (see Table 5). The test set contained 986 annotated sentences for the tribological annotation model categories already introduced in Figure 10.

Table 5. Precision (P), Recall (R) and F1 score for each tribological annotation model category.

Category	Score	BERT			SciBERT			SpanBERT		
		1	2	3	1	2	3	1	2	3
Body structure	P	0.8000	0.7037	0.7692	0.7037	0.6667	0.7097	0.7600	0.7600	0.7241
	R	0.8333	0.7917	0.8333	0.7917	0.8333	0.9167	0.7917	0.7917	0.8750
	F1	0.8163	0.7451	0.8000	0.7451	0.7407	0.8000	0.7755	0.7755	0.7924
Composite element	P	0.7833	0.7241	0.7272	0.7846	0.7286	0.7429	0.7576	0.8030	0.7812
	R	0.7833	0.7000	0.8000	0.8500	0.8500	0.8667	0.8333	0.8833	0.8333
	F1	0.7833	0.7118	0.7619	0.8160	0.7846	0.8000	0.7936	0.8412	0.8064
Environmental medium	P	1.0000	0.8000	1.0000	1.0000	1.0000	1.0000	0.8333	0.8333	1.0000
	R	0.8000	0.8000	0.8000	0.8000	0.6000	0.6000	1.0000	1.0000	0.8000
	F1	0.9000	0.8000	0.9000	0.9000	0.8000	0.8000	0.9167	0.9167	0.9000
Geometry	P	0.9375	0.8824	0.8750	0.7500	0.8235	0.8235	0.8824	0.8889	0.8325
	R	0.7895	0.8950	0.7368	0.6316	0.7368	0.7368	0.7895	0.8421	0.7368
	F1	0.8572	0.8887	0.8000	0.6857	0.7777	0.7777	0.8334	0.8649	0.7817
Intermediate medium	P	0.8462	0.7143	1.0000	0.9286	1.0000	0.8571	1.0000	1.0000	1.0000
	R	0.6111	0.5556	0.6111	0.7222	0.7222	0.6667	0.6667	0.6667	0.6667
	F1	0.7097	0.6250	0.7586	0.8125	0.8387	0.7500	0.8000	0.8000	0.8000
Kinematic parameter	P	0.6667	0.6667	0.6364	0.7273	0.6923	0.6923	0.8000	0.8000	0.7273
	R	0.8000	0.8000	0.7000	0.8000	0.9000	0.9000	0.8000	0.8000	0.8000
	F1	0.7273	0.7273	0.6667	0.7619	0.7826	0.7826	0.8000	0.8000	0.7619

Table 5. Cont.

Category	Score	BERT			SciBERT			SpanBERT		
		1	2	3	1	2	3	1	2	3
Manufacturing process	P	0.6250	0.5000	0.6250	0.8182	0.6923	0.7143	0.7143	0.7500	0.6667
	R	0.4545	0.5455	0.4545	0.8182	0.8182	0.9091	0.4545	0.5455	0.5455
	F1	0.5263	0.5218	0.5263	0.8182	0.7500	0.8000	0.5555	0.6316	0.6000
Operational parameter	P	0.7222	0.7571	0.7647	0.7656	0.7937	0.8197	0.7059	0.7500	0.7424
	R	0.8525	0.8689	0.8525	0.8033	0.8197	0.8197	0.7869	0.8361	0.8033
	F1	0.7820	0.8092	0.8062	0.7840	0.8065	0.8197	0.7442	0.7907	0.7717
Specification	P	0.7143	0.7500	0.7368	0.6842	0.8235	0.7647	0.6190	0.7778	0.7778
	R	0.7895	0.7895	0.7368	0.6842	0.7368	0.6842	0.6842	0.7368	0.7368
	F1	0.7500	0.7692	0.7368	0.6842	0.7777	0.7222	0.6500	0.7567	0.7567
Test method	P	0.8947	0.8571	0.8571	0.8500	0.8095	0.8182	0.8889	0.8421	0.8497
	R	0.8947	0.9474	0.9474	0.8947	0.8947	0.9474	0.8421	0.8421	0.8497
	F1	0.8947	0.9000	0.9000	0.8718	0.8500	0.8781	0.8649	0.8421	0.8497

The resulting F1 scores are summarized in Table 6 for BERT, SciBERT and SpanBERT, which were each calculated in triplicate. As mentioned before, SpanBERT featured the best scores within the second run, which may be due to the annotated entities referring to the tribological categories, that are often spans of words instead of single tokens (e.g., “Scanning electron microscopy”).

Table 6. Evaluation and selection of the NER model. F1 scores for BERT, SciBERT and SpanBERT.

	BERT			SciBERT			SpanBERT		
	1	2	3	1	2	3	1	2	3
F1 (micro)	0.7823	0.7570	0.7782	0.7847	0.7905	0.7992	0.7702	0.8065	0.7879
F1 (macro)	0.7736	0.7443	0.7645	0.7868	0.7859	0.7880	0.7726	0.8012	0.7851

The annotations generated through NER were further aggregated to knowledge objects within the document analysis module. The resulting number of aggregations is shown in Table 7. Annotations are considered incorrectly aggregated if at least two annotations are assigned to the same knowledge objects, although they do not belong together (false positive). Furthermore, if at least two annotations which belong to a knowledge object are not aggregated, they count as false negative. This criterion captures the reductivity of the knowledge object generation while the counts of correctly and incorrectly aggregated annotations provide an insight into the precision of the generation. A precision of 89.5% is reached for the test pool while the recall is about 84.4%. This can be considered as sufficient for the quality of knowledge object generation.

Table 7. Evaluation of knowledge object generation containing the number of annotations, of all knowledge objects as well as correctly aggregative (true positive), incorrectly aggregated (false positive) and not aggregated (false negative) knowledge objects.

Document	Annotations	Knowledge Objects	Correctly Aggregated	Incorrectly Aggregated	Not Aggregated
Doc#1	944	236	59	12	11
Doc#2	296	68	21	1	11
Doc#3	609	98	57	4	2
Doc#4	323	75	32	5	0
Doc#5	400	70	36	2	14
Σ	2572	547	205	24	38

Finally, the document analysis module was evaluated due to its quality of answering the questions from the templates. The criteria for assessing the quality were grouped to the quality of question answering itself and if the decision maker prefers the right answer. The final results over all questions are shown in Table 8. The GT is counted, if at least one answer within the text can be given to the question. The criteria for question answering itself are split into the cases if the expected answer is found in text and/or if at least one additional answer was found independent of the expected answer. The need for the decision maker can be seen from the fact that additional answers besides the expected one were found for all documents. The criteria for the decision maker were thereby split into the cases if the correct answer was preferred by the decision maker, if an incorrect answer was preferred, or if no answer was found or preferred. When the text contains at least one correct answer (GT), the question answering itself found the correct answer with a probability of 60.4%. The decision maker found the correct answer with a probability of 62.3%. At this point it should be noted that the quality of answers is highly influenced by the question templates. This means what questions are asked of the publication to get a desired answer.

Table 8. Evaluation results of all answers to question templates regarding the input parameters (e.g., kinematical parameters), structural information of the tribological system (e.g., geometry) and output parameters (e.g., friction and wear).

Document	GT	Question Answering		Decision Maker		
		Answer Found	Additional Answers	Correct Answer	Incorrect Answer	No Answer
Doc#1	26	16	15	13	1	10
Doc#2	20	13	18	7	11	9
Doc#3	20	9	13	7	4	13
Doc#4	16	8	8	7	4	16
Doc#5	24	12	14	9	6	8
Σ	96	58	68	43	26	56

5. Discussion

In the context of the “knowledge reengineering bottleneck”, we introduced a semantic annotation pipeline to semi-automatically streamline the knowledge aggregation from publications within the domain of tribology. The inputs for the pipeline are publications of experimental investigations from the domain of tribology and in particular experiments of the category model test. The output is structured and linked data in form of json-files, which can be visualized as graphs (cf. Section 4.2). The pipeline is built on state-of-the-art language models and NLP techniques and was evaluated on five representative documents. Since NLP is not in common use within the domain of tribology, there are no datasets and standard documents for training and evaluating language models. This limits the significance of the performance test conducted within this contribution since a Gold Standard accepted by the community is missing and the pipeline cannot be compared to similar projects. However, as we work with standard language models, which are approved to be reliable within NLP communities and we conducted a first evaluation of our fine-tuned models by manually annotating five representative documents, some assertions can still be made about the current performance. Thereby, the document extraction (module 1) has shown reliable performance on different structured and formatted publications under the premise that the provided PDFs are not defect. This was substantiated by one tested PDF document, which contains invisible overlays and therefore shows high deviations from the GT in comparison to the other documents. The PDF extraction is always a critical step within NLP processes as it depends on the quality of the PDF and accessibility of the textual and other entities within the PDF. This is one reason for the modular structure of the pipeline. The PDF extraction is only required if the input publications are in the form of PDF format (which is a common format for textual documents). Since nowadays publications are frequently available online as well, the accessibility of textual data from HTML-Websites via an API is easier when the access is provided by publishers. Therefore,

PDF extraction is a pragmatic approach to access the textual information from publications. The annotation process (module 2) is performed using the SpanBERT language model, which shows remarkable high F1 scores. The NER model introduced within this publication is currently limited to publications on model tests (without claiming completeness), since those are well structured and mostly standardized. To our best knowledge, NER tagsets or language models itself as available for example in the domain of biomedicine (e.g., BioBERT) so far do not exist for the domain of tribology. In the future, the development and training of tribological language models can therefore improve the performance of applications in NLP within the domain of tribology. Furthermore, knowledge object generation is only a first step in named entity linking.

We discussed the role of knowledge objects within semantic knowledge bases within Section 2.3. The knowledge objects here are an aggregation of annotations from the document extracted by the pipeline. However, successful semantic knowledge sharing is usually community driven within a domain (e.g., OBO foundry). An established knowledge graph within the domain of tribology containing knowledge objects can therefore be extended with aggregated objects from the annotation module. Further established knowledge objects can also be enriched by the annotations. Thus, information about entities of interest in the domain of tribology can be semi-automatically acquired. Within the last module, we exploited QA to generate structured output from the unstructured and annotated data. The templates contain questions referring to tribAIIn-ontology (e.g., questions about input parameters, the tribological system structure and output parameters). Overall, the QA system showed plausible answers to the tested question templates. During evaluation, we recognized a frequently appearing misconduct of the decision maker, which often could not differentiate the properties of the body and the counterbody. On the one hand, this can be attributed to an insufficient differentiation within the textual description and on the other hand to the question generation process within the QA system. The experiences with the QA module further led to two major perceptions in the context of extracting information from tribological publications. First, analyzing publications by a QA system can be exploited for a quality check and improvement of standardization of the description of experimental studies and outcomes. Thereby, question templates can be specified as a check list, what a sufficient description of experimental studies and results should contain to enable understanding and reproduction of the results. Second and relating thereto, the question templates itself have to be carefully designed to gain an answer and aggregate structured data from texts. Therefore, analyzing the publication practices and further the research practices of tribologists can give interesting insights for improving knowledge and data aggregation within the domain. However, the pipeline is intended to be human supervised, since trust is a critical issue especially within neural NLP processes which generate output without explanations of the process itself. This is the second reason for the modular architecture. The output from every module can be checked and adapted before continuing with the pipeline. This is especially important if automatic extraction is used to extend semantic knowledge bases or aggregate structured data for further processing. Besides the quality and trust of the results from the pipeline, another important issue is the computational costs. As mentioned within Section 4.3, the training of the language models took less than an hour (20 to 30 min) for each model. The low computational costs are due to the currently available pre-trained language models, which merely must be fine-tuned to be tailored to a specific domain. The execution of the annotation further only takes a few seconds. Therefore, the pipeline can be considered as very efficient compared to manual annotation.

6. Conclusions

Sharing knowledge in publications has a long tradition in scientific research since this is the elemental way of consuming and communicating information and knowledge by human scientists. Within the domain of tribology, the vast amount of available information overcomes the cognitive capacity of humans in terms of efficient aggregation and processing.

Therefore, AI provides a lot of potential to support scientists by handling this flood of information. Nevertheless, the way of knowledge sharing within tribology is still mainly based on publications and thus human focused. Descriptions in natural language are vague and insufficient from a formalization perspective. This phenomenon is due to an intended human consumer, for whom the provision of formal sufficient information would result in far too long publications and re-reading the same information over and over again. In contrast, machines need a formal and explicit model for aggregating and processing information. Therefore, if the available amount of information overwhelms the capacity of human processability, the question arises if we better should create representations for sharing information with an AI instead of humans in the future? The answer is: Not necessarily. As pointed out by Gruber [74], “the purpose of AI is to empower humans with machine intelligence”. This is referred to as “humanistic AI”, an artificial intelligence designed to meet human needs by collaborating with and augmenting people. In terms of an AI empowered tribological knowledge sharing, we introduced a semantic annotation pipeline towards generating knowledge graphs from natural language publications to bridge the gap between a human-understandable and a machine-processable knowledge representation. The pipeline is built upon state-of-the-art NLP methods and is inspired by similar challenges from the biomedical domain. Although we demonstrate the potential of the approach (NER and QA show reliable computational performance scores), further validation of the approach to ensure practical usability is recommended. This includes especially the definition of the specific objective of the extracted information, e.g., for trend studies or identifying research gaps and contradictions within the domain of tribology. Furthermore, the annotation model is currently limited to model tests and is not validated to suit the practical information needs of tribologists. Therefore, user studies for analyzing the capability of information extraction compared with human experts provide possibilities to improve the performance of the approach.

Author Contributions: Conceptualization, P.K. and M.M.; methodology, P.K. and R.D.; software, R.D.; validation, R.D. and P.K.; writing—original draft preparation, P.K. and M.M.; writing—review and editing, P.K., M.M., R.D., B.S. and S.W.; visualization, P.K. and M.M.; supervision, B.S. and S.W.; funding acquisition, S.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The code for the Semantic Annotation Pipeline is available under <https://github.com/snow0815/TriboAnnotation.git>, accessed on 14 December 2021.

Acknowledgments: P. Kügler, B. Schleich and S. Wartzack gratefully acknowledge the financial support of project WA 2913/22-2 within the Priority Program 1921 by the German Research Foundation (DFG). M. Marian acknowledges the support from Pontificia Universidad Católica de Chile.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Holmberg, K.; Erdemir, A. Influence of tribology on global energy consumption, costs and emissions. *Friction* **2017**, *3*, 263–284. [[CrossRef](#)]
2. Zhang, Z.; Yin, N.; Chen, S.; Liu, C. Tribo-informatics: Concept, architecture, and case study. *Friction* **2021**, *3*, 642–655. [[CrossRef](#)]
3. Rosenkranz, A.; Marian, M.; Profito, F.J.; Aragon, N.; Shah, R. The Use of Artificial Intelligence in Tribology—A Perspective. *Lubricants* **2021**, *1*, 2. [[CrossRef](#)]
4. Hasan, M.S.; Kordijazi, A.; Rohatgi, P.K.; Nosonovsky, M. Triboinformatics Approach for Friction and Wear Prediction of Al-Graphite Composites Using Machine Learning Methods. *J. Tribol.* **2022**, *144*, 011701. [[CrossRef](#)]
5. Subrahmanyam, M.; Sujatha, C. Using neural networks for the diagnosis of localized defects in ball bearings. *Tribol. Int.* **1997**, *10*, 739–752. [[CrossRef](#)]
6. Prost, J.; Cihak-Bayr, U.; Neacșu, I.A.; Grundtner, R.; Pirker, F.; Vorlaufer, G. Semi-Supervised Classification of the State of Operation in Self-Lubricating Journal Bearings Using a Random Forest Classifier. *Lubricants* **2021**, *5*, 50. [[CrossRef](#)]

7. Sathiya, P.; Aravindan, S.; Haq, A.N.; Paneerselvam, K. Optimization of friction welding parameters using evolutionary computational techniques. *J. Mater. Process. Technol.* **2009**, *5*, 2576–2584. [CrossRef]
8. Cetinel, H. The artificial neural network based prediction of friction properties of Al₂O₃-TiO₂ coatings. *Ind. Lubr. Tribol.* **2012**, *5*, 288–293. [CrossRef]
9. Boidi, G.; Da Silva, M.R.; Profito, F.J.; Machado, I.F. Using Machine Learning Radial Basis Function (RBF) Method for Predicting Lubricated Friction on Textured and Porous Surfaces. *Surf. Topogr. Metrol. Prop.* **2020**, *4*, 44002. [CrossRef]
10. Bhaumik, S.; Pathak, S.D.; Dey, S.; Datta, S. Artificial intelligence based design of multiple friction modifiers dispersed castor oil and evaluating its tribological properties. *Tribol. Int.* **2019**, *140*, 105813. [CrossRef]
11. Sattari Baboukani, B.; Ye, Z.; Reyes, K.G.; Nalam, P.C. Prediction of Nanoscale Friction for Two-Dimensional Materials Using a Machine Learning Approach. *Tribol. Lett.* **2020**, *68*, 1–14. [CrossRef]
12. Marian, M.; Tremmel, S. Current Trends and Applications of Machine Learning in Tribology—A Review. *Lubricants* **2021**, *9*, 86. [CrossRef]
13. Stipčević, M.; Bowers, J.E. Spatio-temporal optical random number generator. *Opt. Express* **2015**, *9*, 11619–11631. [CrossRef]
14. Vinoth, A.; Datta, S. Design of the ultrahigh molecular weight polyethylene composites with multiple nanoparticles: An artificial intelligence approach. *J. Compos. Mater.* **2020**, *2*, 179–192. [CrossRef]
15. Bhaumik, S.; Mathew, B.R.; Datta, S. Computational intelligence-based design of lubricant with vegetable oil blend and various nano friction modifiers. *Fuel* **2019**, *241*, 733–743. [CrossRef]
16. Feigenbaum, E.A. Knowledge Engineering: The Applied Side of Artificial Intelligence. *Ann. N. Y. Acad. Sci.* **1984**, *426*, 91–107. [CrossRef]
17. Cullen, J.; Bryman, A. The knowledge acquisition bottleneck: Time for reassessment? *Expert Syst.* **1988**, *3*, 216–225. [CrossRef]
18. Tallian, T.E. Tribological Design Decisions Using Computerized Databases. *J. Tribol.* **1987**, *3*, 381–386. [CrossRef]
19. Tallian, T.E. A Computerized Expert System for Tribological Failure Diagnosis. *J. Tribol.* **1989**, *2*, 238–244. [CrossRef]
20. Studer, R.; Benjamins, V.R.; Fensel, D. Knowledge engineering: Principles and methods. *Data Knowl. Eng.* **1998**, *1–2*, 161–197. [CrossRef]
21. Morik, K. Underlying assumptions of knowledge acquisition and machine learning. *Knowl. Acquis.* **1991**, *2*, 137–156. [CrossRef]
22. Hoekstra, R. The knowledge reengineering bottleneck. *Semant. Web* **2010**, *2*, 111–115. [CrossRef]
23. Chandrasegaran, S.K.; Ramani, K.; Sriram, R.D.; Horváth, I.; Bernard, A.; Harik, R.F.; Gao, W. The evolution, challenges, and future of knowledge representation in product design systems. *Comput.-Aided Des.* **2013**, *2*, 204–228. [CrossRef]
24. Verhagen, W.J.; Bermell-Garcia, P.; van Dijk, R.E.; Curran, R. A critical review of Knowledge-Based Engineering: An identification of research challenges. *Adv. Eng. Inform.* **2012**, *1*, 5–15. [CrossRef]
25. Gruber, T.R. A translation approach to portable ontology specifications. *Knowl. Acquis.* **1993**, *2*, 199–220. [CrossRef]
26. Bodenreider, O. Biomedical Ontologies in Action: Role in Knowledge Management, Data Integration and Decision Support. *Yearb. Med. Inform.* **2008**, *1*, 67–79. [CrossRef]
27. The Gene Ontology Consortium. Creating the gene ontology resource: Design and implementation. *Genome Res.* **2001**, *8*, 1425–1433. [CrossRef]
28. Esnaola-Gonzalez, I.; Fernandez, I. Materials' Tribological Characterisation: An OntoCommons Use Case. In Proceedings of the ESWC Workshop DORIC-MM, Online, 7 June 2021.
29. Kügler, P.; Marian, M.; Schleich, B.; Tremmel, S.; Wartzack, S. tribAIIn—Towards an Explicit Specification of Shared Tribological Understanding. *Appl. Sci.* **2020**, *13*, 4421. [CrossRef]
30. Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; Ives, Z. *DBpedia: A Nucleus for a Web of Open Data*; ISWC 2007, ASWC 2007; Springer: Berlin/Heidelberg, Germany, 2007; pp. 722–735. [CrossRef]
31. Giorgi, J.M.; Bader, G.D. Transfer learning for biomedical named entity recognition with neural networks. *Bioinformatics* **2018**, *23*, 4087–4094. [CrossRef]
32. Lee, J.; Yoon, W.; Kim, S.; Kim, D.; Kim, S.; So, C.H.; Kang, J. BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* **2020**, *4*, 1234–1240. [CrossRef]
33. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. *BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding*; NAACL-HLT; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 4171–4186. [CrossRef]
34. Czichos, H.; Habig, K.-H. *Tribologie-Handbuch*, 4th ed.; Springer: Wiesbaden, Germany, 2015.
35. Marian, M. *Numerische Auslegung von Oberflächenmikrostrukturen für Geschmierte Tribologische Kontakte*; FAU University Press: Erlangen, Germany, 2021. [CrossRef]
36. Vakis, A.I.; Yastrebov, V.A.; Scheibert, J.; Nicola, L.; Dini, D.; Minfray, C.; Almqvist, A.; Paggi, M.; Lee, S.; Limbert, G.; et al. Modeling and simulation in tribology across scales: An overview. *Tribol. Int.* **2018**, *125*, 169–199. [CrossRef]
37. Soldatova, L.N.; King, R.D. An ontology of scientific experiments. *J. R. Soc. Interface* **2006**, *11*, 795–803. [CrossRef] [PubMed]
38. W3C Recommendation. OWL 2 Web Ontology Language Overview. Available online: <http://www.w3.org/TR/owl2-overview/> (accessed on 13 October 2021).
39. Baader, F. *The description logic handbook: Theory, Implementation, and Applications*; Cambridge University Press: Cambridge, UK, 2003.
40. W3C Recommendation. RDF 1.1 Turtle: Terse RDF Triple Language. Available online: <https://www.w3.org/TR/turtle/> (accessed on 14 December 2021).

41. Gruber, T.R. Toward principles for the design of ontologies used for knowledge sharing? *Int. J. Hum.-Comput. Stud.* **1995**, *5–6*, 907–928. [CrossRef]
42. Navigli, R.; Velardi, P. From Glossaries to Ontologies: Extracting Semantic Structure from Textual Definitions. In *Ontology Learning and Population. Bridging the Gap between Text and Knowledge*; Buitelaar, P., Cimiano, P., Eds.; IOS Press: Amsterdam, The Netherlands, 2008.
43. Marian, M.; Feile, K.; Rothhammer, B.; Bartz, M.; Wartzack, S.; Seynstahl, A.; Tremmel, S.; Krauß, S.; Merle, B.; Böhm, T.; et al. Ti₃C₂T_x solid lubricant coatings in rolling bearings with remarkable performance beyond state-of-the-art materials. *Appl. Mater. Today* **2021**, *25*, 101202. [CrossRef]
44. Wyatt, B.C.; Rosenkranz, A.; Anasori, B. 2D MXenes: Tunable Mechanical and Tribological Properties. *Adv. Mater.* **2021**, *17*, e2007973. [CrossRef] [PubMed]
45. Marian, M.; Berman, D.; Rota, A.; Jackson, R.L.; Rosenkranz, A. Layered 2D Nanomaterials to Tailor Friction and Wear in Machine Elements—A Review. *Adv. Mater. Interfaces* **2021**, *9*, 2101622. [CrossRef]
46. Marian, M.; Tremmel, S.; Wartzack, S.; Song, G.; Wang, B.; Yu, J.; Rosenkranz, A. Mxene nanosheets as an emerging solid lubricant for machine elements—Towards increased energy efficiency and service life. *Appl. Surf. Sci.* **2020**, *523*, 146503. [CrossRef]
47. Domingue, J. *Handbook of Semantic Web Technologies*; Springer: Berlin, Germany, 2011.
48. Goldberg, Y. Neural Network Methods for Natural Language Processing. *Synth. Lect. Hum. Lang. Technol.* **2017**, *1*, 1–309. [CrossRef]
49. Jurafsky, D.; Martin, J.H. Speech and Language Processing (Draft-Version from Dec 2020). Available online: <https://web.stanford.edu/~jjurafsky/slp3/> (accessed on 11 August 2021).
50. Hu, J. An Overview for Text Representations in NLP. Blogpost: Towards Data Science. Available online: <https://towardsdatascience.com/an-overview-for-text-representations-in-nlp-311253730af1?gi=4f92ddbfac7d> (accessed on 18 October 2021).
51. Schuster, M.; Nakajima, K. Japanese and Korean voice search. In Proceedings of the ICASSP 2012—2012 IEEE International Conference on Acoustics, Speech and Signal Processing, Kyoto, Japan, 25–30 March 2012; IEEE: New York, NY, USA; pp. 5149–5152. [CrossRef]
52. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. *arXiv* **2013**, arXiv:1301.3781.
53. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed Representations of Words and Phrases and Their Compositionality. *arXiv* **2013**, arXiv:1310.4546.
54. Koroteev, M.V. BERT: A Review of Applications in Natural Language Processing and Understanding. *arXiv* **2021**, arXiv:2103.11943.
55. Peters, M.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. *Deep Contextualized Word Representations*; NAC, Association for Computational Linguistics: Stroudsburg, PA, USA, 2018; pp. 2227–2237. [CrossRef]
56. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving Language Understanding by Generative Pre-Training. Available online: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf (accessed on 14 December 2021).
57. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762.
58. Ghelani, S. From Word Embeddings to Pretrained Language Models—A New Age in NLP-Part 2. Blogpost: Towards Data Science. Available online: <https://towardsdatascience.com/from-word-embeddings-to-pretrained-language-models-a-new-age-in-nlp-part-2-e9af9a0bdcd9> (accessed on 7 October 2021).
59. Alammari, J. The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning). Blogpost. Available online: <http://jalammari.github.io/illustrated-bert/> (accessed on 14 December 2021).
60. Alammari, J. The Illustrated Transformer. Blogpost. Available online: <http://jalammari.github.io/illustrated-transformer/> (accessed on 20 October 2021).
61. Beltagy, I.; Lo, K.; Cohan, A. SciBERT: A Pretrained Language Model for Scientific Text. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 3613–3618. [CrossRef]
62. Joshi, M.; Chen, D.; Liu, Y.; Weld, D.S.; Zettlemoyer, L.; Levy, O. SpanBERT: Improving Pre-training by Representing and Predicting Spans. *Trans. Assoc. Comput. Linguist.* **2020**, *8*, 64–77. [CrossRef]
63. Russell, S.J.; Norvig, P. *Artificial Intelligence—A Modern Approach*, 4. Auflage; Pearson: Boston, MA, USA, 2021.
64. Cho, H.; Lee, H. Biomedical named entity recognition using deep neural networks with contextual information. *BMC Bioinform.* **2019**, *1*, 735. [CrossRef]
65. McKie, J.X. PyMuPDF 1.19.0. GitHub Repository. Available online: <https://github.com/pymupdf/PyMuPDF> (accessed on 21 October 2021).
66. Akbik, A.; Bergmann, T.; Blythe, D.; Rasul, K.; Schweter, S.; Vollgraf, R. FLAIR: An easy-to-use framework for state-of-the-art NLP. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations), Minneapolis, MN, USA, 2–7 June 2019; pp. 54–59. [CrossRef]
67. Marian, M.; Song, G.C.; Wang, B.; Fuenzalida, V.M.; Krauß, S.; Merle, B.; Tremmel, S.; Wartzack, S.; Yu, J.; Rosenkranz, A. Effective usage of 2D MXene nanosheets as solid lubricant—Influence of contact pressure and relative humidity. *Appl. Surf. Sci.* **2020**, *531*, 147311. [CrossRef]

68. Mekingwe, G.N.; Akinribide, O.J.; Langa, T.; Obadele, B.A.; Olubambi, P.A.; Lethabane, L.M. Effect of graphite addition on the tribological properties of pure titanium carbonitride prepared by spark plasma sintering. *IOP Conf. Ser. Mater. Sci. Eng.* **2019**, *499*, 12011. [[CrossRef](#)]
69. Li, X.; Xu, J. Coordinating influence of multilayer graphene and spherical SnAgCu for improving tribological properties of a 20CrMnTi material. *RSC Adv.* **2018**, *25*, 14129–14137. [[CrossRef](#)]
70. Wang, Z.; Hu, S.; Feng, C.; Chen, E. The high temperature and varying temperature tribological performance of TiC coatings. In *IOP Conference Series: Materials Science and Engineering*; IOP Publishing: Bristol, UK, 2018; p. 22032. [[CrossRef](#)]
71. Byeong-Choon, G.; In-Sik, C. Microstructural Analysis and Wear Performance of Carbon-Fiber-Reinforced SiC Composite for Brake Pads. *Materials* **2017**, *7*, 701. [[CrossRef](#)]
72. Hinton, G.E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R.R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv* **2012**, arXiv:1207.0580.
73. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *1*, 1929–1958.
74. Gruber, T.R. How AI Can Enhance Our Memory, Work and Social Lives. TED Talk. 2017. Available online: https://www.ted.com/talks/tom_gruber_how_ai_can_enhance_our_memory_work_and_social_lives (accessed on 14 December 2021).