TECHNISCHE UNIVERSITÄT WIEN
Vienna University of Technology

# Master Thesis

# Transformer Event Extraction Explainer: A Tool for improving Explainability of Transformer Models for Industrial Maintenance Applications

carried out to obtain the degree of Master of Science (MSc or Dipl.-Ing. or DI), submitted at TU Wien, Faculty of Mechanical and Industrial Engineering, by

**Lukas Künig BSc**

[redacted]

under the supervision of

**Univ.Prof. Dr.-Ing. Fazel Ansari**

Institute of Management Science, E330, Research Group of Production and Maintenance Management

**Univ. Lektor. Dipl.-Ing. Linus Kohl**

Institute of Management Science, E330, Research Group of Production and Maintenance Management

Vienna, June 2024

[redacted]

Lukas Künig

TECHNISCHE UNIVERSITÄT WIEN
Vienna University of Technology



In lieu of oath, I declare that I wrote this thesis and performed the associated research using only the literature cited in this volume. If text passages from sources are used literally, they are marked as such.

I confirm that this work is original and has not been submitted elsewhere for any examination, nor is it currently under consideration for a thesis elsewhere.

I acknowledge that the submitted work will be checked electronically using suitable and state-of-the-art means (plagiarism detection software). On the one hand, this ensures that the submitted work was prepared according to the high-quality standards within the applicable rules to ensure good scientific practice “Code of Conduct” at the TU Wien. On the other hand, a comparison with other student theses avoids violations of my copyright.

Vienna, June 2024

[redacted]

Lukas Künig

# Acknowledgements

I want to take this opportunity to express my deepest thanks to all those who have supported and motivated me in preparing this thesis.

First of all, I thank my advisor, Univ.Prof. Dr.-Ing. Fazel Ansari, and my supervisor, Univ. Lektor. Dipl.-Ing. Linus Kohl. Above all, I would like to thank you for the excellent professional support and constructive feedback throughout the process of writing the thesis. I appreciate your advice and expertise.

I would also like to thank my fellow students and all those close to me, especially my girlfriend Verena Webhofer, who supported and accompanied me outside of my studies.

Finally, I would like to thank my parents, Annja and Norbert Künig, whose support has made my studies possible and who have always been there for me.

# Abstract

AI models, especially transformers, are becoming increasingly important in industrial maintenance. One of the biggest challenges of these models is the lack of explainability in the model decisions. Most of the time, it is difficult for humans to determine how the model arrived at a particular decision, which leads to mistrust and scepticism towards the models. Therefore, explainability is crucial to increase trust in these models so that they can be applied in real-world applications. The Transformer Event Extraction Explainer (TEEE) application improves the explainability of transformer models by clearly explaining the model's decisions.

This diploma thesis focuses on developing and evaluating an event extraction application that improves the explainability of model decisions in industrial maintenance. The main objective is to create a model to extract relevant maintenance events in unstructured texts and explain their importance for model decision-making.

The application is based on transformer models, particularly Google's BERT[1] model, which has been fine-tuned for extracting events from texts. The TEEE application makes it possible to quantify the influence of individual words on the model predictions and make them explainable through visual representations. This is done with the help of SHAP[2] values, which indicate positive and negative influences on the event extraction probability at the word level. An overall evaluation is made using a plot that shows the average event probability and the event quantity for the entire analyzed text.

This work's results show that TEEE can reliably extract relevant events from industrial maintenance texts and transparently present their impact on the model decision. The developed method enables a better understanding of the decision-making process in event extraction with a transformer and thus increases confidence in applying such models.

Overall, this thesis contributes to the connection of transformer models and actual applications in industrial maintenance and forms the basis for future optimizations and adaptations, especially by using training-specific data from the respective domain.

[1] Bidirectional Encoder Representations from Transformers
[2] SHapley Additive exPlanations

# Kurzfassung

KI-Modelle, insbesondere Transformer, sind in der industriellen Instandhaltung immer wichtiger geworden. Eine der größten Herausforderungen bei der Anwendung dieser Modelle ist die mangelnde Erklärbarkeit der Modellentscheidungen. Meistens ist es für Menschen schwer zu erkennen, wie das Modell zu einer bestimmten Entscheidung gekommen ist, was zu Misstrauen und Skepsis führt. Die Erklärbarkeit ist von großer Bedeutung, um das Vertrauen in Transformer Modelle zu stärken und ihren Einsatz in praktischen Anwendungen zu ermöglichen. Der Transformer Event Extraction Explainer (TEEE) verbessert die Erklärbarkeit von Vorhersagen von Transformer Modellen, indem er die Entscheidungen des Modells nachvollziehbar macht.

Diese Diplomarbeit beschäftigt sich mit der Entwicklung und Evaluierung einer Anwendung zur Ereignisextraktion, welche die Erklärbarkeit der Modellentscheidungen im Bereich der industriellen Instandhaltung verbessert. Das Hauptziel besteht darin, eine Applikation zu entwickeln, welche relevante Ereignisse aus der industriellen Instandhaltung in unstrukturierten Texten extrahiert und die Bedeutung dieser für die Entscheidungsfindung des Modells erklären kann.

Die Applikation basiert auf der Verwendung von Transformer-Modellen, insbesondere dem BERT-Modell von Google, das für die Extraktion von Ereignissen aus Texten feinabgestimmt wurde. Der entwickelte TEEE ermöglicht es, den Einfluss einzelner Wörter auf die Modellvorhersagen zu quantifizieren und durch visuelle Darstellungen verständlich zu machen. Dies geschieht mit Hilfe von SHAP-Werten (SHapley Additive exPlanations), die positive und negative Einflüsse auf die Ereigniserkennungswahrscheinlichkeit auf Wortebene anzeigen. Eine Gesamtbewertung erfolgt mit Hilfe eines übersichtlichen Plots, der die mittlere Ereigniswahrscheinlichkeit und die Ereignisquantität für den gesamten analysierten

Die Ergebnisse dieser Arbeit zeigen, dass TEEE in der Lage ist, relevante Ereignisse aus industriellen Instandhaltungstexten zuverlässig zu extrahieren und deren Einfluss auf die Modellentscheidung transparent darzustellen. Die entwickelte Methode ermöglicht ein besseres Verständnis der Entscheidungsfindung bei der

Ereignisextraktion mit einem Transformer und erhöht somit das Vertrauen in die Anwendung solcher Modelle.

Insgesamt trägt diese Arbeit zur Verbindung von Transformer-Modellen und realen Anwendungen in der industriellen Instandhaltung bei und bildet die Basis für zukünftige Optimierungen und Anpassungen, insbesondere durch die Verwendung trainingsspezifischer Daten aus der jeweiligen Domäne.

# List of abbreviations

| | |
|---|---|
| AI | Artificial Intelligence |
| AAI | Accountable Artificial Intelligence |
| BERT | Bidirectional Encoder Representations from Transformers |
| CNN | Convolutional Neural Network |
| CPS | Cyber-Physical Systems |
| CPPS | Cyber-Physical Production Systems |
| CPU | Central Processing Unit |
| CSV | Comma Separated Value |
| DDR4 | Double Data Rate 4 |
| DL | Deep Learning |
| DNNs | Deep Neural Networks |
| DSR | Design Science Research |
| ERP | Enterprise Resource Planning |
| FAI | Fair Artificial Intelligence |
| GDDR5 | Graphics Double Data Rate 5 |
| GPU | Graphics Processing Unit |
| GAI | Green Artificial Intelligence |
| Grad-CAM | Gradient-weighted Class Activation Mapping |
| GRU | Gated Recurrent Unit |
| GPT | Generative Pre-trained Transformer |
| GUI | Graphical User Interface |
| HTML | Hypertext Markup Language |
| HCAI | Human-Centric Artificial Intelligence |

| | |
|---|---|
| IIoT | Industrial Internet of Things |
| LRP | Layer-Wise Relevance Propagation |
| LSTM | Long Short-Term Memory |
| MES | Manufacturing Execution System |
| MLM | Masked-Language Modeling |
| NBC | Naïve Bayes Classifier |
| NER | Named-entity Recognition |
| NLP | Natural Language Processing |
| NLU | Natural Language Understanding |
| PdM | Predictive Maintenance |
| POS | Part-of-speech Tagging |
| PriMa | Prescriptive Maintenance Model for Cyber-Physical Production Systems |
| RAI | Responsible Artificial Intelligence |
| RNN | Recurrent Neural Network |
| SOC | Sum of Squares due to Change |
| TLP | Technical Language Processing |
| TEEE | Transformer Event Extraction Explainer |
| TXT | Text File Format |
| ULMFiT | Universal Language Model Fine-Tuning |
| XAI | Explainable Artificial Intelligence |

# Table of contents


**1 Introduction ..... 1**

1.1 Problem Definition ..... 2

1.2 Research Question and Aim of the Thesis ..... 4

1.3 Methodology of Research and Architecture of the artefact ..... 6

**2 Theoretical Background ..... 9**

2.1 Transformation of maintenance strategies in industry 4.0 ..... 9

2.2 Technical Language Processing (TLP) ..... 14

2.2.1 NLP Methods for Industrial Maintenance ..... 14

2.3 Transformer Models ..... 18

2.3.1 Deep Neural Networks (DNNs) ..... 18

2.3.2 Attention Mechanism ..... 19

2.3.3 The Transformer ..... 20

2.3.4 Bidirectional and Unidirectional Transformer Models ..... 21

2.3.5 Encoder and Decoder Transformer Models ..... 21

2.3.6 Transfer Learning in NLP ..... 23

**3 State-of-the-Art Explainability of Transformer Models ..... 25**

3.1 Methodology of the Systematic Literature Review ..... 25

3.2 Summary and Results ..... 32

**4 Transformer Models for Zero-Shot Event Extraction in Industrial Use-Cases ..... 35**

4.1 Hugging Face Transformer Models ..... 36

4.2 Evaluation of Transformer Models for Event Extraction in Industrial Maintenance ..... 37

4.2.1 Evaluation Methodology ..... 37

4.2.2 Evaluation Results ..... 40

4.3 Explainability of Transformer Models ..... 45

4.3.1 XAI Terminology ..... 47

4.3.2 Dimensions of Transformer Model Explainability ..... 49

4.3.3 Transformer Explainability Taxonomy ..... 52

4.4 Implementation of an Explainability Framework for Zero-Shot Event Extraction ..... 56

4.4.1 The “Transformer Event Extraction Explainer” (TEEE) ..... 58

4.4.2 Key components of TEEE ..... 61

**5 Evaluation of the Transformer Event Extraction Explainer (TEEE) ..... 65**

5.1 Evaluation Methodology and Dataset ..... 65
5.2 Quantitative Evaluation of TEEE Runtime ..... 66
5.3 Qualitative Evaluation of TEEE Explanations ..... 68

**6 Conclusion and Outlook ..... 73**

6.1 Limitations and Future Work ..... 75

**References ..... 77**

**List of Figures ..... 87**

**List of Tables ..... 88**

**Appendix ..... 89**

# 1 Introduction

Introducing Industry 4.0 has led to a comprehensive transformation in the manufacturing industry. The manufacturing environment is constantly becoming more connected, complex, and transformative (Peres et al., 2020). As a result, more and more data is being generated, as shown in Figure 1, and informed decision-making based on this data is becoming increasingly important for manufacturing companies (C. Li et al., 2022). A subsidiary discipline of artificial intelligence (AI), natural language processing (NLP) is a key technology for informed decision-making. Many research institutions are focusing on AI technologies. Although AI technologies are being actively developed in research institutions, their practical application in an industrial environment needs to catch up with the state-of-the-art in research (Peres et al., 2020).

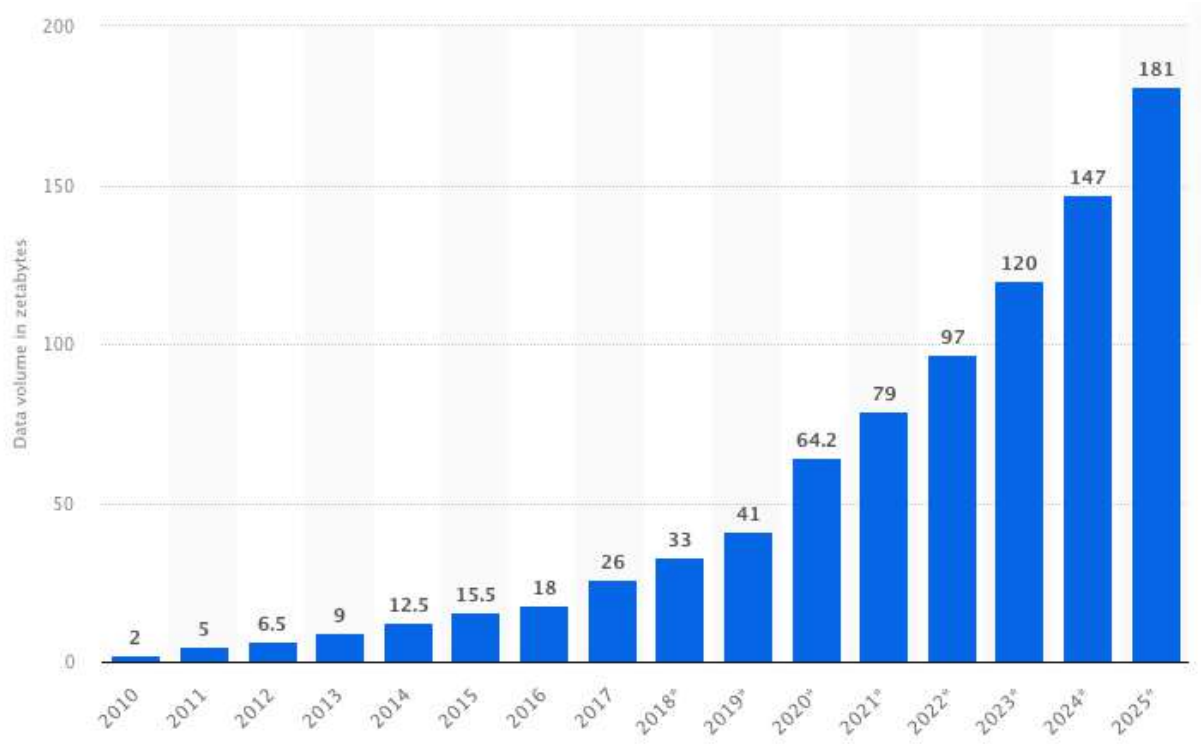



**Figure 1** Volume of worldwide Data (Statista, 2024)

The predictive maintenance (PdM) technologies market has been growing in recent years, with a report by IOT Analytics (2021) predicting its growth from 6.9 Billion USD in 2021 to 28.2 Billion USD by 2026 (IOT Analytics, 2021). PdM aims to detect impending machine failures in advance and thus avoid machine downtimes in production (Lee et al., 2006). By using historical data and specialist knowledge in combination with statistical or machine learning models, trends or correlations in the

data can be identified, thus allowing for the minimization of machine downtimes. (Peres et al., 2020).

The use of transformer models has seen a significant increase in the field of NLP. Transformer models currently represent the state-of-the-art in NLP. However, their deep neural network architecture complicates making these models explainable or interpretable to human understanding (Tamekuri et al., 2022). To use these models in critical areas such as industrial maintenance, they have to gain the trust of all stakeholders. This trust depends on the explainability of the models, highlighting the importance of achieving transparency and interpretability in the operation of transformer models (Besinger et al., 2023). Therefore, the main challenge is bridging the gap between their performance and explainability (Di Flumeri, 2022). In this chapter, the problem is defined, followed by the research question and the aim of the thesis. The methodology and approach used are then discussed and finally, the architecture of the artefact, namely TEEE[3], is presented.

## 1.1 Problem Definition

Industry 4.0 is one of the major developments in the field of maintenance, focusing mainly on the analysis and application of data in the industrial sector. In general, Industry 4.0 focuses on generating knowledge from data (May et al., 2022). In Industry 4.0, structured sensor data is mainly used for data analysis (Yan et al., 2017). Professionals' experience and knowledge are often documented but not formalized and processed, which does not ensure the efficient use of this data (May et al., 2022). For this reason, natural language data is not considered in many cases (May et al., 2022). Event extraction is an essential application of NLP in the industrial maintenance sector. Event extraction in natural language text can be used to identify maintenance-related events, trends, and patterns. Despite the growing understanding in the area of NLP, further research is needed on the explainability of transformer model decisions, especially in industrial maintenance (P1) (Ding et al., 2022). In the following thesis, the term transformer models is interchangeably used, referring to the field of NLP.

NLP methods, especially transformer models, have demonstrated efficacy in various scientific domains, but their utility in industrial maintenance use cases remains

[3] Transformer Event Extraction Explainer

uncertain (P2) (Ansari et al., 2021). The analysis of maintenance-related texts presents a significant challenge due to their inherent characteristics. These texts may contain incomplete information and various errors that affect data quality (Mahlamäki et al., 2016). The use of individual jargon by machine operators in maintenance reports especially negatively impacts the quality of the data (Sexton & Fuge, 2019). Consequently, this thesis investigates the potential usefulness of state-of-the-art transformer models, such as BERT, in industrial maintenance settings. Specifically, the current state-of-the-art transformer models will be examined to assess its suitability for industrial maintenance applications.

In an industrial environment, models such as BERT must make accurate predictions. Explainability is key to achieving this trust (Di Flumeri, 2022). Explainability of transformer models is the ability of a human to justify the results of the transformer model (Namatēvs et al., 2022). However, the effectiveness of event extraction cannot be judged from this, as it depends on several other factors such as the F1[4] score. This thesis focuses on the explainability of transformer models for zero-shot event extraction in industrial maintenance use cases. Zero-shot event extraction is a downstream task where a model extracts specific information from a document without having seen any annotated examples of that information during training (H. Zhang et al., 2022).

For this reason, the effectiveness of event extraction is evaluated in a simplified way based on the F1 score, the accuracy, and the model runtime. The understanding of the logic behind the model predictions of transformer models in industrial maintenance scenarios through practical experimentation and analysis is incomplete (P3) (Klaise et al., 2020). To address P2 and P3, an industrial dataset sourced explicitly from the maintenance field will be utilized throughout the practical component of this thesis. Table 1 describes the problems P1, P2 and P3 of this thesis.

[4] The F1 score is the harmonic mean of a model's precision and recall.

**Table 1** Problems P1, P2 and P3

| Problem | Description |
|---|---|
| P1: | Lack of understanding of how transformer models can be explained, especially within the context of industrial maintenance. |
| P2: | Lack of practical application of transformer models in industrial maintenance use cases, especially for event extraction. |
| P3: | Lack of explainability of transformer models predictions for event extraction in industrial maintenance use cases. |

## 1.2 Research Question and Aim of the Thesis

To generate knowledge from natural language texts, NLP methods offer a large variety of different approaches. The application of transformer models, such as BERT, is one of the most sophisticated approaches. The solution to the problem (P1) will be a literature review on the explainability of transformer models, especially in industrial maintenance (O1).

Transformer models are already being used successfully in various technology fields, such as conversational agents. Applying transformer models to a domain-specific dataset, focusing on event extraction will clarify the current utilization and success rate of transformer models in industrial maintenance (P2). The F1 score and the accuracy will be used to evaluate the fine-tuned transformer models. This will identify transformer models' applicability in industrial maintenance use cases (O2).

The explainability of the transformer model predictions in the area of industrial maintenance will be evaluated using quantitative and qualitative methods. To achieve this objective, (P3) will be addressed, and the explainability of transformer model predictions, especially for event extraction, will be evaluated in industrial maintenance use cases (O3).

This thesis's primary objective will be to clarify the explainability of transformer models and apply these to specific industrial maintenance use cases. A morphology of transformer models will be created by conducting literature research on the explainability of transformer models in industrial maintenance use cases and utilizing

state-of-the-art methodology in transformer models. Based on the considerations above, the primary research inquiry emerges as follows:

***"How to increase the explainability of transformer model predictions for zero-shot event extraction in industrial maintenance?"***

Table 2 presents the sub research questions RQ1, RQ2 and RQ3.

**Table 2** Sub Research Questions for P1, P2 and P3

| Abbreviation | Research Question |
|---|---|
| RQ1: | What explainability methodology can be used to explain transformer model predictions for zero-shot event extraction? |
| RQ2: | What is the utilization and success rate of transformer models when applied to an industrial maintenance dataset for zero-sot event extraction tasks? |
| RQ3: | What measurement methods can be used to evaluate the explainability of transformer models when applied to an industrial maintenance dataset for zero-shot event extraction tasks? |

Based on the sub-research questions RQ1, RQ2 and RQ3 defined in Table 2, the objectives of this thesis are defined in Table 3.

**Table 3** Objectives of this Thesis

| Abbreviation | Objective |
|---|---|
| O1: | Identify state-of-the-art for understanding and enhancing the explainability of transformer model predictions |
| O2: | Evaluation of transformer models for zero-shot event extraction in industrial maintenance |
| O3: | Evaluating explainability of transformer models for zero-shot event extraction in industrial maintenance |

## 1.3 Methodology of Research and Architecture of the artefact

The methodology of this thesis is based on the three-cycle view of the Design Science Research (DSR) framework by (Hevner, 2007). This methodology aims to create a new and innovative artefact that should be evaluated through appropriate approaches. Artefacts are constructs, models, methods, and instantiations (Hevner et al., 2004). In the context of this work, the artefact is a tool, namely TEEE, which enhances the explainability of transformer models in industrial maintenance. To ensure clear definitions, boundaries, guidelines, and outcomes of this work, it will be carried out according to the framework of the DSR, as seen in Figure 2 by (Hevner,2007).

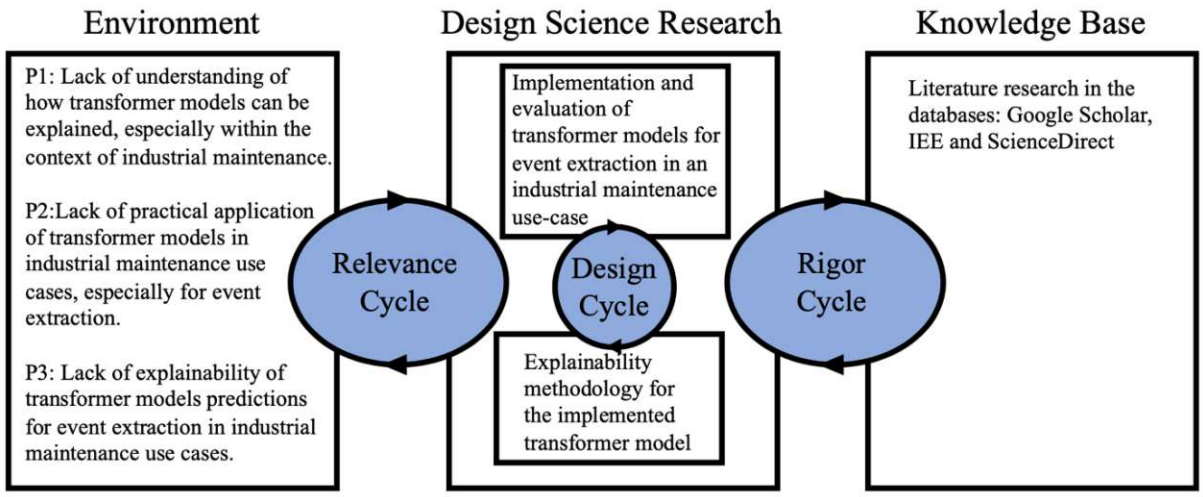



**Figure 2** Design Science Research Cycle based on (Hevner, 2007)

DSR aims to improve the environment through new and innovative artefacts and document the process of artefact discovery (Simon, 2008). In this work, an explainability software tool for transformer model-based zero-shot event extraction in industrial maintenance will emerge as an artefact. Primarily, the relevance cycle initiates the definition of the problems, the opportunities, and the requirements in the specific environment of the application. The relevance cycle also defines the success criteria for the research project. The research results should then be reintegrated into the background for analysis. The concept of three cycles is understood as an iterative solution approach applied in each cycle (Hevner, 2007).

The rigor cycle focuses on the knowledge base and state-of-the-art. (Hevner, 2007) distinguishes between two types of knowledge:

- The expertise that constitutes the current best practices within the research's application area.

- Artefacts have already been discovered and evaluated in the research area.

The rigor cycle helps to ensure that innovative research is carried out and not just routine design based on known procedures (Hevner et al., 2004). All achieved findings, theories and artefacts are added to the knowledge base after successful testing in the application environment (Hevner, 2007). In this thesis, the rigor cycle ensures that the development of the TEEE is based on the latest existing knowledge in the field of explainability of transformer models.

The design cycle is the primary component of the DSR cycle. The design cycle iterates between the artefact's design, evaluation, and the implementation of improvements. This means the design is iterated in the design cycle until it fulfils the requirements. The requirements that must be fulfilled are defined in the relevance cycle. The rigor cycle analyzes the fundamental knowledge base (Hevner, 2007). In this thesis, the development of the TEEE is performed in the design cycle. In the relevance cycle, different transformer models are evaluated for the application in the TEEE on the one hand and the explainability of the TEEE is evaluated on the other hand.

This thesis will be elaborated according to the framework of the three cycles by (Hevner, 2007). The requirements are raised in the recurring meetings with the supervisor. In the context of this thesis, the supervisor represents the environment. The artefact is generated based on these requirements and the expertise out of the knowledge database (Rigor Cycle). After the design cycle is completed, the artefact is applied to appropriate use cases defined by the supervisor. The accomplished thesis represents the contribution to the knowledge base within the framework of the rigor cycle.

It is worth noting that the application "Transformer Event Extraction Explainer", shortened to TEEE, emerges as the artefact of this master thesis. TEEE, which is based on the BERT model, is designed for human interpretation of transformer model-based event extraction predictions. TEEE is a tool that takes a maintenance text from a machine logbook and a list of possible events to extract from the text as input. Initially, the input passes through a specially developed Event Extraction Pipeline in TEEE, which calculates the event's probability predictions. Subsequently, a dynamic

threshold for event detection is computed, and the predictions are compared against this threshold. This calculation is based on a combination of the mean and standard deviation of the event scores. Next, Shapley values for all extracted events are calculated for each token. If the text contains multiple sentences, this process is repeated for each sentence. TEEE generates two output diagrams: The overall results plot visually represents the average event extraction probability and the count of events extracted from the given text. The explainability plot with Shapley values provides explanations for each detected event within every sentence token. This plot visualizes the contribution of each token to the model's prediction. Figure 3 below illustrates the overall architecture of TEEE using a simple example.

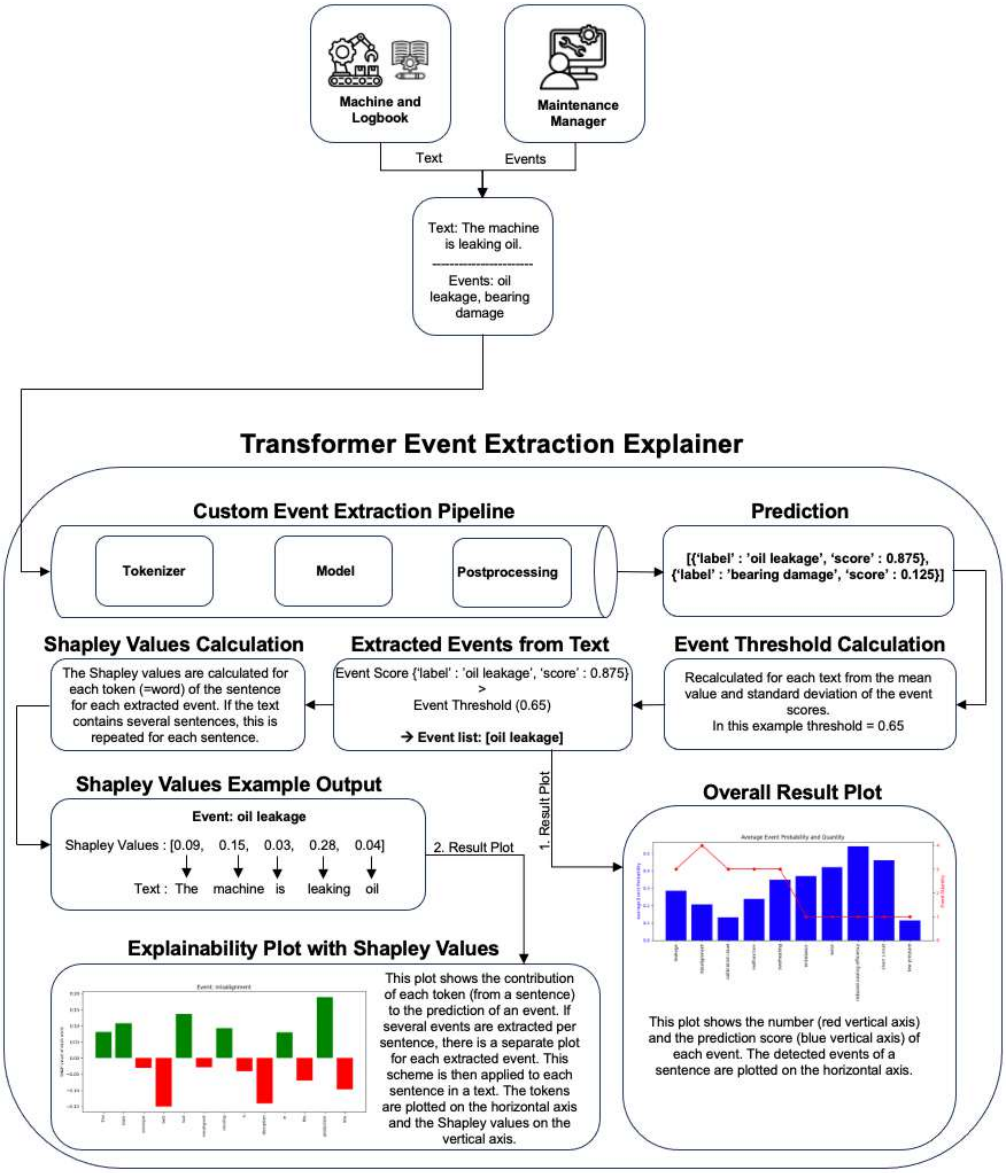



**Figure 3** Overall Architecture of TEEE

# 2 Theoretical Background

Data-driven approaches have gained significant importance in numerous sectors during the current era of digitization and Industry 4.0. The ability to gather and analyze large volumes of data has changed how businesses and industries operate. The industrial maintenance sector, in particular, has undergone one of the most significant transformations. This transformation includes the introduction of data-driven technologies, such as predictive maintenance or IIoT (Industrial Internet of Things) (Matyas, 2022). For this reason, this chapter explores the theoretical basis of data-driven industrial maintenance in Section 2.1, NLP in industrial maintenance in Section 2.2, and the transformer model in general in Section 2.3.

## 2.1 Transformation of maintenance strategies in industry 4.0

Given the increasing complexity of production processes, the classical maintenance methods - failure repair, time-controlled periodic maintenance, and condition-based maintenance - are insufficient to ensure the reliability demanded by the plant. Classical maintenance strategies can still be successfully employed in constantly operating mass production systems (Nemeth et al., 2015).

The interconnection of numerous systems and the possibility of processing and analyzing large amounts of data and obtaining helpful information is part of the fourth industrial revolution, also known as Industry 4.0 (Kagermann et al., 2013). Industry 4.0 aims to make all materials, products, production facilities, tools, transport technologies, conveyor and storage systems, and buildings “smart”. (Matyas, 2022).

The following section introduces three classic maintenance methods: run-to-failure, time-controlled periodic, and condition-based maintenance. It then discusses two innovative industrial maintenance approaches: predictive and prescriptive maintenance.

In the maintenance strategy known as run-to-failure, machines or systems are operated until they break down, and then the necessary repairs are made. This approach results in the longest possible maintenance interval but often leads to

unplanned downtime during production. Maintenance typically involves finding a skilled mechanic to diagnose and fix the problem, procuring spare parts, and testing the system to ensure it functions properly before restarting production. However, the run-to-failure strategy has limited usefulness in modern industrial maintenance. This approach is only recommended in cases where the affected machine is redundant or of secondary importance to the production process. In most cases, other maintenance strategies are preferable to minimize downtime and maintain optimal production efficiency. (Matyas, 2022).

Time-controlled periodic maintenance involves the preventive maintenance of predefined assemblies or individual parts after a set lifetime. This means that the machine is maintained after the predefined period, regardless of its actual condition. This maintenance strategy is beneficial when machine failure could potentially impact safety or the environment or when the estimated lifetime of the parts is known, and the other components are still in good condition. Planned maintenance is generally quicker and, therefore, cheaper than unplanned maintenance. It is essential to keep maintenance and equipment downtime costs as low as possible in industrial maintenance. To achieve this, the periodic maintenance interval must be adjusted to the utilization reserve of the parts under observation. If maintenance is performed too early, the utilization reserve of the parts is not optimally utilized (Matyas, 2022).

On the other hand, if maintenance is delayed, it may result in increased wear of other components or an unplanned machine fault, which is the worst-case scenario. Determining the optimal period for time-controlled periodic maintenance is complex and time-consuming. However, this problem can be solved using the condition-based maintenance strategy (Matyas, 2022).

Condition-based maintenance is an approach that involves adjusting maintenance processes based on the specific degree of wear of the object being maintained. By using suitable monitoring and diagnostic systems, the level of wear is measured, and when there is a significant deviation from the required machine performance, an appropriate information system initiates the maintenance process. This approach ensures a highly efficient and dynamic adjustment of maintenance intervals and utilization reserves. In most cases, there are warning signs of a potential fault before a genuine fault occurs in a machine. The period between potential and genuine faults can vary from milliseconds to several years. The primary goal of condition-based

maintenance is to predict failures immediately to avoid unexpected faults. Maintenance measures are applied after a potential fault has been identified to ensure the machine can deliver the required performance. Another essential aim of the condition-based maintenance strategy is to monitor the actual degree of machine wear. The degree of wear is measured using monitoring systems and system diagnostics, often called "technical diagnostics" in academic literature. In the industry, such systems used to monitor machine conditions are known as "condition monitoring" systems. The condition-based maintenance strategy has significant economic benefits, including decreased operating costs, plant expenses, maintenance expenses, downtime expenses, and technical advantages. By continuously monitoring the machine's condition, the plant's safety can always be guaranteed (Matyas, 2022).

Many companies have flexible manufacturing systems for a wide range of products. A predictive and holistic maintenance strategy is crucial to reliably using such flexible production systems while saving resources. It must incorporate data from many systems, e.g., condition monitoring systems, quality and machine data, and historical knowledge about failure events (Nemeth et al., 2015). Predictive maintenance utilizes various data sources to identify anomalous behaviour within a plant (diagnosis), accurately anticipate faults that may occur in the future (forecasting), and enable well-informed decisions beforehand (proactive decision-making) (Bousdekis et al., 2019). The anticipation of wear signs and their effects on the production process is crucial for the development of future maintenance planning. Therefore, predictive maintenance models access historical machine condition data. Probabilistic models, such as the Weibull distribution, can be applied to this data to gain insight into the future wear behaviour of the individual components or the entire machine (Siener & Aurich, 2011). The predictive maintenance market has been growing in recent years, with a report by Kondyurin (2022) predicting its growth from 6.9 billion USD in 2021 to 28.2 billion USD by 2026. The amount of data generated in this sector increases significantly yearly (IOT Analytics, 2021).

As part of Industry 4.0, industrial maintenance must be updated with "smart" methods. This new industrial maintenance era is commonly called "Maintenance 4.0" (Matyas, 2022). One of the major problems with traditional maintenance strategies is their one-sided perspective, such as condition-based maintenance. Additionally, they lack a comprehensive maintenance strategy, such as "Total Productive Maintenance." The

need for timely data on machine conditions makes maintaining machines at optimal operating times and wear levels difficult, which should align with production requirements and desired quality. An integrated and predictive maintenance approach requires a holistic view of the machine, product, and process. With such a strategy, pre-calculated machine life calculations can be compared in real-time with the actual wear of machine components during production and adjusted accordingly. A suitable model can help determine maintenance time points, product quality, and energy consumption based on a combined view of condition monitoring data, quality control data, data from previously recorded failure patterns, and machine loads (Matyas, 2022).

In today's advanced technological environment, embedded systems have enabled advanced networking between the internet and each other. This trend is causing a convergence between the physical and digital domains, leading to the emergence of "Cyber-Physical Systems" (CPS). This concerns explicitly "Cyber-Physical Production Systems" (CPPS) in the industrial maintenance environment. CPPS represent the integration of embedded systems with various processes, including production, logistics, engineering, coordination, and management. These systems capture physical data directly from sensors while actuators intervene in the physical processes. Digital networks connect these systems, giving them access to globally available data and services. CPPS also have multimodal human-machine interfaces, making them open sociotechnical systems and enabling various new functions and services for the industrial maintenance sector (Kagermann et al., 2013).

In their paper, Ansari et al. (2019) state that the emergence of CPPS as Industry 4.0 technology is triggering a paradigm shift from descriptive to prescriptive maintenance. In this context, Ansari et al. (2019) introduce PriMa[5], a novel prescriptive maintenance model for CPPS. Marques & Giacotto (2019) describe in their 2019 study prescriptive maintenance as a technology that provides real-time adaptive suggestions about tasks to be performed with the help of AI. The distinction between predictive maintenance and prescriptive maintenance can be defined as follows: whereas predictive maintenance is based on the analysis of data patterns and trends to predict failure, prescriptive maintenance also takes into account the maintenance process of the

[5] Prescriptive Maintenance Model

respective organization and thus provides suggestions and supports the solution-finding process (Marques & Giacotto, 2019).

Ansari et al. (2019) propose a four-layer structure for PriMa, comprising data management, a predictive maintenance toolbox, a recommender, and a decision support dashboard. On top of this, they identify an additional overarching layer for semantic-based learning and reasoning. The PriMa method is designed to enhance the processing of large, heterogeneous datasets and to facilitate the generation of decision-supporting measures for the optimization of maintenance plans (Ansari et al., 2019). The following Figure 4 illustrates the overall architecture of Prima, introduced by Ansari et al. (2019).

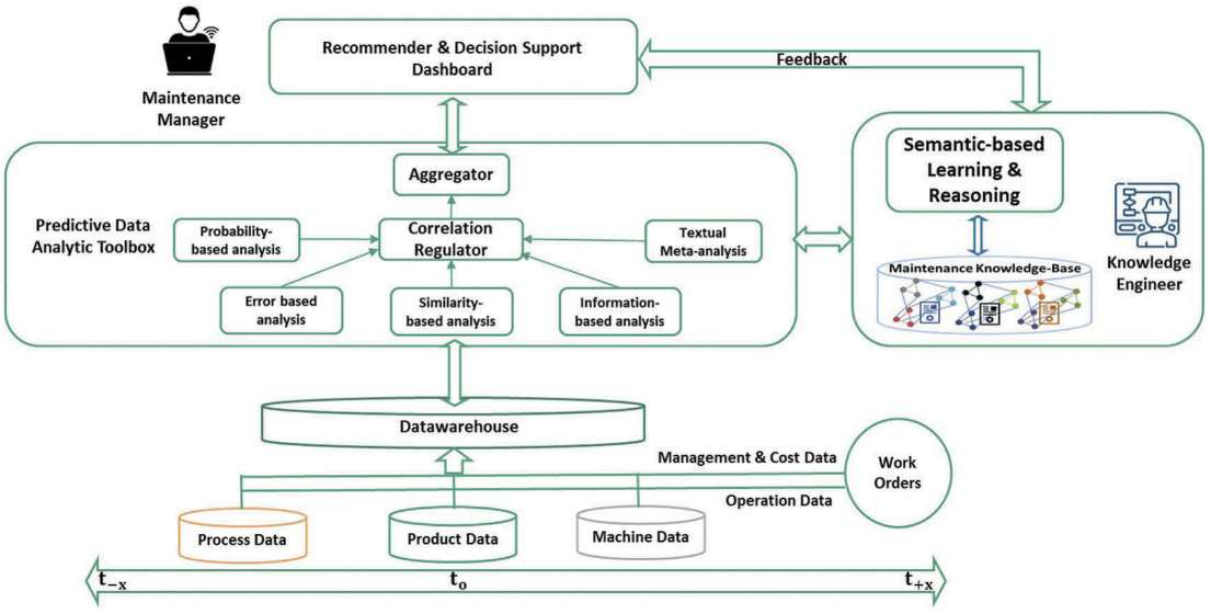



**Figure 4** Overall Architecture of PriMa (Ansari et al., 2019)

The data management layer, which is based on a scalable data warehousing solution, continuously collects maintenance data, which primarily consists of management, cost, and operational data from machines, processes, and products. This data can be mapped with both the horizontal and vertical data flow of the CPPS and all participating processes and components (e.g., maintenance managers, engineers, MES (Manufacturing Execution System), ERP (Enterprise Resource Planning)). One of the principal attributes of PriMa is its capacity to accommodate the multimodality of maintenance data. As a result, various aspects of maintenance are considered, and several influencing factors can be linked in order to obtain a comprehensive picture of the system status and thus generate new maintenance knowledge (Ansari et al., 2019).

## 2.2 Technical Language Processing (TLP)

In the industrial maintenance sector, while structured sensor data is primarily used for data analysis, text, as a form of unstructured data, also plays a significant role. The experience and knowledge of professionals are often documented but need to be formalized and processed to ensure the efficient use of this data. For this reason, natural language data is not considered in many cases (May et al., 2022). Section 2.2 provides a comprehensive analysis of the significance of TLP, which is an adaptation of NLP in the context of engineering and industry. Brundage et al. (2021) introduced in their 2021 paper the term TLP as a domain-driven approach to use NLP in a technical engineering setting. In their 2021 paper, Brundage et al. (2021) introduced the term TLP as a domain-driven approach to using NLP in an engineering environment. Brundage et al. (2021) define TLP as an iterative process involving human input to optimize NLP tools on engineering data. The underlying methodologies are therefore identical for NLP and TLP. Consequently, the most relevant NLP methods for the domain of industrial maintenance are introduced in Section 2.2.1.

According to Quarteroni (2018), NLP is a subfield of AI that primarily uses machine learning techniques to analyze natural language text. NLP has a long history, with significant milestones, such as developing the first search engine, namely SMART[6], to allow natural language queries in 1960 (Salton et al., 1975). The scientific literature shows that NLP has made significant progress using deep neural networks (DNNs) in the last few years. Before the introduction of the transformer in 2016 by Vaswani et al. (2017), mainly recurrent neural networks (RNNs) and long short-term memory (LSTM) models were used for NLP applications. The state-of-the-art models for NLP, especially the transformer, are introduced in Section 2.3 of this thesis.

### 2.2.1 NLP Methods for Industrial Maintenance

NLP is widely used in both research and practical applications. It covers various methods commonly employed for specific applications, such as part-of-speech tagging (POS), named entity recognition (NER), sentiment analysis, text classification and text generation.

[6] System for the Mechanical Analysis and Retrieval of Text

Event extraction is an essential application of NLP in the industrial maintenance sector. Event extraction in natural language text can be used to identify maintenance-related events, trends, and patterns.

Natural language data is mainly generated in logbooks for failure documentation in industrial maintenance. Gandomi & Haider (2015) show in their paper that approximately 80-90% of all business-relevant data is only available in unstructured form, mainly text data. AI techniques can be integrated into human-generated documents and reports to transform unstructured data into an automated prediction and reasoning framework. An AI-based method is essential to maximize the value of textual data in logbooks. AI methods can extract and convert all relevant textual information into a structured format, providing valuable and structured insights for better decision-making (Ansari et al., 2021).

Logbooks typically contain free text fields that are filled with non-standard text content. These texts may contain incomplete information and various errors that affect data quality (Akhbardeh et al., 2020). Machine operators' use of individual jargon in logbooks negatively impacts the data quality (Sexton & Fuge, 2019). The practical application of NLP in industrial maintenance is limited by the lack of pre-trained models specifically trained on the jargon and language patterns prevalent in this domain. As a result, the adoption and use of NLP methods in industrial maintenance tasks have not reached their full potential (Akhbardeh et al., 2020).

The following paragraphs present the most common NLP methods used in the industrial maintenance sector. In recent years, **text classification** has become increasingly important in industrial maintenance, enabling efficient analysis of large volumes of textual data. According to Kowsari et al. (2019), most text classification applications can be divided into four phases, see Figure 5: Feature extraction, dimension reduction, classifier selection, and evaluations.

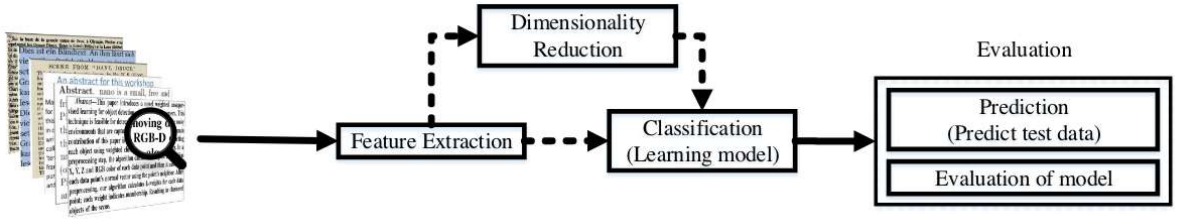



**Figure 5** Text Classification Pipeline (Kowsari et al., 2019)

The unstructured text must be transformed into a structured feature space in the feature extraction phase through data cleaning and formal feature extraction methods. In the dimensionality reduction phase, techniques are applied to the structured data, for example, converting all text to lowercase. The optimal classifier for the classification task is selected in the classifier selection phase. The Naïve Bayes Classifier (NBC) is very common. In the final stage of the classification pipeline, the evaluation phase, the classification model is evaluated, and various evaluation methods can be used. The accuracy evaluation is most commonly used (Kowsari et al., 2019).

An example of classification application in industrial maintenance can be found in a case study by Edwards et al. (2008). This case study aimed to classify repair jobs from a dam pump maintenance logbook as planned or unplanned. This case study aimed to investigate whether it is possible to classify natural language text into structured attributes to make business decisions based on data instead of best guesses (Edwards et al., 2008).

In 2018, when the paper “Natural Language Processing for Industrial Applications” by Quarteroni (2018) was published, a notable trend in using NLP in the industrial sector was the application of **conversational agents**. In contrast to conventional conversational agents designed to carry on a conversation, conversational agents in the industrial sector are used as task-oriented dialogue systems that cooperate with operators to complete a specific task. Conversational agents are primarily used in the industry to provide interactive systems to customers and reduce the cost of human agents (Quarteroni, 2018).

For example, a conversational agent could be used in a car manufacturer’s service hotline. In this scenario, the conversational agent would handle pre-defined service cases, while human agents can focus on the more complex and individual service requests.

The popularity of smart home devices such as Amazon Echo™ and Google Home™ has increased people’s awareness of the benefits of conversational agents. This has opened other potential applications for conversational agents in the industrial sector. One application uses conversational agents as shortcuts for specific industrial processes or services. For example, conversational agents can automate the booking work equipment such as cars (Quarteroni, 2018).

**Information extraction** is the most essential NLP method for this thesis and one of the most relevant methods in industrial maintenance. Information extraction automatically detects and extracts certain information from natural language text. Information extraction can be classified into closed-domain and open-domain information extraction. In closed-domain information extraction, the model is aware of the information it needs to extract. By contrast, an open-domain information extraction model does not know which information to extract. The model focuses on extracting new and unexpected information from texts (J. Liu et al., 2021).

Conventional information extraction models are usually trained using supervised learning methods. The problem is that such models can only detect information if the model has learned it. Furthermore, supervised learning is very time-consuming and, therefore, expensive. Zero-shot information extraction models can address these limitations of conventional information extraction models. H. Zhang et al. (2022) define zero-shot information extraction as a problem where a document consisting of one or more sentences and a list of information is known. The zero-shot information extraction model must now extract information in the document from the known list of information without prior annotation in the training phase (H. Zhang et al., 2022). Figure 6 below illustrates the extraction of zero-shot information. The information passed to the model in this example are Leakage, Oil change, and Noises.

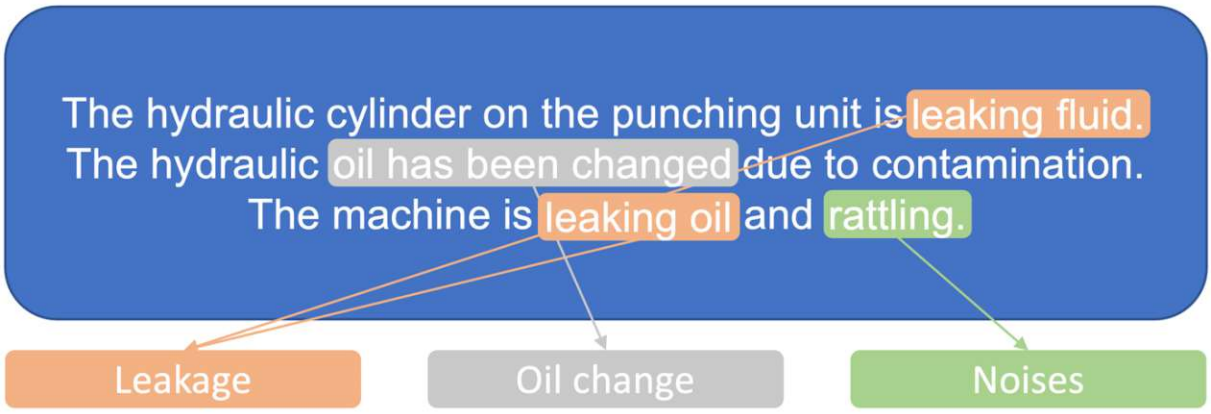



**Figure 6** Zero-Shot Information Extraction Task Demonstration

## 2.3 Transformer Models

In recent years, significant progress has been made in NLP, primarily through the application of transformer models (H. Liu et al., 2020). Before introducing the transformer model in this thesis, it is important to define deep neural networks (DNNs) and the attention mechanism in Sections 2.3.1 and 2.3.2.

### 2.3.1 Deep Neural Networks (DNNs)

In general, neural networks in the context of AI are models that connect artificial neurons, also called nodes, together. Each of these neurons performs simple calculations, but together, they form a complex model that can solve different tasks (Shah et al., 2018). Various definitions of DNNs can be found in AI in the scientific literature. Yi et al. (2016) use the term DNN to refer to neural networks with a network depth greater than or equal to four. In contrast, Shah et al. (2018) refer to a DNN when the number of hidden layers of neurons is greater than one. In this thesis, the term DNN refers to the definition of Shah et al. (2018). The depth of a DNN corresponds to the number of hidden layers of neurons, as shown in Figure 7 (Yi et al., 2016).

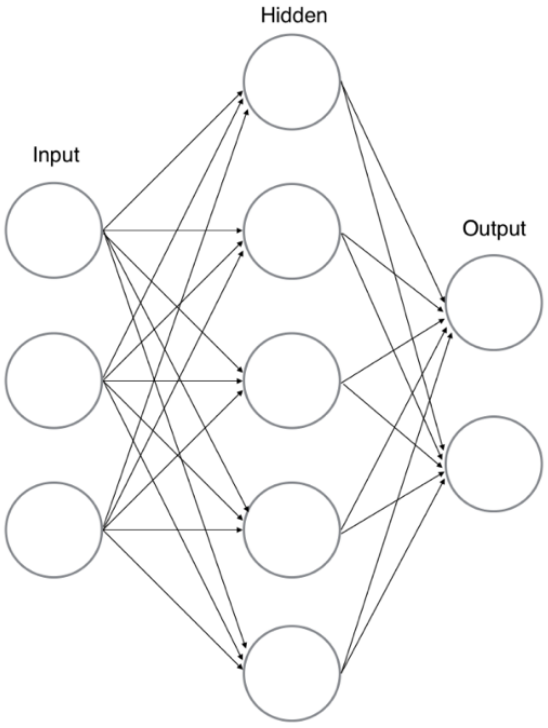



**Figure 7** Structure of a DNN with Depth 1 (Lopez & Kalita, 2017)

DNNs have become increasingly important due to their empirical success in various fields of AI, especially in NLP (Barredo Arrieta et al., 2020). DNNs have successfully solved complex learning problems (Lopez & Kalita, 2017). The structure of DNNs is inspired by nature, namely the mammalian visual system. This visual system contains many layers of a neural network. In this visual system, information is processed layer by layer from the retina to the visual centre (Serre et al., 2007). The layers in a DNN are often referred to as embedded layers in the scientific literature. The information in DNNs is processed, as in the mammalian visual system, resulting in a progressive abstraction of the input information. The number of embedded layers in a neural network positively correlates with extracting complex and abstract features from the data (Shah et al., 2018). Cohen et al. (2019) have shown that DNNs outperform usual neural networks, as DNNs improve feature abstraction with each additional embedded layer.

### 2.3.2 Attention Mechanism

The Google paper that introduced the transformer model in 2017 by Vaswani et al. (2017) is titled “Attention Is All You Need”. To understand the architecture of the transformer model, it is essential first to understand the attention mechanism based on the human visual system. In human vision, only a tiny part of the image can be focused, and this small area is called the fovea. The rest of the image, which is out of focus, is called the periphery. To enable humans to identify the desired information from the image, the fovea focuses on different significant parts until it has enough information to recognize the image. This feature extraction process is called the visual attention mechanism (Alpaydin, 1995). In the context of AI, the attention mechanism stores a group of hidden information; the size of this information depends on the size of the model input. The attention mechanism selects this information so that, in the context of the other information, it obtains a holistic understanding of the input, like the human visual mechanism, which requires several significant parts of the image to recognize it. The attention mechanism dynamically selects which positions of the input it focuses on, and it adjusts the weights assigned to each position. This method allows the model to vary the memory length to scale itself for different complex applications (Kim et al., 2017). Transformer models utilize a special type of attention mechanism called self-attention. Using self-attention, the positions in the same sequence can be related to

themselves. As a result, self-attention focuses on the relationships between positions in the same input sequence (Vaswani et al., 2017). By applying a self-attention mechanism to a text, the single words in this text are related to other words so that the model can capture complex relationships.

### 2.3.3 The Transformer

The transformer architecture belongs to the category of DNNs. Google introduced it in 2017, and it is the first transduction model based entirely on the self-attention mechanism, without using RNNs or convolution. The transformer architecture is based on an encoder-decoder structure. The encoder takes an input sequence $(x_1, \dots, x_n)$ of symbol representations and generates a continuous output sequence $\boldsymbol{z} = (z_1, \dots, z_n)$ that can be processed by the following transformer operations. The decoder gets $\boldsymbol{z}$ as input and sequentially computes the output $(y_1, \dots, y_m)$. At each successive step, the model uses the output of the previous steps and considers it in calculating the new step. This characteristic is called auto-regression. (Vaswani et al., 2017). Figure 8 shows the transformer architecture. The encoder (left half) and decoder (right half) use the self-attention mechanism.

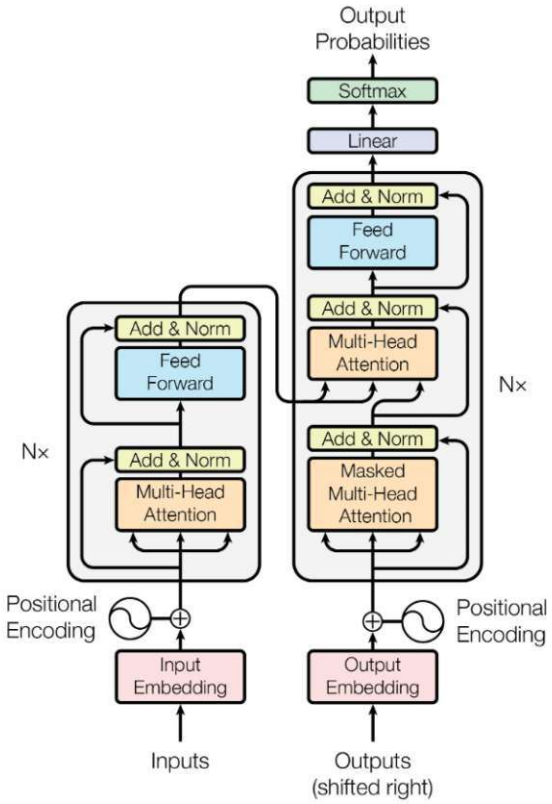



**Figure 8** The Transformer Model Architecture (Vaswani et al., 2017)

The encoder and the decoder consist of N = 6 identical layers. In the encoder, each layer has two sub-layers. The first sub-layer of the encoder is the multi-head self-attention mechanism, and the second sub-layer is a fully connected feed-forward network. In the decoder, each layer has three sub-layers. The second and third sub-layers are identical to those of the encoder. The first sub-layer is different from the encoder layers. The first sub-layer in the decoder is a masked multi-head attention layer that is applied to the encoder output. The result is always normalized after each sub-layer. The decoder output is linearly transformed, and the soft-max function is applied to the output. This converts the decoder output into the predicted probabilities for the next token (Vaswani et al., 2017).

### 2.3.4 Bidirectional and Unidirectional Transformer Models

Transformer models can be distinguished by how they process the input text, bidirectional or unidirectional (Alawneh et al., 2020). This section introduces the differences between bidirectional and unidirectional transformer models.

Bidirectional models, such as BERT, always have forward and backward layers (Alawneh et al., 2020). This implies that a bidirectional transformer model always considers the token on the left and the token on the right in the prediction. This bidirectional approach allows the model to understand better the input text (Devlin et al., 2019).

In comparison, unidirectional transformer models only have forward layers. They only interpret the input text in one direction, either left to right or right to left. An example of a unidirectional model are the GPT[7]-models, which process the text from left to right (Brown et al., 2020). Unidirectional models are particularly effective for text-generation tasks because they generate text sequentially (Devlin et al., 2019).

### 2.3.5 Encoder and Decoder Transformer Models

Generally, three types of transformer models can be distinguished: encoder-decoder models, encoder-only models, and decoder-only models. The functionality of the decoder and encoder layer in the transformer architecture has already been explained

[7] Generative pre-trained transformer

in Section 2.3.3. Therefore, it will not be described in more detail. Encoder-decoder transformer models are mainly used in sequence-to-sequence modelling (e.g., neural machine translation) (Brown et al., 2020). An encoder-only transformer model uses the encoder of the transformer architecture to represent the input. This encoder-only architecture is often used for NLU (Natural Language Understanding) tasks such as text classification (Sexton & Fuge, 2019). BERT, developed by Google, is one of the most well-known encoder-only models. A decoder-only transformer model is typically used for text-generation tasks (Devlin et al., 2019). GPT models by OpenAI are among the most well-known decoder-only models (Brown et al., 2020). The following two sections present the best-known bidirectional encoder, BERT, and the best-known unidirectional decoder, GPT-3.

Devlin et al. (2019) introduced 2019 BERT, Bidirectional Encoder Representations from Transformers. The novelty of BERT at the time of its presentation was its ability to learn deep bidirectional representations by pre-training on unlabeled text. BERT considers the left and right context in all layers of the model. The developers of BERT used “Masked-Language Modelling” (MLM) and “next sentence prediction” as pre-training objectives. This allows BERT to be easily fine-tuned for many different state-of-the-art NLP applications, such as question answering or named entity recognition. Labelled data is required to fine-tune the model to adapt BERT to specific downstream tasks. It is characteristic of BERT that it can be fine-tuned with only one additional output layer (Devlin et al., 2019). The architecture of BERT is very similar to the transformer architecture introduced by Vaswani et al. (2017). The main distinction is that BERT utilizes only the encoder component of the transformer architecture proposed by Vaswani et al. (2017).

In their paper, Devlin et al. (2019) presented two BERT models with different parameters, named $BERT_{BASE}$ and $BERT_{LARGE}$. In total, $BERT_{BASE}$ has 110 million parameters, and $BERT_{LARGE}$ has 340 million parameters. $BERT_{LARGE}$, with its significantly higher number of parameters compared to $BERT_{BASE}$, achieves slightly better performance in several benchmarks (Devlin et al., 2019).

Since 2019, many BERT-based models have been introduced and published in the literature. According to Y. Liu et al. (2019), BERT is significantly under-trained and subsequent models based on BERT typically outperform it. Two notable examples of models based on BERT are RoBERTa (Robustly optimized BERT approach),

developed by Y. Liu et al. (2019), and XLM (cross-lingual language model), developed by Lample & Conneau (2019). Both models, RoBERTa (Y. Liu et al., 2019) and XLM (Lample & Conneau, 2019), outperform BERT in standard performance tests. However, due to its popularity, BERT is still often used as a starting model for many tasks.

Brown et al. (2020) introduced GPT-3 in 2020, which stands for Generative Pretrained Transformer 3. GPT-3 is an evolution of GPT-2. The models share the same architecture. GPT-3 is based on an unidirectional and autoregressive decoder-only transformer that generates the output from left to right. GPT-3's self-attention mechanism and feed-forward neural networks make it particularly effective at interpreting long-term dependencies in input and generating text from them. When GPT-3 was introduced, it was outstanding for its many parameters. GPT-3 has 175 billion parameters (Brown et al., 2020). In comparison, $BERT_{BASE}$ has 110 million parameters (Devlin et al., 2019). Bidirectional models such as BERT have a higher fine-tuning efficiency for downstream tasks than unidirectional models such as GPT-3. The scientific literature currently lacks studies of models that combine the scale of GPT-3, which provides a better understanding of dependencies in human language, with a bidirectional approach that improves the fine-tuning capabilities of the model (Brown et al., 2020).

### 2.3.6 Transfer Learning in NLP

Transfer learning is an approach to optimize the training process of transformer models. Transfer learning allows the learned knowledge of an existing model, such as BERT, to be used as a base model for a new model with a different task. In this way, the training effort required to train a model adapted to a new application can be minimized (Briceno-Mena et al., 2022).

In particular, the transfer learning method ULMFiT (Universal Language Model Fine-tuning) introduced by Howard & Ruder (2018) enables effective transfer learning for all NLP applications with universal language models. The first step of ULMFiT is pre-training the general domain language model. In this phase, the model is pre-trained with unlabeled text, for example, from Wikipedia. The second step of the ULMFiT method marks the beginning of fine-tuning, focusing on domain adaptation. The pre-

trained language model is adapted to the required domain corpus in this step. As the model has already been pre-trained on a large dataset in step one, a small dataset is sufficient for this step. The third and last step of the ULMFiT method is the final fine-tuning step. In this step, the classifier of the model is fine-tuned with a small dataset. With the introduction of the ULMFiT framework in 2018, transformers could be easily adapted to individual NLP applications, making transformers state-of-the-art in NLP (Howard & Ruder, 2018).

# 3 State-of-the-Art Explainability of Transformer Models

Section 2.3 introduced the complex architecture of transformer models. One of the most challenging aspects of transformer models is that humans are unable to understand the inner workings of these complex deep-learning models. Du et al. (2019) demonstrate in their paper that numerous approaches to explain such models have already been developed, yet further research is required.

Consequently, this section presents the state-of-the-art on the explainability of transformer models through a systematic literature review. This chapter first presents the methodology of the systematic literature review. This process involves the identification of the search string, the definition of the exclusion criteria, and the screening of the papers. The objective of this state-of-the-art comparison is to identify the most appropriate explainability approach for the TEEE.

## 3.1 Methodology of the Systematic Literature Review

The systematic literature review is based on the strategy presented by Zonta et al. (2020) and Peres et al. (2020), which both conduct systematic reviews evaluating and summarizing the application of AI in the field of Industry 4.0. Figure 9 illustrates the methodology employed in the systematic literature review, which was used to identify the relevant literature for this study.

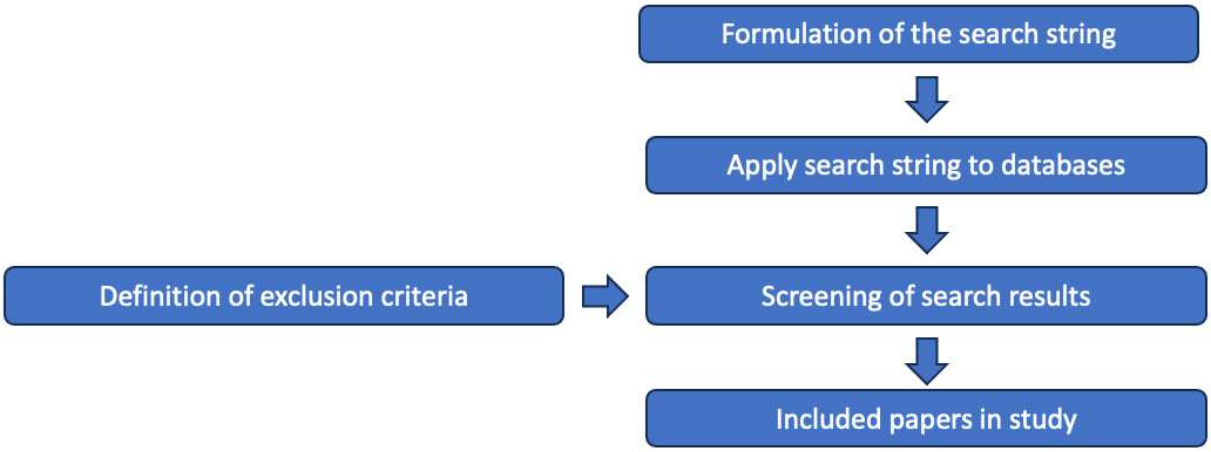



**Figure 9** Steps of Systematic Literature Review

Defining an appropriate search string for a systematic literature search is critical to any research project. The search string was constructed step by step using the most comprehensive database, Google Scholar. Initially, the term “transformer explainability” and its synonyms were applied to the database. Based on the first 30 results, the quality of the results for all synonyms was evaluated. This revealed that the optimal results were obtained with the terms “transformer explainability” and “transformer interpretability”. The initial two terms identified were then varied with synonyms of NLP and the results were evaluated once more. This procedure was then repeated with synonyms of “industrial maintenance” until the final search string in Figure 10 was determined and the quality of the results was correspondingly high.

Once successfully identified, the search string is applied to Google Scholar, IEEE Xplore, and Science Direct databases. This process requires careful selection and construction of the search string, primarily due to the limitations associated with applying the search string to these databases and the subsequent impact on the relevance of the search results. Specifically, these limitations can take the form of restrictions on the number of Boolean operators allowed per query, as illustrated by Science Direct, which accepts a maximum of eight Boolean operators. Similarly, Google Scholar imposes a character limit of 256 characters per query. Awareness of these limitations must influence the design of the search string to ensure that the most relevant literature is identified in line with the research objectives. The search string was primarily constructed using Google Scholar as the starting database. This choice was motivated by the large size of the Google Scholar database compared to IEEE Xplore and Science Direct, which provide a more comprehensive range of literature. The extraction from the databases was conducted on the 26th of February, 2023.

The first research question investigates the explainability of transformer models through a systematic literature review. Special attention is given to the explainability of transformer model predictions in the field of NLP for industrial maintenance applications.

A limited corpus of literature exists on the explainability of transformer models, with a particular focus on industrial maintenance. This limitation challenged the formulation of the search string, and consequently, this specific aspect was excluded from it. The decision to exclude it was made to avoid the limitations resulting in a lack of results from the systematic literature review. Therefore, it guarantees the inclusion of all

relevant literature concerning the industrial maintenance sector in the final search result. The comprehensiveness of this approach ensures an exhaustive review of the first objective of this thesis. The final search string is shown in Figure 10. A further restriction was introduced to the search to reduce the large number of search results and increase the density of relevant search results. In addition to the search string shown in Figure 10, the terms “explainability” or “interpretability” have to appear in the title of each paper. The remaining part of the search string must only appear in the abstract, keyword list, or heading for the article to be included.

transformer AND (explainability OR interpretability) AND (NLP OR "natural language processing")

**Figure 10** Search String

The search string was applied to the databases Google Scholar, IEEE Xplore and Science Direct, and the results were extracted. All databases are configured so only publications published after 2018 are returned as hits; this limitation was chosen in line with (Zonta et al., 2020). This restriction was implemented because research in the field of AI explainability, particularly regarding transformer models, is evolving rapidly. This ensures that the thesis is based on the state-of-the-art and does not rely on outdated methods or findings. Before starting the screening process, defining the exclusion criteria for the search results is necessary. These criteria play a crucial role in the screening process and help to identify and remove irrelevant resources. The following Table 4 lists these exclusion criteria.

**Table 4** Exclusion Criteria

| Exclusion criteria | Description |
|---|---|
| Criterion 0 | Only peer-reviewed publications were considered |
| Criterion 1 | Remove duplicates |
| Criterion 2 | Remove publications that only reference other publications |
| Criterion 3 | Remove publications that do not have transformer AND explainability in the title (or synonyms thereof) |
| Criterion 4 | Remove all publications that are not in English or German |
| Criterion 5 | Remove all publications that do not address the explainability of transformer models for NLP |

The first criterion, 0, excludes all publications which were non-peer-reviewed. Non-peer-reviewed publications are often at an early research stage and could prove inferior in scientific quality. Criterion 1 removes all duplicates to avoid unnecessary database expansion and ensures that each study included provides unique information. In the next exclusion step, criterion 2, all studies that only reference other publications are excluded. These studies do not contain new research results and are secondary sources that only reference other works, thereby not improving the quality of the database for this thesis. Criterion 3 includes only those publications that contain the terms “transformer” and “explainability”, or synonyms of these two terms, in the title. This reduces the likelihood of including papers only marginally related to the topic or considering it in a different context. Criterion 4 excludes all publications that are not written in English or German. The purpose of this exclusion criterion is to avoid possible misunderstandings or misinterpretations due to translation errors. After exclusion criterion 4, all abstracts of the remaining papers are reviewed. With criterion 5, as the last exclusion criterion, all publications not addressing the explainability of transformer models are excluded. This criterion ensures the relevance of the selected studies for the paper and allows a deeper analysis of the explainability of transformer models.

Figure 11 below visually outlines the screening process utilized in this thesis, detailing how search results from each database were processed and specifying the criteria used to exclude publications. After applying all six predetermined exclusion criteria, 34 relevant studies remained for further review in this thesis.

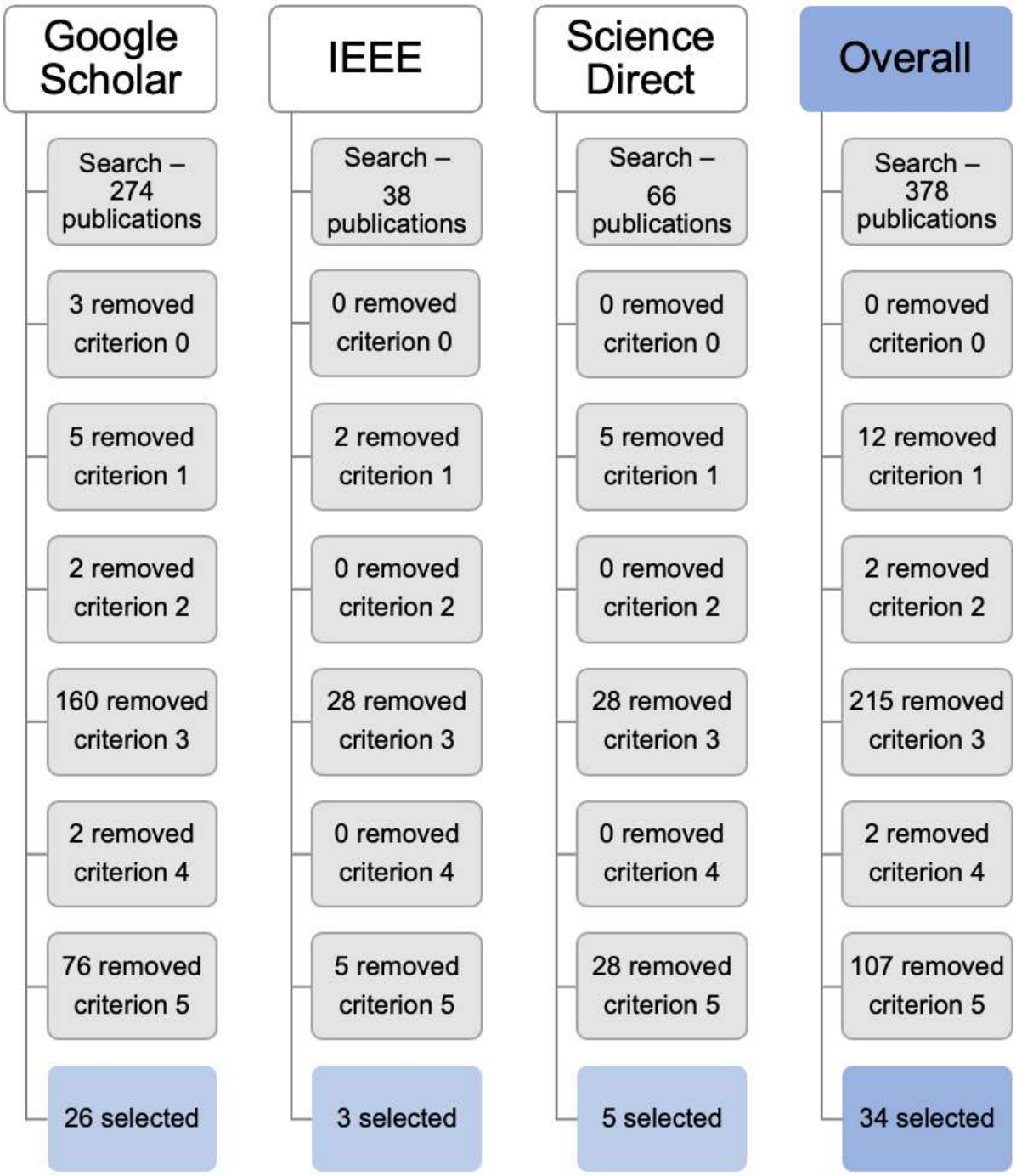



**Figure 11** Screening Process

Table 5 lists all publications included in the systematic literature review after applying the six exclusion criteria to the identified papers. In addition to the papers from the screening process, other relevant papers that are essential for understanding and gaining a comprehensive perspective on the explainability of transformer models were identified. These papers are presented in Table 16, located in the appendix of this thesis. Including these additional papers aims to ensure a broader perspective and increase the thesis' relevance. Table 5 provides a concise overview of each of the

selected papers, indicating whether they are related to industrial maintenance and the publication type and source of the initial selection.

**Table 5** Selected Publications from Screening

| Article | Aim of the paper | Maintenance related | Type | First selection source |
|---|---|---|---|---|
| (Attanasio et al., 2022) | The paper is a benchmark study for hate speech detection with transformer-based NLP models. | X | Conference Paper | Google Scholar |
| (Alammar, 2021) | The paper presents an open-source library, called Ecco, which increases transparency of NLP transformer models. | X | Conference Paper | Google Scholar |
| (Vashishth et al., 2019) | The paper researches the explainability of attention weights in NLP models. | X | Preprint | Google Scholar |
| (Zini & Awad, 2022) | The paper presents methods to explain the inner workings of deep networks, such as transformer models and a comprehensive investigation of these methods for NLP models. | X | Journal Article | Google Scholar |
| (Danilevsky et al., 2020) | The paper presents a taxonomy for the classification of explanations and identifies the most important explainability techniques. | X | Conference Paper | Google Scholar |
| (Naylor et al., 2021) | The paper demonstrates the state-of-the-art in the explainability of NLP AI models, such as transformer analyzes the concepts of explainability and interpretability through a case study of mortality prediction in clinical notes. | X | Preprint | Google Scholar |
| (L. Wang et al., 2022) | The paper introduces a benchmark to evaluate the interpretability of neural models covering three NLP tasks: sentiment analysis, textual similarity, and reading comprehension. | X | Preprint | Google Scholar |
| (Schwenke & Atzmueller, 2021) | The paper explains transformer models for time series classification. The authors present interpretation methods that use visualisations to reveal the attention patterns of transformer models. | X | Journal Article | Google Scholar |
| (Mylonas et al., 2022) | The paper presents a new technique that selects the most faithful attention-based interpretation method by combining different head, layer, and matrix operations. | X | Preprint | Google Scholar |
| (Du et al., 2019) | The paper presents a survey of the existing interpretability methods to increase the interpretability of machine learning. | X | Preprint | Google Scholar |
| (Dong et al., 2022) | The paper explores the relationship between the word saliency and the word properties to explain the predictions of NLP models. | X | Preprint | Google Scholar |
| (Madsen et al., 2023) | The paper presents a categorization of post-hoc interpretability methods and evaluates the value of each method for the human understanding. | X | Journal Article | Google Scholar |

| Article | Aim of the paper | Maintenance related | Type | First selection source |
|---|---|---|---|---|
| (Rychener et al., 2023) | The paper outlines the limitations of LIME[8] and SHAP when using complex BERT-based classifiers. | X | Conference Paper | Google Scholar |
| (Rychener et al., 2023) | The paper outlines the limitations of LIME and SHAP when using complex BERT-based classifiers. | X | Conference Paper | Google Scholar |
| (Molnar, 2021) | The paper presents a guide for making black-box AI models explainable. | X | Journal Article | Google Scholar |
| (Martindale & Stewart, 2021) | In the paper a transformer explainability package, called TX^2, for Jupiter Notebooks is developed. | X | Journal Article | Google Scholar |
| (Szczepański et al., 2021) | The paper presents an explainability method for BERT-based fake news detection models based on LIME and anchors. | X | Journal Article | Google Scholar |
| (Eyzaguirre et al., 2021) | The paper presents a modified BERT-based model, called DACT-BERT, which has an increased interpretability by adding an adaptive computation mechanism to the pipeline. | X | Journal Article | Google Scholar |
| (Arya et al., 2019) | The paper presents a toolkit with eight explainability methods, two evaluation metrics and a taxonomy of explainability methods. | X | Preprint | Google Scholar |
| (Turbé et al., 2022) | The paper presents an approach to evaluate the performance of interpretability methods for time series classification. | X | Preprint | Google Scholar |
| (Sen et al., 2020) | The paper compares human versus computational attention mechanism for text classification. | X | Conference Paper | Google Scholar |
| (Tamekuri et al., 2022) | The paper classifies open-data news documents by their theme and proposes an interpretability method for this use case. | X | Journal Article | Google Scholar |
| (Atanasova et al., 2020) | In the paper a method is developed for evaluating existing explainability methods for text classification. | X | Preprint | Google Scholar |
| (Tang et al., 2020) | This paper introduces an interpretability approach for event extraction that balances generalization and interpretability by jointly training a classifier and a rule decoder within an encoder-decoder architecture. | X | Conference Paper | Google Scholar |
| (Cheong et al., 2022) | This paper evaluates the explainability of traditional (XGBoost) and deep learning (LSTM with Attention) models on longitudinal healthcare data, using SHAP, LRP, and Attention. | X | Preprint | Google Scholar |
| (X. Li et al., 2022) | The paper presents a taxonomy for explainability methods and surveys the performance metrics of the methods. | X | Journal Article | Google Scholar |

[8] Local Interpretable Model-agnostic Explanations

| Article | Aim of the paper | Maintenance related | Type | First selection source |
|---|---|---|---|---|
| (Namatēvs et al., 2022) | The paper presents a taxonomy for explainability methods and presents the most important definitions in the context of AI explainability. | X | Journal Article | Google Scholar |
| (S. Liu et al., 2021) | In the paper two explainability methods, called AGrad and RePAGrad, are developed. They produce directional relevance scores based on the attention weights. | X | Conference Paper | IEEE Xplore |
| (Adadi & Berrada, 2018) | This paper provides an introduction for researchers and practitioners to XAI and outlining future research directions in the literature. | X | Journal Article | IEEE Xplore |
| (Z. Zhang et al., 2022) | This paper presents a transformer, originally trained on natural language, for sensor fusion tasks in industrial monitoring. The paper analyses the model interpretability with the attention mechanism. | ✓ | Journal Article | IEEE Xplore |
| (Barredo Arrieta et al., 2020) | This paper presents the challenges in the field of XAI and introduces a taxonomy for deep neural networks explainability. | X | Journal Article | Science Direct |
| (Jiao et al., 2022) | This paper presents a framework for bearing fault diagnosis using a transformer model. The paper analyzes the interpretability of the model prediction by using the attention mechanism. | ✓ | Journal Article | Science Direct |
| (Lu et al., 2022) | This paper introduces a framework, called ExpDEE, for document-level event extraction that increases the explainability by detecting references to sentence-level events during the extraction process. | X | Journal Article | Science Direct |
| (Guo et al., 2022) | This paper presents an attention network for tool wear monitoring in high-speed milling, which enhances both monitoring accuracy and interpretability. | ✓ | Journal Article | Science Direct |
| (Montavon et al., 2018) | This paper is an introduction for the explainability of the predictions of deep neural network models. The paper focuses on the LRP technique. | x | Journal Article | Science Direct |

## 3.2 Summary and Results

This chapter presents the results of the systematic literature review on the state-of-the-art of the explainability of transformer models, with a focus on their application in the domain of industrial maintenance. This chapter first presents the methodology used for the systematic literature review. This process includes the definition of the search string, the definition of the exclusion criteria, and finally the screening of the results. As part of the screening process, 34 scientific papers were identified that are relevant to

the state-of-the-art about the explainability of transformer models. The focus here is on the domain of industrial maintenance. As mentioned in Section 3.1, there is a limited body of literature on the explainability of transformer models, with a particular focus on industrial maintenance. Therefore, this aspect was not explicitly included in the definition of the search string in order to identify relevant state-of-the-art papers from other domains. Figure 12 shows that with the search string defined in Figure 10, a total of three papers were identified that address the explainability of transformer models in the context of maintenance.

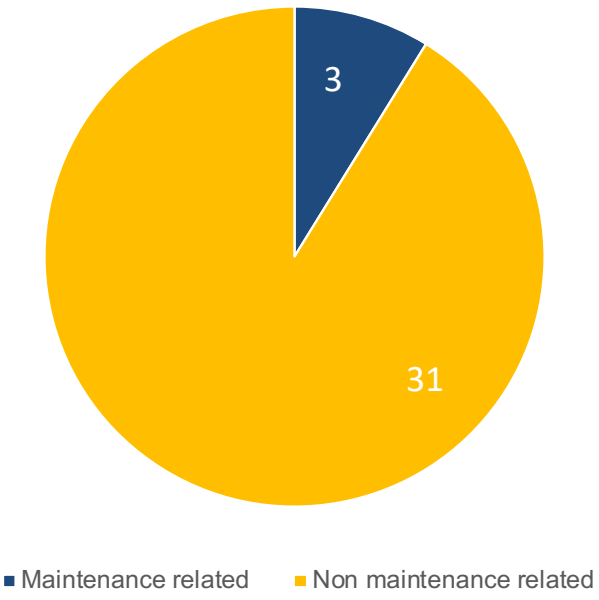



**Figure 12** Breakdown of Maintenance-Related Screening Results and Non-Maintenance-Related Screening Results

In their paper, Guo et al. (2022) present an attention network for tool wear monitoring in high-speed milling, which improves both monitoring accuracy and interpretability. Jiao et al. (2022) introduce a framework using a transformer model for bearing fault diagnosis and analyze the model's predictions for interpretability using the attention mechanism. Similarly, Z. Zhang et al. (2022) use a transformer-based model, trained with natural language, for a sensor fusion task. This model is tested on three datasets: a hydraulic system dataset, a bearing dataset, and a transmission dataset, with a focus on feature extraction. The interpretability of the model is analyzed using the attention mechanism (Z. Zhang et al., 2022).

These studies by Guo et al. (2022), Jiao et al. (2022), and Z. Zhang et al. (2022) are all maintenance-related and investigate interpretability through the attention mechanism. Conversely, other works that are not maintenance-related, such as the paper by Attanasio et al. (2022) on hate speech detection, conclude that SHAP and SOC provide more plausible and trustworthy explanations for predictions than gradient-based and attention-based methods. This conclusion is further supported by Turbé et al. (2022), which addresses the interpretability of neural networks in time series classification, finding that SHAP and Integrated Gradients offer the most effective explanations for their datasets.

This analysis demonstrates that all three maintenance-related papers rely on the attention mechanism for interpretability. However, the screening did not identify any papers that explain the predictions of transformer models in the field of industrial maintenance using post-hoc methods such as SHAP or LIME. In contrast, numerous other studies, such as those by Turbé et al. (2022) and Attanasio et al. (2022), illustrate that in other domains, post-hoc methods like SHAP provide the most effective explanations for AI predictions. Consequently, this state-of-the-art analysis highlights a significant research gap. While the attention mechanism is commonly used for the interpretability of transformer models in maintenance-related studies, there is a notable absence of studies applying post-hoc explainability methods such as SHAP or LIME to transformer model predictions in the field of industrial maintenance. In contrast, numerous studies in other domains demonstrate the superiority of post-hoc methods like SHAP for providing effective explanations for predictions. This gap indicates the contribution of this thesis to the scientific literature: the explainability of transformer model predictions in industrial maintenance using a post-hoc method, namely SHAP.

# 4 Transformer Models for Zero-Shot Event Extraction in Industrial Use-Cases

The development of transformer models, such as BERT or GPT models, is a complex and computationally intensive process (Dankar et al., 2023). This is due to the necessity of immense computational resources, extensive data preprocessing, and robust infrastructure to effectively manage and train their high number of parameters. With the introduction of the transformer by (Vaswani et al., 2017), a transformer translation model was trained from scratch. This model was trained on numerous sentence pairs in different languages. However, in most use cases for NLP with transformer models, there is too little or no annotated data to train a model in this way (Tunstall, 2022). Also, for industrial applications, the annotated data required to train a transformer model in this way must be improved. In addition, implementing transformer models would be very complex and resource-intensive, making them from scratch-implementation unattractive for industrial use. The solution to these challenges is the concept of transfer learning, introduced in Section 2.3.6, which enables fine-tuning of pre-trained models to adapt to specific tasks with less data and fewer computational resources.

In the first Section, 4.1, of this chapter, the Hugging Face Transformers library is introduced, followed by the presentation of four selected models from this library that have been specifically optimized for zero-shot event extraction. Section 4.2 then evaluates the event extraction capabilities of these four models within the realm of industrial maintenance scenarios, addressing Research Question 2 (RQ2). The model exhibiting superior performance in this evaluation, detailed in Section 4.2.2, is chosen as the foundational model for the TEEE, the artefact of this thesis, introduced in Section 4.4.

Following the model selection of TEEE's event extraction capabilities, the next step involves identifying the most suitable explainability framework for explaining the predictions made by the applied transformer model. To address this sub-goal, in Section 4.3 the relevant explainable artificial intelligence (XAI) model terminology is introduced. Thereafter, Section 4.3.2 presents the dimensions of explainability for transformer models, setting the groundwork for the introduction of a taxonomy for transformer model explainability for event extraction in Section 4.3.3. Utilizing this

taxonomy, the most effective method for elucidating the predictions of the TEEE is determined.

## 4.1 Hugging Face Transformer Models

After transformers became very popular in the field of NLP in 2018 through the ULMFiT[9] framework (Howard & Ruder, 2018), it was difficult for different research institutes to develop their transformer models in different frameworks (e.g., PyTorch or TensorFlow). This made it also difficult for companies and others interested in NLP to use the transformer models developed by large institutions such as Google or OpenAI in practical applications. The open-source library developed as Hugging Face Transformers could resolve these problems (Tunstall, 2022). The Hugging Face Transformers library offers a variety of pre-trained models for different NLP applications. Therefore, fine-tuned multilingual models for zero-shot event extraction are used for the practical part of this thesis. The following section presents and evaluates the four most exciting models for zero-shot event extraction in English and German from the Hugging Face Transformers library for the industrial use case in this thesis (Tunstall, 2022).

The **model mDeBERTa-v3-base-mnli-xnli** (referred to as Transformer Model 1) is based on a model introduced by Microsoft called DeBERTaV3 (He et al., 2023), which was pre-trained on the CC100 dataset by Conneau et al. (2020). Laurer et al. (2022) subsequently fine-tuned the mDeBERTa model for the downstream task of multilingual zero-shot event extraction. For fine-tuning, they used the XNLI (Conneau et al., 2018) and MNLI (Williams et al., 2018) datasets (Laurer et al., 2022).

The **model multilingual-MiniLMv2-L6-mnli-xnli** (referred to as Transformer Model 2) is based on the MiniLM-L6 (W. Wang et al., 2020) model by Microsoft and was distilled from the XLM-RoBERTa-large (Conneau et al., 2020) model. Distillation in the AI context means that a smaller model, in this case, MiniLM-L6, has been trained to imitate the behaviour of a larger model (XLM-RoBERTa-large). The main advantage of distillation is that the smaller model can run faster and use less computational resources than the larger language model while trying to maintain the performance of the larger model (Hsieh et al., 2023). Laurer et al. (2022) subsequently fine-tuned the

[9] Universal Language Model Fine-tuning

MiniLM-L6 model for the downstream task of multilingual zero-shot event extraction. For fine-tuning, they used the XNLI (Conneau et al., 2018) and MNLI (Williams et al., 2018) datasets (Laurer et al., 2022).

The **model xlm-v-base-mnli-xnli** (referred to as Transformer Model 3) is based on a model introduced by Meta AI called XLM-V-base (Liang et al., 2023), which was pre-trained on the CC100 dataset by Conneau et al. (2020). Laurer et al. (2022) subsequently fine-tuned the XLM-V-base model for the downstream task of multilingual zero-shot event extraction. For fine-tuning, they used the XNLI (Conneau et al., 2018) and MNLI (Williams et al., 2018) datasets (Laurer et al., 2022).

The **model ernie-m-large-mnli-xnli** (referred to as Transformer Model 4) is based on a model introduced by Meta AI called RoBERTa (Y. Liu et al., 2019), which was pre-trained on the CC100 dataset by Conneau et al. (2020). Ouyang et al. (2021) subsequently fine-tuned the RoBERTa model for the downstream task of multilingual zero-shot event extraction. For fine-tuning, they used the XNLI (Conneau et al., 2018) and MNLI (Williams et al., 2018) datasets. The ERNIE-M model outperforms similar RoBERTa models of the same size (Laurer et al., 2022).

## 4.2 Evaluation of Transformer Models for Event Extraction in Industrial Maintenance

### 4.2.1 Evaluation Methodology

In this section, the four models presented in the previous section are evaluated in terms of how they perform in the downstream task of zero-shot event extraction in industrial maintenance. The models were selected from a total of 14 models that support both English and German and are fine-tuned for this task in the Hugging Face Transformer library (as of 30.06.23). The four models that were chosen are mDeBERTa-v3-base-mnli-xnli, multilingual MiniLMv2-L6-mnli-xnli, xlm-v-base-mnli-xnli, and ernie-m-large-mnli-xnli. These models have been chosen because they are based on foundational models from well-known organizations and are all state-of-the-art. These models are directly relevant to the objective of this thesis, as they having been explicitly developed for zero-shot event extraction using different approaches to address this problem. The

evaluation of all 14 models is irrelevant to this thesis, as the four selected models are a representative sample of the 14 models.

Two different **evaluation datasets** have been used to gather relevant benchmarks to evaluate the models. One of the datasets, comprising 40 entries, contains one event for each sentence, while the other dataset, also comprising 40 entries, always contains two events per sentence. Both datasets were translated into English to evaluate the models' ability to process English and German texts. This results in four different datasets on which the models are evaluated. These datasets are fictitious and are related to industrial maintenance. However, it closely resembles scenarios that can occur within industrial settings, which provides a credible context for assessing the models. This ensures that the evaluation is based on a context reflecting possible real-world applications, even though the data is not from real-world applications. The $Benchmark\_test\_german$ dataset, written in German, assigns one specific event to each entry, while the $Benchmark\_test\_english$, written in English, follows the same pattern. The $Benchmark\_test\_german\_multiple$ dataset assigns two specific events to each entry, while the $Benchmark\_test\_english\_multiple$ dataset also follows the same pattern. The $Benchmark\_test\_german$ and $Benchmark\_test\_english$ datasets each contain six unique events and comprise 40 observations each. In contrast, the $Benchmark\_test\_german\_multiple$ and $Benchmark\_test\_english\_multiple$ datasets each have twelve unique events and comprise 80 observations each. In Table 6 below are some examples of one entry from each dataset.

**Table 6** Example Entries with Corresponding Events from the Various Datasets

| Dataset | Entry | Event |
|---|---|---|
| $Benchmark\_test\_german$ | Die Drehmaschine hat plötzlich aufgehört zu arbeiten. | Maschinenstopp |
| $Benchmark\_test\_english$ | The late has suddenly stopped working. | Machine stop |
| $Benchmark\_test\_german\_multiple$ | Die Maschine hat den Betrieb wieder gestoppt, und während des Betriebs wurden übermäßige Vibrationen beobachtet. | Maschinenstopp, Übermäßige Vibrationen |
| $Benchmark\_test\_english\_multiple$ | The machine has stopped operation again, and excessive vibrations were observed during operation. | Machine stop, Excessive vibration |

The evaluation of a language model, particularly in the context of classification and event extraction, is often assessed using **accuracy** as the primary metric. Accuracy is

the percentage of correct model predictions out of all model predictions, as shown in Equation 1 (Hossin & M.N, 2015).

**Equation 1** Accuracy (Sokolova et al., 2006)

$$Accuracy = \frac{tp + tn}{tp + fp + fn + tn}$$

An accuracy of 1 means that all the predictions made by the model are correct, while an accuracy of 0 means the exact opposite. In this case, these are the correct extracted events in a dataset unknown to the model.

The following accuracy values are determined for the four models using the four datasets as follows: Accuaracy_single_event_german, Accuaracy_single_event_english, Accuaracy_multiple_event_german, Accuaracy_multiple_event_english.

The **F1 score** is another important metric for evaluating several NLP downstream tasks; the formula is shown in Equation 2.

**Equation 2** F1 Score (Sokolova et al., 2006)

$$F1\ Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

The F1 score is the harmonic mean of a model's precision and recall. The precision of a model is the ratio of true positive predictions to the sum of true and false positive predictions. Therefore, the precision of a model indicates how many of the positive predictions are actually positive. The recall is the ratio of true positive predictions to the sum of true positive and false negative predictions. Recall indicates how many of the true positive predictions were correctly predicted by the model.

The following accuracy values are determined for the four models using the four datasets as follows: F1_single_event_german, F1_single_event_english, F1_multiple_event_german, F1_multiple_event_english.

The **model runtime** is the computational time the model requires to perform its task on a given dataset. In the context of this thesis, it is the time a model takes to perform event extraction on the evaluation datasets. This evaluation metric is crucial for this thesis as it allows the evaluation of the model's efficiency. For this thesis, a model with a high accuracy and F1 score but a very high runtime would not be optimal, as quick decisions are required, especially in industry. Conversely, models with a slightly lower

accuracy and F1 score but a much better runtime are usually the best choice for most applications.

The model runtime for the four selected models is calculated for each of the four datasets:

$$\text{Runtime\_single\_event\_german}, \text{Runtime\_single\_event\_english},$$
$$\text{Runtime\_multiple\_event\_german}, \text{and } \text{Runtime\_multiple\_event\_english}.$$

To determine the model runtime, the computing power of the hardware on which the model runs is decisive. For this evaluation, model runtimes are measured on a 2018 Apple MacBook Pro with a 6-core Intel i7 processor clocked at 2.6GHz, 16GB of DDR4 memory, and a Radeon Pro 560X graphics card with 4GB of GDDR5 memory.

### 4.2.2 Evaluation Results

In this section, the models are evaluated quantitatively and qualitatively on four fictional industrial maintenance datasets.

Quantitative Results: The following Table 7 and Table 8 show the evaluation results of the four models for the downstream tasks of zero-shot event extraction with industrial maintenance datasets. In the interest of clarity, the evaluation results of the models have been split into two separate tables. Each table shows the scores for two of the models. Table 7 presents the quantitative evaluation results of the transformer model 1 and 2.

**Table 7** Quantitative Evaluation Transformer Models 1 & 2

| | **Transformer Model 1** | **Transformer Model 2** |
|---|---|---|
| Base Model Name | DeBERTaV3 base | XLM-RoBERTa-large |
| Base Model pretraining Dataset | CC100 multilingual | 2.5TB of filtered Common Crawl data |
| Fine-tuned Model Name | MoritzLaurer/mDeBERTa-v3-base-mnli-xnli | MoritzLaurer/multilingual-MiniLMv2-L6-mnli-xnli |
| Fine-tuned Model pretraining Datasets | XNLI, MNLI | XNLI, MNLI |
| Accuaracy_single_event_german | 0,85 | 0,68 |
| Accuaracy_single_event_english | 0,98 | 0,75 |
| Accuaracy_multiple_event_german | 0,16 | 0,38 |
| Accuaracy_multiple_event_english | 0,55 | 0,50 |
| F1_single_event_german | 0,85 | 0,68 |
| F1_single_event_english | 0,97 | 0,76 |
| F1_multiple_event_german | 0,13 | 0,34 |
| F1_multiple_event_english | 0,54 | 0,48 |
| Runtime_single_event_german | 424,16 seconds | 26,19 seconds |
| Runtime_single_event_english | 447,88 seconds | 27,55 seconds |
| Runtime_multiple_event_german | 794,77 seconds | 46,83 seconds |
| Runtime_multiple_event_english | 941,55 seconds | 68,98 seconds |

Table 8 presents the quantitative evaluation results of transformer model 3 and 4.

**Table 8** Quantitative Evaluation Transformer Models 3 & 4

| | **Transformer Model 3** | **Transformer Model 4** |
|---|---|---|
| Base Model Name | XLM-V-base model | RoBERTa |
| Base Model pretraining Dataset | 2.5TB of filtered Common Crawl data | CC100 |
| Fine-tuned Model Name | MoritzLaurer/xlm-v-base-mnli-xnli | MoritzLaurer/ernie-m-large-mnli-xnli |
| Fine-tuned Model pretraining Datasets | XNLI, MNLI | XNLI, MNLI |
| Accuaracy_single_event_german | 0,93 | 1,00 |
| Accuaracy_single_event_english | 0,90 | 0,94 |
| Accuaracy_multiple_event_german | 0,14 | 0,22 |
| Accuaracy_multiple_event_english | 0,55 | 0,55 |
| F1_single_event_german | 0,92 | 1,00 |
| F1_single_event_english | 0,90 | 0,94 |
| F1_multiple_event_german | 0,11 | 0,22 |
| F1_multiple_event_english | 0,54 | 0,53 |
| Runtime_single_event_german | 135,00 seconds | 478,27 seconds |
| Runtime_single_event_english | 137,71 seconds | 420,450seconds |
| Runtime_multiple_event_german | 249,74 seconds | 841,77 seconds |
| Runtime_multiple_event_english | 332,92 seconds | 965,70 seconds |

**Qualitative Results:** The qualitative analysis aims to compare the strengths and weaknesses of each model in practical applications. The numerical results of the

quantitative results and the conclusions of the qualitative analysis are used to find the overall best model for the use case of zero-shot event extraction for the industrial maintenance sector.

The transformer models employed in this thesis are based on widely used models, such as DeBERTaV3, XLM-RoBERTa, XLM-V, and RoBERTa. To ensure a uniform benchmark test, the models underwent fine-tuning on identical datasets, namely XNLI and MNLI.

When evaluating the accuracy and F1 scores, Model 4, in particular, stands out with the highest F1 scores and accuracy levels for the extraction of single events in German and English. Model 3 demonstrates good F1 scores and accuracy in extracting single events in German and English. However, it produced the worst results in extracting multiple German events. Model 2 obtains the lowest F1 scores and accuracy values on average, but it can achieve the most stable results across all aspects, including German, English, and single or multiple event extraction. Model 2 achieved the highest F1 score and accuracy for multiple event extraction in German. Model 1 demonstrates good F1 scores and accuracy in extracting single events in German and English. However, the model did not achieve a good F1 score or accuracy in extracting multiple German events.

The models should be highly efficient in processing German and English use cases, and extracting single or multiple events. The quantitative evaluation results show significant differences in performance between the four models for extracting single and multiple events. All four models have the worst accuracy and F1 score performance when extracting multiple German events. Nevertheless, this is a critical aspect of this thesis, as most maintenance texts in the German-speaking area are in German and contain multiple events per sentence.

Looking at the runtimes of the models, it is evident that Model 2 outperforms the rest in both languages and tasks, making it the most effective among the analyzed transformer models.

**Model Selection:** Based on the presented qualitative and quantitative results, selecting the best model for zero-shot event extraction in the industrial maintenance sector requires a careful balance between performance and efficiency. Maintaining a balance between the two factors is of significant importance, as high performance

ensures accurate and reliable event extraction, essential for informed maintenance decisions. Efficiency is equally important in practical use, given the constraints on computational resources, real-time processing, and cost-effectiveness present in industrial environments. Although a complex model is usually highly accurate, it is impractical in industrial environments due to the high computational requirements, slower processing times, and higher costs. Consequently, the optimal model must prioritize efficiency while providing high accuracy to meet the application's resource-constrained and time-critical industrial requirements. Figure 13 presents the single event accuracy and the single event F1 scores. Model 4 produces the best results in German, while Model 1 produces the best results in English. In contrast, Model 2 delivers the worst results in both languages.

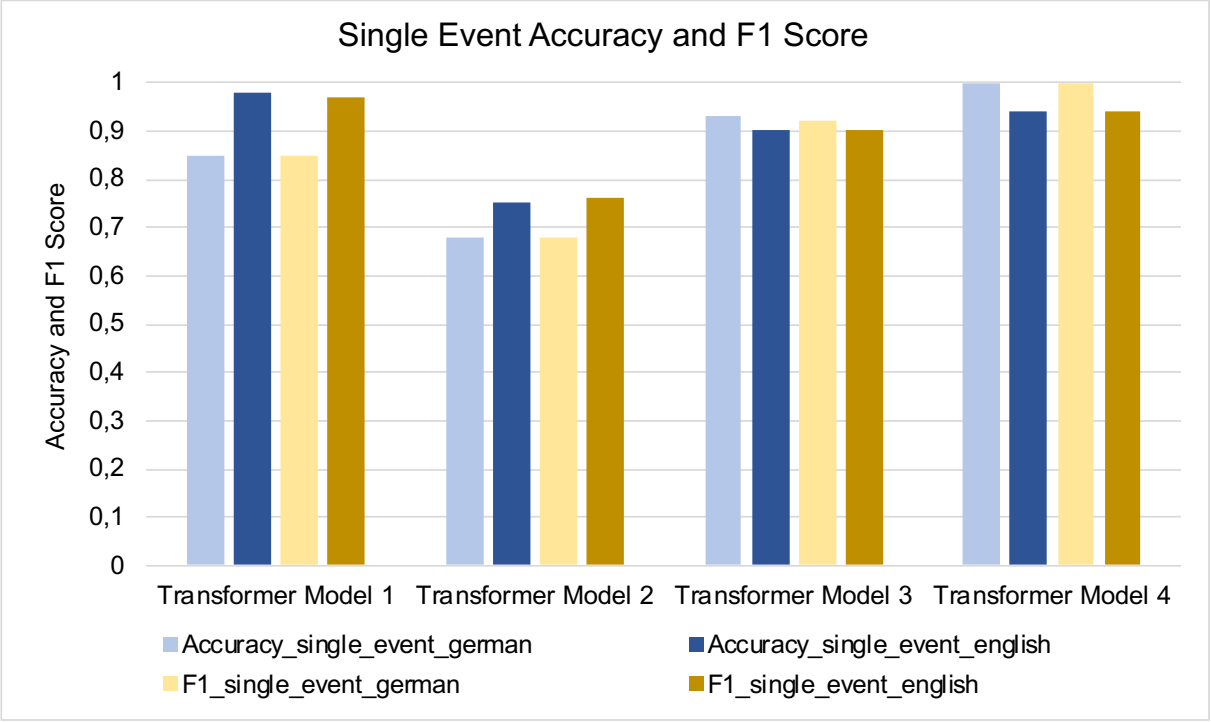



**Figure 13** Single Event Accuracy and F1 Score

Figure 14 presents the multiple event accuracy and the multiple event F1 scores. All models yield comparable outcomes in the extraction of multiple English events. Model 2 demonstrates superior performance in the extraction of multiple German events, which is crucial for the practical application of this thesis.

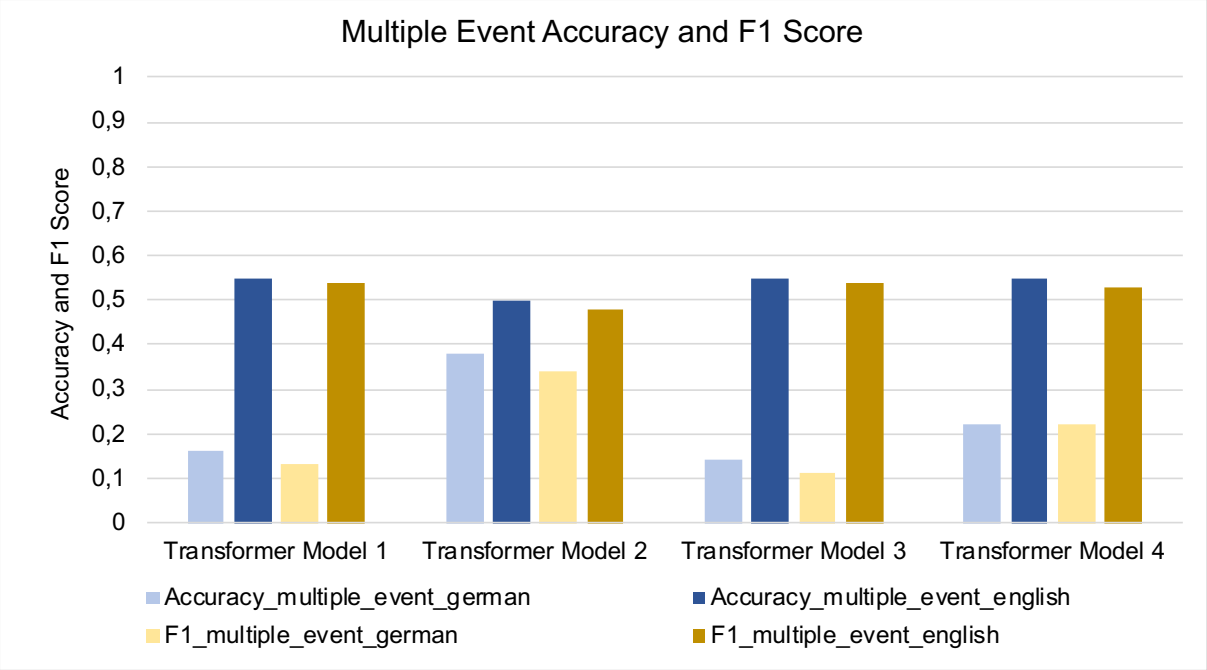
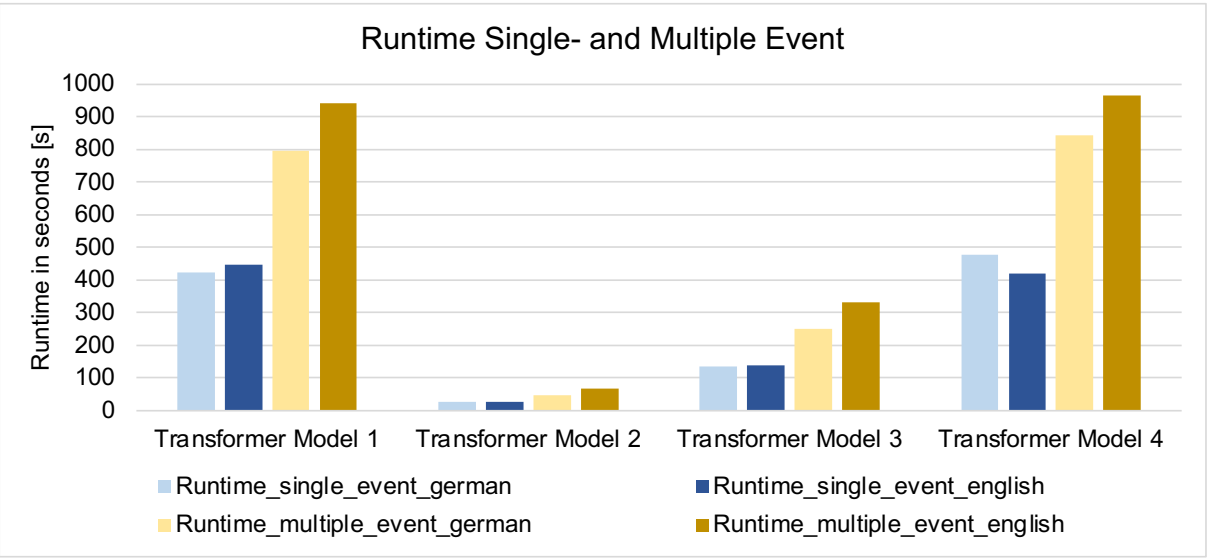



**Figure 14** Multiple Event Accuracy and F1 Score

Figure 15 presents the runtime for single and multiple event extraction. This comparison demonstrates that the runtimes of Model 1 and Model 4 range from 400 to almost 1000 seconds, respectively. Model 2 is the only model to have runtimes of less than 100 seconds for all tasks and for both German and English event extraction, which is a significant advantage in the practical application of industrial maintenance.

**Figure 15** Runtime Single- and Multiple Event




It is important to highlight the clear superiority of Model 2 in multiple event extraction in German, as shown in Figure 14. The remaining models achieve results that are at least 33% worse than those of Model 2 in the multiple event extraction task in German.

Model accuracy can be significantly enhanced through further research and model fine-tuning in the context of industrial maintenance. Therefore, the consistent and stable performance of Model 2 across all categories, combined with its superior runtime efficiency in both German and English for single and multiple events, makes it the best-suited model for this thesis. Although it may not achieve the highest scores in all fields, its outstanding efficiency aligns best with the requirements of the targeted use case in the industrial maintenance sector in the German-speaking region.

TEEE is primarily intended to support decision-making in industrial maintenance processes. For an AI application to be used as an instrument in real decision-making processes, the tool should be both accurate and transparent. This connection is crucial because the selected model's applicability in real industrial scenarios should be complemented by its transparency and comprehensibility. The focus on explainability in the following chapters is a natural progression from model selection to ensure that the chosen model is effective and meets the increasing demand for clarity and confidence in AI applications.

## 4.3 Explainability of Transformer Models

In recent years, the application of AI has increased significantly in many sectors. The increase in the number of AI publications is shown in Figure 16. The current results achieved with AI applications are promising and show a high potential for further improvements. In almost all sectors, it is crucial for AI models to not only produce significant results but also to be explainable to gain the trust of all stakeholders. This critical aspect of AI development is called explainable artificial intelligence (XAI). XAI aims to break down the black-box nature of AI models to increase stakeholders' confidence in the model's predictions. Trustworthiness is essential for using AI models in critical areas such as medicine and management. Current research has focused on the application of AI, but now the issue of XAI must be addressed, especially in the area of transformer models (Barredo Arrieta et al., 2020).

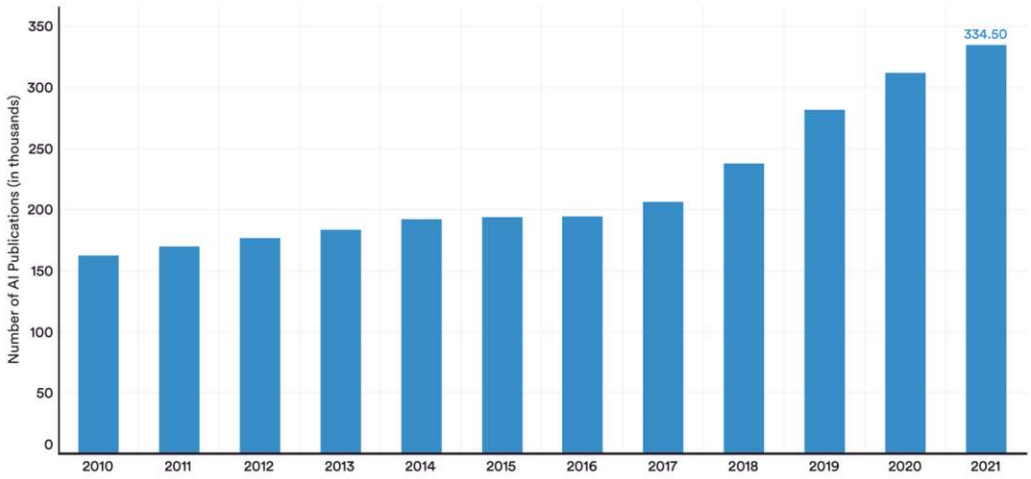



**Figure 16** Number of AI Publications in the World 2010-2021 (The AI Index Report 2022 – Artificial Intelligence Index, 2022)

In Section 4.3.1, the terminology of explainability and interpretability is discussed as they are often incorrectly used as synonyms. Subsequently, to establish a taxonomy for the explainability of transformer models, the basic dimensions of transformer explainability (local/global, ante-hoc/post-hoc, model-agnostic/model-specific) are introduced in Section 4.3.2. This taxonomy is established in Section 4.3.3, based on which the most appropriate explainability method for NLP transformer models is selected. This is based on the method proposed by Liu et al. (2021), which specifically addresses the explainability of transformer models in text classification. This work focuses on the explainability of transformer models, a sub-area of deep neural networks.

Transformer models rely on the self-attention mechanism, through this mechanism, every word in a sentence can be related to every other word, increasing the contextual understanding of the model. As a result, transformer models can capture contextual relationships better than conventional ML models. However, due to this self-attention mechanism, the decision-making process is far more complex than with conventional ML models (Vaswani et al., 2017). In addition, most transformer models are based on large pre-trained models such as BERT or GPT models, which have been trained on large amounts of text data. Pre-training and subsequent fine-tuning use transfer learning to adapt these models to specific downstream tasks (Howard & Ruder, 2018).

For this reason, the following sections focus on the relevant methodologies for deep neural networks. Section 4.3.1 defines key XAI model explainability terms, forming the basis for a taxonomy of transformer model explainability for event extraction in Section

4.3.3. Using this taxonomy, the most effective method to elucidate the predictions of this thesis's artefact, the TEEE, is identified.

### 4.3.1 XAI Terminology

Responsible Artificial Intelligence (RAI) is introduced before defining XAI terminology, i.e., a sub-area of RAI. Dignum (2017) introduced the concept of RAI. He described the characteristics of accountability, responsibility and transparency as the "ART" of AI. Benjamins et al. (2019) list the attributes of explainability, fairness, human-centricity, and privacy & security as the main characteristics of an RAI. Besinger et al. (2023) added the dimensions of Green AI and Accountable AI to optimize the RAI concept for the manufacturing sector. According to Besinger et al. (2023), the manufacturing domain of RAI encompasses the following dimensions: Accountable AI (AAI), Explainable AI (XAI), Fair AI (FAI), Human-Centric AI (HCAI), Green AI (GAI), and Privacy & Security. Figure 17 illustrates the interaction of the individual RAI domains. For this thesis, the XAI domain is primarily relevant.

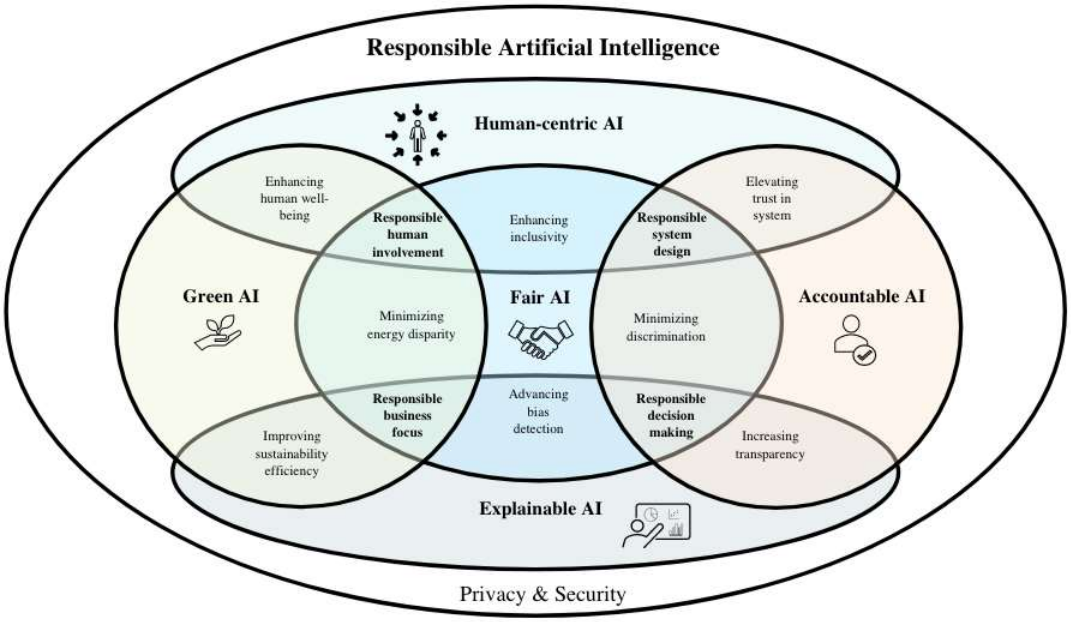



**Figure 17** Responsible AI for the Domain of Manufacturing (Besinger et al., 2023)

In order to be consistent throughout this thesis, in the following sections important XAI terminology, i.e., interpretability, explainability, black-box and white-box models are introduced.

The definition of **interpretability and explainability** varies in the literature, and there is a need for more terminological clarity. However, the concepts of interpretability and explainability are fundamentally different, and this misuse of terminology prevents the harmonization of terminology in the field of XAI (Barredo Arrieta et al., 2020). Numerous definitions of interpretability and explainability have been proposed in the existing literature. According to Gilpin et al. (2019), interpretability is defined as the goal of elucidating the inner workings of systems in a way that is understandable to humans. This definition is close to those of the authors (Doshi-Velez and Kim, 2017, p.2): *"Interpretability is the ability to explain or to present in understandable terms to a human"*. Many other definitions of interpretability exist in the scientific literature. The core message of most definitions is that interpretability is the ability of a system to be understood by humans.

In contrast, for Barredo Arrieta et al. (2020), explainability can be seen as an active characteristic of a model. Explainability involves any action or procedure that explains the output of a model's prediction. According to the definition proposed by Montavon et al. (2018), explainability is a compilation of features within the interpretable range that contribute to the decision-making process for a specific example. In the survey by the authors (Namatēvs et al., 2022, p.308 ), they define the terminology of explainability as follows: *"Explainability means the ability by which a human can justify the cause of the explanatory rule of the deep learning (DL) model's results."* Once the terminology between interpretability and explainability has been established, it should be noted that all the following sections and the practical part of this work refer to explainability, especially regarding transformer models.

Before defining the terms white-box and black-box models, it is necessary to clarify what model **transparency** means. In broad terms, a model is considered transparent if it is understandable. **White-box models** are always transparent and therefore explainable by design. Such models require no further explanation to be considered explainable. In recent years, DNNs have become increasingly important due to their empirical success in various fields of AI, especially in NLP (Barredo Arrieta et al., 2020). Transformer models belong to the category of DNNs. DNNs have many layers and parameters, and their internal workings are not easily understood; therefore, DNNs are considered black-box models.

**Black-box models** are the opposite of white-box models, as shown in Figure 18. In black-box models, the input and output are known, but everything in between is a black box, which is the opposite of transparent (Castelvecchi, 2016). This work focuses on the explainability of transformer models, which are black-box models. For this reason, the following explainability methodologies refer to the explainability of black-box models.

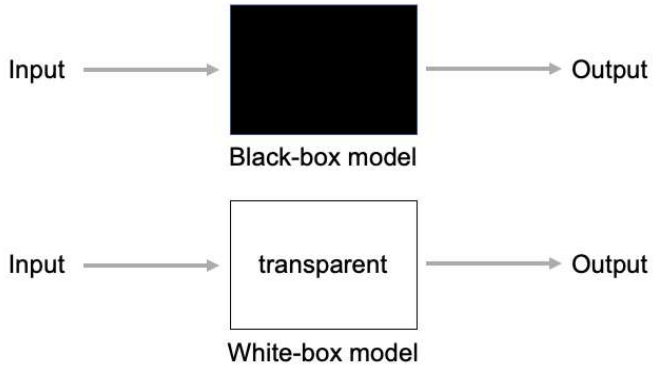



**Figure 18** Black-box Model and White-box Model

### 4.3.2 Dimensions of Transformer Model Explainability

In the subsequent sections, the key dimensions of transformer model explainability are introduced to establish a taxonomy for the explainability of transformer models in Section 4.3.3.

The terms **local and global explainability** are often used in the XAI literature. This distinction is referred to as the scope of explainability in this thesis. There are many definitions of local and global explainability in the literature, but there is general agreement on the fundamental meaning of these terms. Local explainability focuses on the explainability of each prediction outcome. The goal of local explainability is to evaluate the contribution of each input feature to the produced output; for example, in text classification, each word in the input text is evaluated for its contribution to the final classification. Local explainability is concerned with resolving the question: What is the reason for a particular decision made by the model (Namatēvs et al., 2022)?

In contrast, global explainability focuses on the explainability of the model itself rather than its predictions. Global explainability describes the behaviour of the model. The logic of the model can be followed from the input to output (Arya et al., 2019). This

thesis aims to break down the black-box nature of the transformer model's predictions to increase the confidence of the stakeholders. Therefore, the explainability of individual predictions is essential, which is why the practical part of this thesis focuses on local explainability.

According to Namatēvs et al. (2022), explainability methods can be categorized into **model-specific and model-agnostic**, based on their methodological view of explanation. Model-specific explainability methods explain a particular model based on its unique architecture and decision-making processes (Adadi & Berrada, 2018). Model-specific explainability methods analyze the inner workings of the network (Du et al., 2019). Model-specific methods can be applied to a particular class of models (Namatēvs et al., 2022).

Model-agnostic methods are used to explain black-box models. Model-agnostic methods are not limited in their application to a particular class of models (Namatēvs et al., 2022). They require access to the predictive function of the model to explain it. For the explanation of DL models, especially transformer models, model-agnostic methods are used post-hoc (Zini & Awad, 2022). Post-hoc terminology is specified in the following.

Ante-hoc originates from Latin and means "before". **Ante-hoc explainability methods** are built into the architecture of a model and are incorporated into the development process. Ante-hoc explanations illustrate the rationale behind the decision-making process between model input and output. Ante-hoc explainability is generally considered an intrinsic approach to model explainability. In the scientific literature, the terms self-explaining or directly interpretable methods are often used for ante-hoc explanations (Namatēvs et al., 2022).

In contrast to ante-hoc explanations, **post-hoc methods** require further steps after the model prediction, such as a second model to provide the explanations (Danilevsky et al., 2020). Post-hoc explanations address the decision-making process of a trained model that is not explainable by design and the relationships between each input feature and the prediction results (Turbé et al., 2022). Generally, post-hoc approaches offer more flexibility in their application to different models, but they provide less explanation about the DL model. Post-hoc methods do not explain the internal

workings of the model. However, they explain how the output is generated, such as by identifying the relevant input features for the final output (Namatēvs et al., 2022).

**Perturbation-based** methods evaluate the importance of input features for the final output by systematically perturbing the input feature and evaluating the changes in the output (S. Liu et al., 2021). Perturbations can include masking, removing, or changing input features. Perturbations can be achieved by permuting the feature values or replacing the feature values with a random sample from a uniform distribution. Based on these perturbations, the different input features are evaluated for their contribution to the output (Namatēvs et al., 2022). According to Namatēvs et al. (2022), perturbation-based methods rely on the manipulation (altering, removing or deleting) of input features or intermediate layers (activations).

**Gradient-based** explainability approaches calculate the rate of change of the output with respect to input changes (Atanasova et al., 2020). In gradient-based methods, the gradients are used to measure the change in the prediction within a local environment around the original point when certain input features are changed. In the scientific literature, many variants of gradient-based methods are mentioned. In this work, the term gradient-based methods is used broadly to refer to the current state-of-the-art gradient-based methods that sum up the gradient value at multiple points (S. Liu et al., 2021).

According to Liu et al. (2021), **propagation-based** methods are defined as methods that propagate layer by layer from the model output to the input features. This method of propagation is known as backward propagation. In the scientific literature, various propagation rules, such as forward propagation, are detailed (S. Liu et al., 2021). A further definition of propagation-based explainability methods is given by (Namatēvs et al., 2022, p.319): *"Explainability of the DL model can be explained by considering the deep network as a function (each neuron or group of neurons) by using gradient and backpropagation axioms of the function of interest to define the explanatory rule."*

Before clarifying the definition of **attention-based methods**, the term attention map needs to be defined. According to Sen et al. (2020), an attention map can be defined as a vector where each vector value is associated with a word positioned at the corresponding location within the text being analyzed. The vector values indicate the relevance of the corresponding word for a classification task. This is also called the

level of attention (Sen et al., 2020). Attention-based methods focus on understanding the logic behind DNNs with self-attention maps (S. Liu et al., 2021). Attention-based explainability methods are used in sequence-based tasks and work with a conditional distribution over a given input sequence of variable size. A weighted combination of all encoded input vectors is then calculated. These weights reflect the relevance of the input features to the output. The higher the weights, the greater the relevance of an input feature. If this calculation is performed for only one input sequence of the model, it is called self-attention (Namatēvs et al., 2022).

The following Table 9 shows a comparison matrix of perturbation-, gradient-, propagation-, and attention-based explainability methods.

**Table 9** Feature Matrix of Perturbation-, Gradient-, Propagation-, and Attention-Based Explainability Methods

| Feature | Perturbation-based | Gradient-based | Propagation-based | Attention-based |
|---|---|---|---|---|
| Principle | Systematically alters the input feature and evaluates the changes in the output. | Compares gradients of output with respect to input. | Propagate layer by layer from the model output to the input features. | Use attention weights as importance measures. |
| Explainability scope | Local | Local and global | Local and global | Local |
| Methodological explanatory | Model-agnostic | Model-specific | Model-specific | Model-specific |
| Examples | SHAP, LIME | Grad-CAM (Gradient-weighted Class Activation Mapping) | LRP (Layerwise Relevance Propagation) | Self-Attention Mechanism |
| Strengths | No model modification needed (works with any model). Intuitive to understand. | Precise and detailed explanations. | Robust against model changes if well implemented. | Easy to visualize and interpret. |
| Weaknesses | Computationally intensive (resource-heavy). | Requires gradient information. | Depends on network architecture. Understanding of model internals necessary. | Only useful for attention-based models. Attention weights not always correlate with importance. |

### 4.3.3 Transformer Explainability Taxonomy

Several basic approaches are currently being adopted in research on the classification of XAI. Barredo Arrieta et al. (2020) have provided a taxonomy that distinguishes between local and global explanations and also between model-agnostic and model-specific explanations. This taxonomy, or similar ones, is very common in the literature.

In their paper, Adadi & Berrada (2018) distinguish between the same two explainability approaches as Barredo Arrieta et al. (2020) in the context of model explainability. The taxonomy of Arya et al. (2019) focuses on the following questions:

- What is explained (data or model)?

- How is it explained (ante-hoc or post-hoc)?

- At what scale is it explained (local or global)?

In their paper, Liu et al. (2021) present a taxonomy for the explainability of transformer models, particularly for NLP applications. The taxonomy proposed by Liu et al. (2021) is used as the basic framework, providing a comprehensive basis for analysis. Additionally, this work extends the taxonomy by including relevant additional dimensions that contribute to a more comprehensive classification of the explainability of transformer models.

According to Liu et al. (2021), explanation methods for DNNs can be divided into perturbation-based, gradient-based, propagation-based, and attention-based methods. The terminology of these methods has already been introduced in Section 4.3.2. To increase the applicability of the taxonomy framework introduced by Liu et al. (2021) to a wide range of transformer applications, several additional dimensions are incorporated into the taxonomy introduced in this thesis. These additional dimensions expand the scope of the taxonomy and enable a more comprehensive and multipurpose classification system for the explainability methods of transformer models. The taxonomies proposed by Barredo Arrieta et al. (2020) and Arya et al. (2019) serve as valuable references in developing the taxonomy framework in this work. An overview of the taxonomy introduced in this thesis can be found in Table 10.

**Table 10** Taxonomy for the Explainability of Transformer Models

| **Explainability Taxonomy** | **Characteristics** | | | |
|---|---|---|---|---|
| Explainability scope | Global explainability | Local explainability | | |
| Methodological explanatory | Model-specific | Model-agnostic | | |
| Explainability stage | Ante-hoc explainability | Post-hoc explainability | | |
| Explainability methods for NLP | Perturbation-based methods | Gradient-based methods | Propagation based methods | Attention based methods |

The taxonomy introduced in this thesis includes the dimensions of explainability methods for NLP, stage of explainability, methodological explanatory, and scope of explainability. In the practical part of this thesis, the evaluation of explainability focuses on several transformer models that have been fine-tuned, especially for event extraction tasks. Given the need to assess explainability across multiple models, a local, model-agnostic, post-hoc explainability method is employed. Considering these aspects, two specific explainability methods, namely SHAP introduced by Lundberg & Lee (2017) and LIME (Local Interpretable Model-agnostic Explanations) introduced by Ribeiro et al. (2016), stand out prominently within the scientific literature. Therefore, SHAP and LIME are introduced in this section.

**LIME (Local Interpretable Model-agnostic Explanations)** is an explainability method that explains the predictions of various models. It achieves this by creating a local approximation of the model's prediction using an interpretable model. When considering a complex model at a local level, it can be observed that any complex model is linear at this level. This insight allows an excellent local approximation using an interpretable model (Ribeiro et al., 2016). In the context of LIME, the original instance is first altered to obtain a new set of samples. These perturbed samples are then used to make predictions with the complex model $f$. Subsequently, the samples are assigned weights based on their proximity to the original instance $x$. The weighting function is given by Equation 3 (Ribeiro et al., 2016):

**Equation 3** LIME weighting function (Ribeiro et al., 2016)

$$\pi_x(z) = \exp\left(-\frac{D(x,z)^2}{\sigma^2}\right)$$

In this equation, $D(x,z)$ represents the distance between $x$ and the perturbed sample $z$ with the width $\sigma$. The next step is to train a simple model $g$ based on the aforementioned weights to approximate $f$ by $x$. This is achieved by minimizing Equation 4 (Ribeiro et al., 2016).

**Equation 4** LIME function (Ribeiro et al., 2016)

$$\zeta(x) = \underset{g \in G}{\operatorname{argmin}} \sum_{z,z' \in Z} \pi_x(z)(f(z) - g(z'))^2$$

In Equation 4, $G$ represents the class of interpretable models, $Z$ represents the dataset of perturbed samples with the associated labels, and $z'$ is an interpretable instance from $z$ around the interpretable instance of $x$ which is $x'$. LIME is a computationally intensive method that requires extensive computing resources to generate explanations. Furthermore, finding a suitable approximation function can be challenging (Molnar, 2021). Due to these practical limitations, LIME is not used in the practical part of this thesis.

**SHAP (SHapley Additive exPlanations)** is a local, model-agnostic, post-hoc, and perturbation-based explainability framework for interpreting model predictions based on game theory. By utilizing SHAP, the importance of each feature to the model's predictions can be evaluated. The model's prediction can be seen as a game in the context of game theory (Lundberg & Lee, 2017). Each input feature acts as a player in the game (Shapley & Roth, 1988). The SHAP values represent the importance of each feature to the model's prediction (Lundberg & Lee, 2017).

SHAP is a perturbation-based explainability method that operates by perturbing the input features to understand the relationship between the model prediction and the input features. By systematically changing the values of individual features while keeping the remaining features constant, SHAP quantifies the importance of each feature to the model's prediction (S. Liu et al., 2021). In their paper, Lundberg & Lee (2017) introduced different variants of SHAP, each with specific advantages and disadvantages depending on the use cases. SHAP values are unitless. They quantify how much a particular feature influences the prediction relative to the baseline. The

baseline is the average prediction of the model over the entire dataset. The classic Shapley values are calculated according to the following Equation 5. It is necessary to retrain all feature subsets $S \subseteq F$ to the model. $F$ represents the set of all features. SHAP assigns a value to each feature which reflects its importance for the model prediction. To calculate the influence on the model prediction, a model $f_{S\cup\{i\}}$ is trained with the feature in question and a model $f_S$ without this feature. The two models' predictions for the current input are then compared: $f_{S\cup\{i\}}(x_{S\cup\{i\}}) - f_S(x_S)$. In this context, $x_S$ represents the values of the input feature in the text $S$. The impact of omitting one feature influences other features. Consequently, the differences in these features are calculated for all possible combinations of characteristics $S \subseteq F \setminus \{i\}$ **(**Lundberg & Lee**,** 2017). The formula for calculating the Shapley values is shown in Equation 5.

**Equation 5** Calculation of Shapley values (Lundberg & Lee, 2017)

$$\phi_i = \sum_{S\subseteq F\setminus\{i\}} \frac{|S|!\,(|F| - |S| - 1)!}{|F|!}\left[f_{S\cup\{i\}}(x_{S\cup\{i\}}) - f_S(x_S)\right]$$

The subsequent section introduces the TEEE artefact, which is implemented to facilitate the explainability of zero-shot event extraction using SHAP values. This tool utilizes the model selected in Section 4.2 for the downstream task of zero-shot event extraction. The TEEE employs SHAP to analyze the contribution of each word within a sentence to the model's predictions, thus elucidating the model's decision-making process in identifying events without prior specific training on those events.

## 4.4 Implementation of an Explainability Framework for Zero-Shot Event Extraction

Before introducing the explainability framework for zero-shot event extraction in this chapter, the overall architecture of TEEE is briefly reviewed. TEEE is a tool that processes maintenance texts from a machine logbook and a list of possible events to extract from those logbooks. First, the text passes through TEEE's Event Extraction Pipeline, which calculates the probability of each defined event. A dynamic threshold based on the mean and standard deviation of the event scores is then used to detect the events. Shapley values are then calculated for each token and event to provide

detailed explanations of the model predictions. For multi-sentence texts, this process is repeated for each sentence. TEEE produces two output graphs: one showing the average event extraction probability and the number of extracted events, and another showing the SHAP values for each token and event detected. Figure 3 in Section 1.3 illustrates the overall architecture of TEEE using a simple example.

Before developing TEEE, it was imperative to identify an appropriate transformer model for zero-shot event extraction and to determine a framework capable of elucidating model decisions. Consequently, these two critical tasks were defined as objectives of this thesis. In Section 4.2, different transformer models underwent rigorous testing to select one that excelled in accuracy and pragmatically aligned with the demands of industrial maintenance. Furthermore, as detailed in Section 4.3, a thorough investigation was conducted to find a fitting explainability framework (SHAP), which was subsequently incorporated into TEEE. This integration aims to render the model's predictions transparent and comprehensible to end-users, thereby creating trust and smoothing the pathway for adopting transformer models in essential maintenance tasks.

The SHAP values play a crucial role in enhancing the explainability of the TEEE, especially in the context of industrial maintenance. As described in Chapter 4.3.3, SHAP assigns each feature (word) from the input text an importance score, which indicates how much each feature contributes to the model's prediction (event extraction). For example, in the context of industrial maintenance, TEEE could extract "vibration" as an event from a machine logbook. Using the SHAP values, the user can then understand that, for example, "shock" and "increased" are the words from the logbook with the highest contribution to the "vibration" model decision. Therefore, TEEE can use SHAP to highlight which specific terms or phrases led to the prediction of a particular event, which can help maintenance personnel understand the reasoning behind each prediction, increasing the confidence and usability of TEEE. By using SHAP, the TEEE decision-making process becomes more transparent. Maintenance teams can see consistent explanations for predictions, which helps verify the reliability of TEEE. For example, if the term "overheating" consistently appears with high SHAP values in the machine logbook when predicting the event "engine failure," it indicates that TEEE accurately identifies relevant events. The correlation between high SHAP values and known problems such as overheating allows the maintenance personnel to

better understand and trust the model's predictions. This shows that the model is able to identify words that are relevant to a model decision, in this case, event extraction. This enables the maintenance personnel to understand which words have the greatest impact on the model decision, significantly increasing the transparency of the model decision.

### 4.4.1 The "Transformer Event Extraction Explainer" (TEEE)

The application "Transformer Event Extraction Explainer", shortened to TEEE, emerges as the artefact of this master thesis. TEEE provides an intuitive graphical user interface (GUI) designed for human interpretation of transformer model-based event extraction predictions. After initiating TEEE, users can upload text and event files through the graphical user interface. Once loaded, users may analyze individual sentences in the text or display a comprehensive overview of the predicted events.

TEEE is designed to predict and extract significant maintenance events from industrial maintenance texts. These industrial maintenance texts often contain valuable information about machine performance, faults, and necessary repairs. The primary aim is to identify and extract the events specified by the user with an explanation of the predictions. Various industrial maintenance tasks can be performed using this data (extracted events) from the TEEE. In a practical scenario, a use case for TEEE is fault detection. For this application, various potential faults must be defined and submitted to TEEE as events. TEEE then processes industrial maintenance texts to identify and extract these predefined faults. For this practical use case, TEEE must first be given the potential faults (events), such as "vibrations" or "overheating", and the maintenance logbook text to analyze. TEEE then analyzes the text for mentions of these faults and extracts the relevant events that indicate potential faults. After TEEE has extracted the relevant events, the user can use the Shapley values to analyze how the prediction was generated, i.e., the contribution of each word in the sentence to the extraction of the event. The extracted events can then be synchronized with historical and real-time machine data in a suitable data analytics tool to validate the extracted events and predict future faults. This correlation makes it possible to combine the data generated by TEEE to predict future failures and recommend preventive maintenance actions. Figure 19 below shows the TEEE user interface with a short user guide.

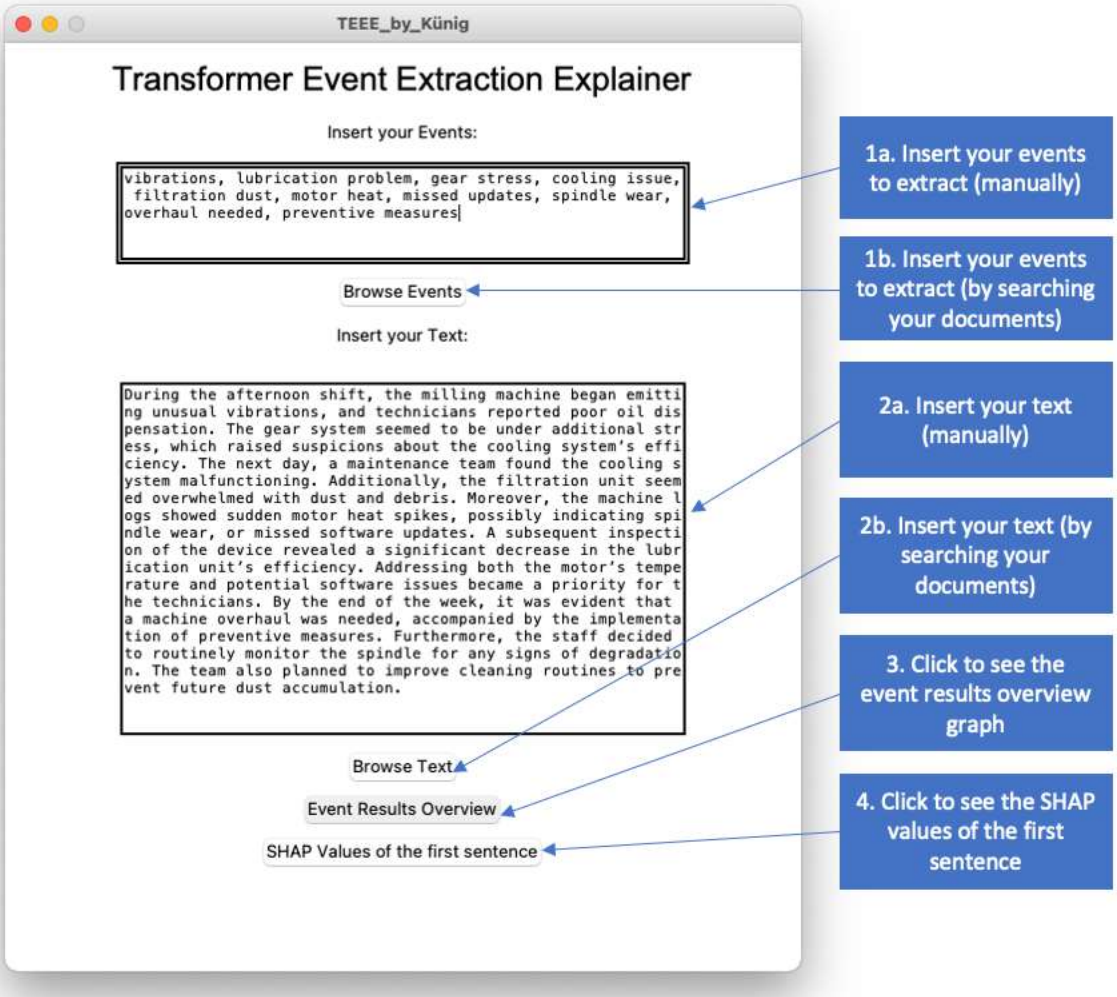



**Figure 19** TEEE User Interface

TEEE provides a visual representation of the average event extraction probability, the number of events extracted from the given text, and in-depth SHAP explanations for each detected event within a sentence. A bar chart displays the SHAP values of every word within the sentence. These explanations are crucial to make the predictions of the transformer model transparent and understandable for humans.

Events can be entered manually in the “Insert Events” text box (1a) or loaded directly using the “Browse Events” button (1b). Text can also be entered directly in the text box (2a) or by using the “Browse Text” button (2b) on the interface. The events must be entered in comma-separated value (CSV) format (e.g., leak, oil loss). In the same way, the text in the “Insert text” box can be written manually or loaded directly. The text must be in standard text file format (TXT), and sentences must be separated by a full stop (e.g., The machine is losing oil. The bearing is making a noise.).

Once the text and events have been entered into the appropriate text boxes, users can click the “Event Results Overview” button (3), shown in Figure 20, to display a graph showing the average probability and number of events extracted from the text.

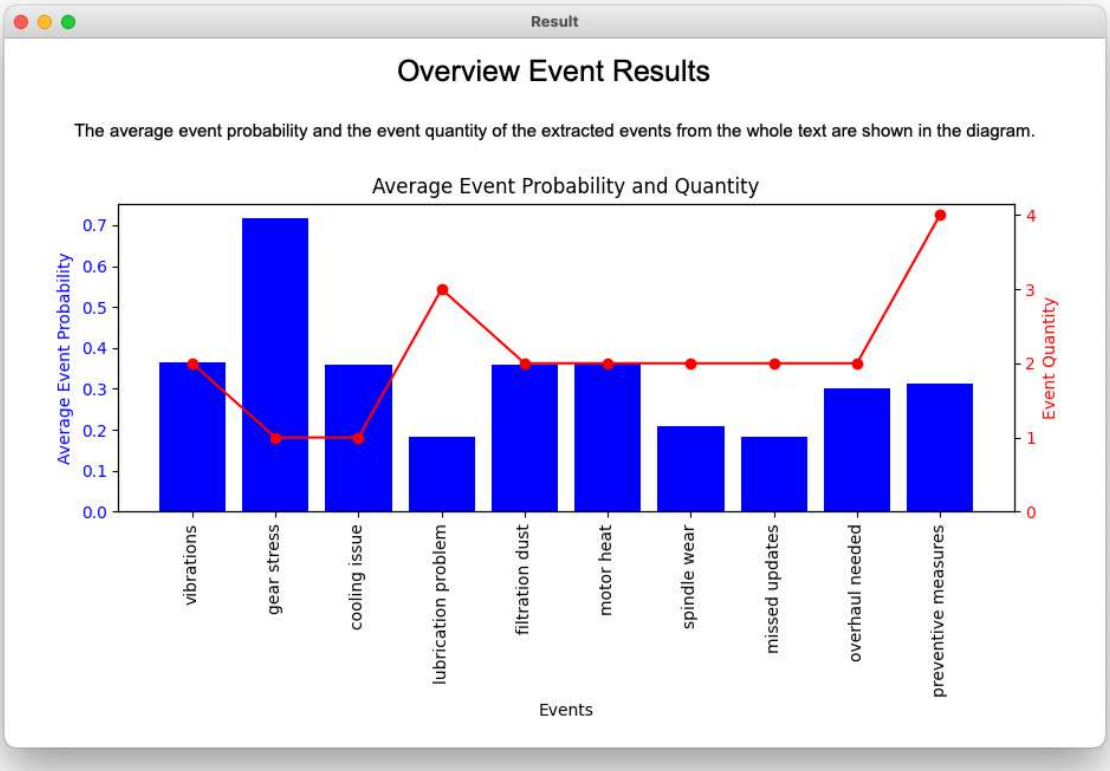



**Figure 20** Overview Event Results Graph

The user can also press the “SHAP Values of the first sentence” button (4), shown in Figure 21, to get a list of detected events from the first sentence. For each event from the first sentence detected by the model, a graph shows the SHAP value of each word from the sentence for the corresponding event. Pressing the button again displays the same information for the second sentence so that each sentence of the text can be viewed in detail.

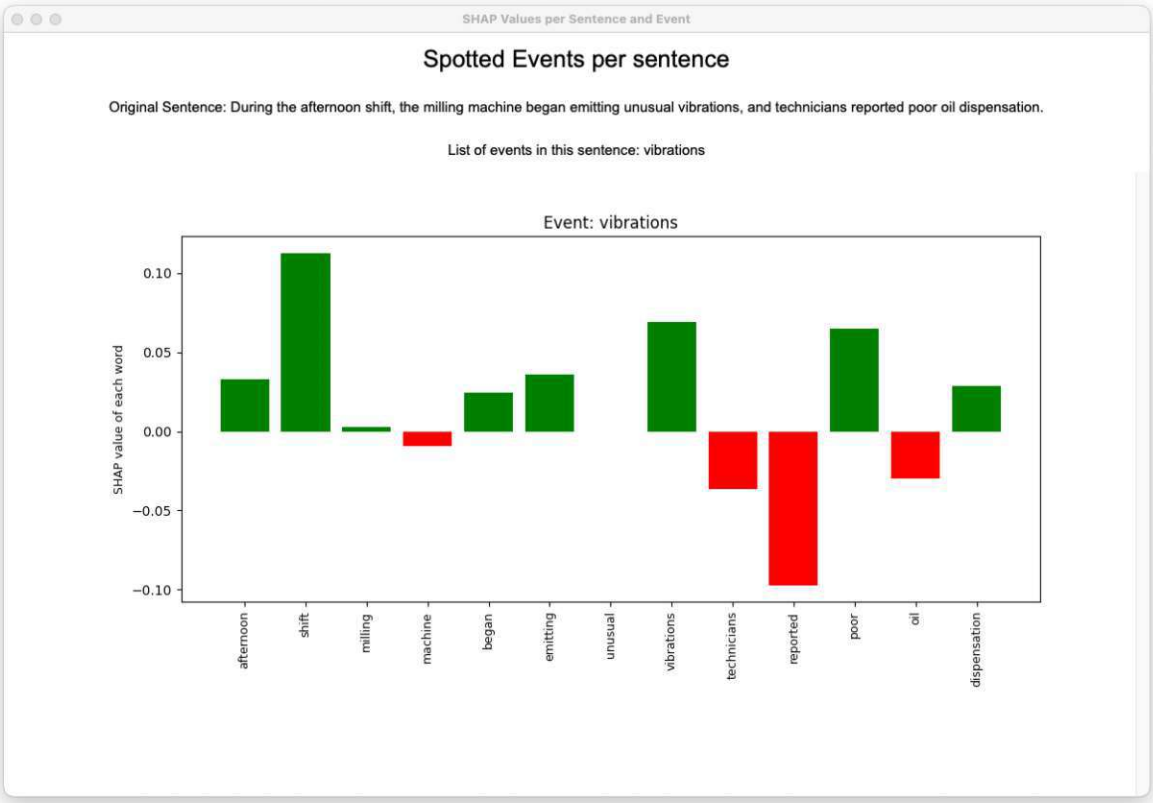



**Figure 21** SHAP Values for the first Sentence

### 4.4.2 Key components of TEEE

**Event Extraction Pipeline:** The pre-processing of input data, in the case of this thesis, namely the text and list of events, is a crucial task in NLP. In the summer of 2023 (i.e., time of conducting this research), Hugging Face did not provide an explicit pipeline for zero-shot event extraction in its Hugging Face Hub. Therefore, a specialized pipeline needed to be designed for the subsequent process of zero-shot event extraction. The developed event extraction pipeline is based on the “ZeroShotClassificationPipeline” from Hugging Face and was adapted to the needs of this thesis. The key modification to the “ZeroShotClassificationPipeline” is its flexible classification labeling adjustment. This empowers the pipeline to handle customized user events, respectively, labels. Conversely, the “ZeroShotClassificationPipeline” strictly deals with fixed, predefined labels. After adjusting the flexible label classification, the input is tokenized, the loaded Hugging Face model is applied to the input, and finally, post-processing is performed. Moreover, the __call__ method of the “ZeroShotClassificationPipeline” has been redefined to guarantee that the pipeline produces the anticipated labels and their

associated scores in a structured format that suits the demands of the event extraction procedure. The event extraction pipeline architecture is shown in Figure 22 and the event extraction pipeline code is shown in Figure 29 in the Appendix.

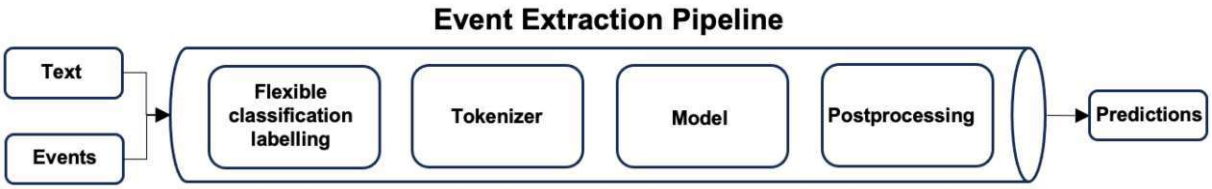



**Figure 22** Event Extraction Pipeline Architecture

**Event processing:** For each sentence, the model anticipates possible events and their respective probabilities. The TEEE uses a dynamic threshold to extract only relevant events. The algorithm determines the probabilities of all potential events, isolating only those with a probability surpassing the threshold. Rather than employing a static, predetermined threshold, the threshold is calculated using the mean and standard deviation of all event probabilities. This adaptable approach ensures that the threshold adjusts to the variability and central tendency of the values, allowing for a more precise extraction of significant events. The event extraction pipeline code is shown in Figure 29 of the Appendix.

**Explainability Plot with Shapley Values:** The explainability plot with Shapley values is generated for each sentence and each extracted event within the sentence. Stopwords, such as articles, are excluded from TEEE as the exclusion of stopwords significantly increases the performance of NLP models, such as TEEE (Sarica & Luo, 2021). The following Table 11 presents an example input for the TEEE.

**Table 11** Fictional Example Sentence with Evaluation Events

| | |
|---|---|
| Example sentence | The main conveyor belt had misaligned, causing a disruption in the production line. |
| Events to extract | misalignment, calibration issue, malfunction |

Figure 23 below shows the explainability plot showing the Shapley values for the example sentence (stopwords excluded) containing the identified misalignment event.

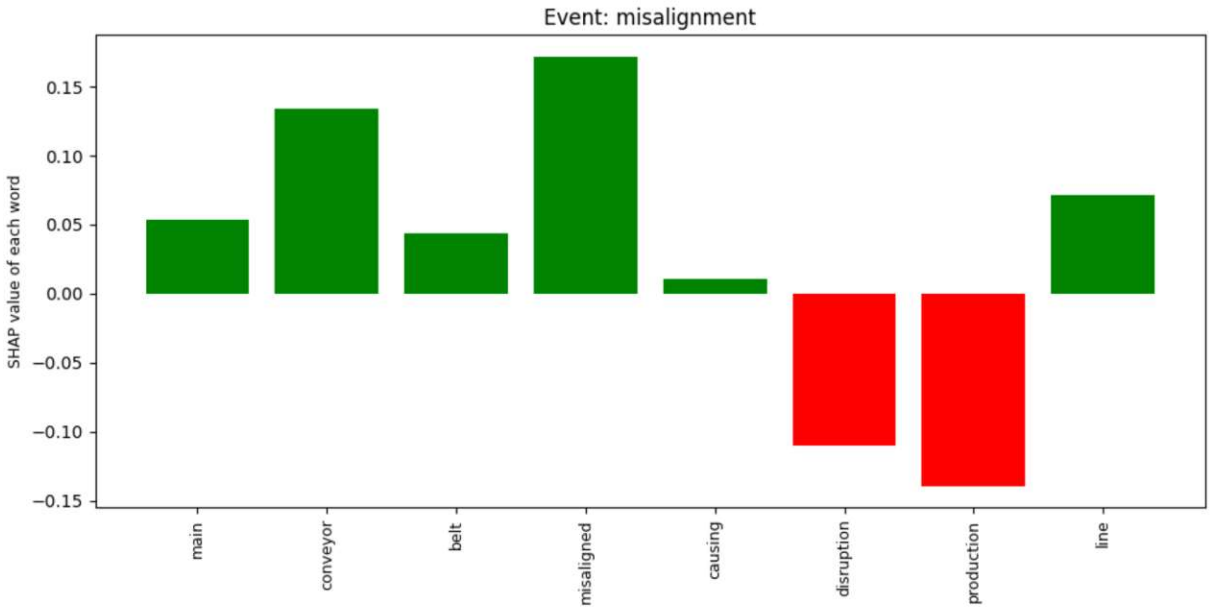



**Figure 23** Explainability Plot with Shapley Values

The horizontal axis shows the example sentence words separately, while the vertical axis shows the SHAP values assigned to each. SHAP values can be positive, negative, or zero. The use of different colors, mainly green for positive SHAP values and red for negative, provides an intuitive way to quickly identify which words the model considers to be driving forces for or against the occurrence of a particular event. A positive SHAP value indicates that a word increases the likelihood of extracting the event, while a negative value decreases the probability of the event occurring. In this context, SHAP values explain how each word in a sentence affects the model's prediction. SHAP values are unitless. They quantify how much a particular feature influences the prediction relative to the baseline. The baseline is the average prediction of the model over the entire dataset. For example, a SHAP value of 0.16 for the word "misaligned" means that the presence of this word increases the prediction probability for the event "misalignment" by 0.16 units from the baseline. These units are on the same scale of the probability value of the model prediction (event value), which in the case of TEEE is between 0 and 1.

**Overall Result Plot:** The overall result plot displays all events extracted from the text with their corresponding extraction probability and quantity. Figure 24 below shows an example of an overall result plot.

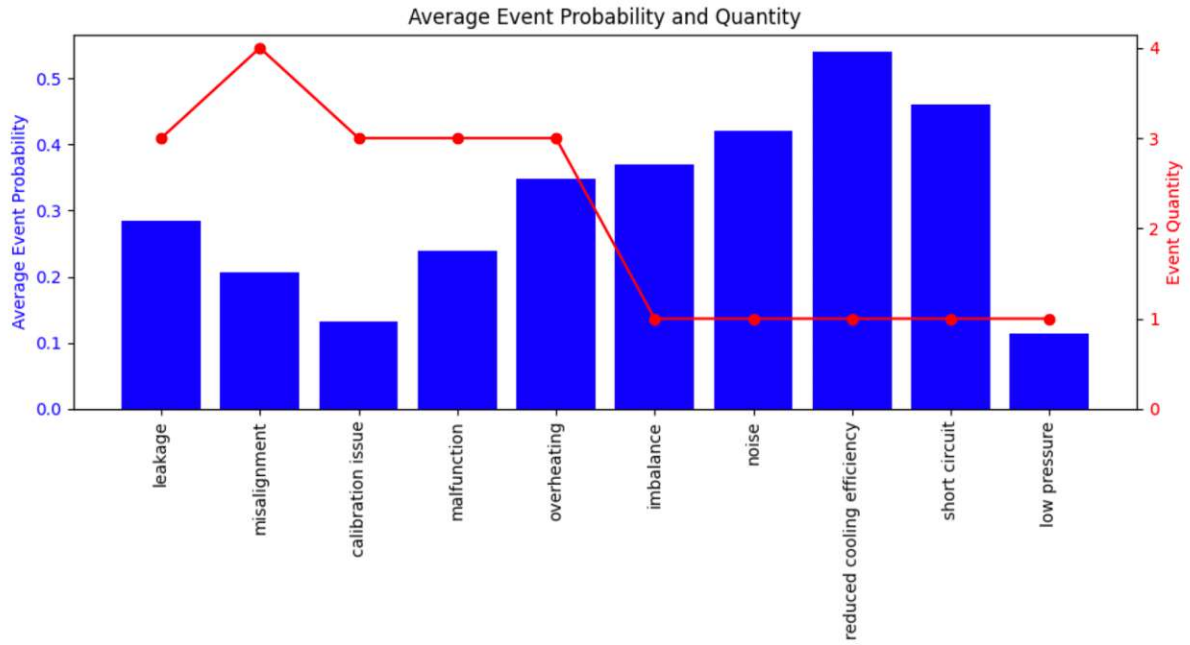



**Figure 24** Overall Result Plot

This diagram compares two primary metrics: the average event probability (event score) and the event quantity. The horizontal axis shows the detected events. The left vertical axis, shown in blue, plots the average probability for each event based on the model's predictions. This indicates how likely the model believes a given event will occur based on the text. In parallel, the right vertical axis, shown in red, lists the number (quantity) of identified events in the text. This shows how often a particular event was detected in the analyzed text segment. The red line with markers illustrates this count. The combination of these two metrics allows us to understand both the model's reliability in detecting a particular event and the relevance or frequency of that event in the given text. For example, an event frequently detected with high probability could be considered a critical or dominant theme in the analyzed text segment.

# 5 Evaluation of the Transformer Event Extraction Explainer (TEEE)

After identifying and evaluating all necessary components for developing the TEEE in Chapter 3, Chapter 4 evaluates the TEEE based on the previous section describing the implementation and key elements. This section focuses on a comprehensive evaluation of the TEEE, aiming to measure its performance and highlight any limitations. The evaluation provides both quantitative metrics and qualitative insights. Evaluation datasets are introduced in Section 5.1 and are used to measure the runtimes of the application and provide a quantitative analysis in Section 5.2 of TEEE efficiency. The same datasets are used as the basis for a qualitative assessment in Section 5.3, analyzing the accuracy and relevance of the TEEE output. This two-pronged approach provides a holistic understanding of the TEEE's capabilities and potential areas for improvement.

## 5.1 Evaluation Methodology and Dataset

The TEEE is evaluated in this section using fictional English and German industrial maintenance datasets; see Table 12 and Table 13. In the evaluation with the German dataset, the context from the English dataset is intentionally used but written simply to assess how the model handles poorly formulated texts. However, it closely resembles scenarios that can occur within industrial maintenance settings, providing a credible context for assessing the TEEE. This ensures that the evaluation is based on a context that reflects possible real-world applications, even though the data is not from real-world applications. The evaluation datasets consist of ten sentences with ten potential events. The evaluation text is first divided into separate tokens and then fed into the "ZeroShotClassificationPipeline", introduced in Chapter 4.4.2. These tokens represent the individual words in the text, excluding any punctuation marks. This division helps TEEE capture the text's structural and semantic aspects. In the subsequent analysis, each token of a sentence is plotted on the horizontal axis of the explainability plot. This enables users to view the SHAP value of each token, through which they can determine the contribution of each token to the model's decision. The datasets were designed to cover the diversity of real industrial maintenance scenarios.

The model runtime is analyzed quantitatively, and the application's performance is analyzed qualitatively. Table 12 shows the English dataset.

**Table 12** English Evaluation Text and Events (English Dataset)

| | |
|---|---|
| Evaluation Text | During the afternoon shift, the milling machine began emitting unusual vibrations, and technicians reported poor oil dispensation. The gear system seemed to be under additional stress, which raised suspicions about the cooling system's efficiency. The next day, a maintenance team found the cooling system malfunctioning. Additionally, the filtration unit seemed overwhelmed with dust and debris. Moreover, the machine logs showed sudden motor heat spikes, possibly indicating spindle wear, or missed software updates. A subsequent inspection of the device revealed a significant decrease in the lubrication unit's efficiency. Addressing both the motor's temperature and potential software issues became a priority for the technicians. By the end of the week, it was evident that a machine overhaul was needed, accompanied by the implementation of preventive measures. Furthermore, the staff decided to routinely monitor the spindle for any signs of degradation. The team also planned to improve cleaning routines to prevent future dust accumulation. |
| Evaluation Events | vibrations, lubrication problem, gear stress, cooling issue, filtration dust, motor heat, missed updates, spindle wear, overhaul needed, preventive measures |

Table 13 shows the German dataset. The text and events are adapted from the English dataset but written in an industrial maintenance style.

**Table 13** German Evaluation Text and Events (German Dataset)

| | |
|---|---|
| Evaluation Text | Fräsmaschine vibrierte stark, schlechte Ölabgabe. Getriebe unter Stress, Verdacht auf Kühlsystem. Kühlsystem defekt, Filtereinheit voll mit Staub. Motor-Temperaturspitzen, möglicher Spindelverschleiß oder verpasste Software-Updates. Schmierungseinheit ineffizient. Motortemperatur und Softwareprobleme beheben. Maschinenüberholung nötig, präventive Maßnahmen eingeführt. Spindel regelmäßig überwachen, Reinigungsroutinen verbessern, Staubansammlung verhindern. Pumpe blockiert, Kühlprobleme. Filter verstopft, Schmierprobleme. |
| Evaluation Events | Vibrationen, Schmierprobleme, Getriebestress, Kühlungsprobleme, Filterstaub, Motorwärme, verpasste Updates, Spindelverschleiß, notwendige Überholung, Präventivmaßnahmen |

## 5.2 Quantitative Evaluation of TEEE Runtime

A crucial aspect of this thesis is to evaluate the computational efficiency of the TEEE. The runtime required to load the explainability plots with SHAP values was measured for each sentence. In addition, the time taken to load the overall results plot in its entirety was also measured. Table 14 and Table 15 below show the quantitative outcomes of the runtime for the English and German datasets.

**Table 14** Quantitative Evaluation of TEEE Runtime (English Dataset)

| Task to be calculated | Runtime |
|---|---|
| Explainability Plot Sentence 1 | 28 seconds |
| Explainability Plot Sentence 2 | 19 seconds |
| Explainability Plot Sentence 3 | 7 seconds |
| Explainability Plot Sentence 4 | 18 seconds |
| Explainability Plot Sentence 5 | 23 seconds |
| Explainability Plot Sentence 6 | 11 seconds |
| Explainability Plot Sentence 7 | 23 seconds |
| Explainability Plot Sentence 8 | 43 seconds |
| Explainability Plot Sentence 9 | 29 seconds |
| Explainability Plot Sentence 10 | 15 seconds |
| **Overall Result Plot** | **233 seconds** |

On average, it took around 21.6 seconds to load the explainability plots for each sentence of the English dataset. The total runtime for the explainability plots was 216 seconds for all sentences. The overall result plot required about 233 seconds, approximately 1.1 times the cumulative time of the individual explainability plots.

Table 15 below shows the quantitative outcomes of the German dataset.

**Table 15** Quantitative Evaluation of TEEE Runtime (German Dataset)

| Task to be calculated | Runtime |
|---|---|
| Explainability Plot Sentence 1 | 14 seconds |
| Explainability Plot Sentence 2 | 9 seconds |
| Explainability Plot Sentence 3 | 8 seconds |
| Explainability Plot Sentence 4 | 31 seconds |
| Explainability Plot Sentence 5 | 5 seconds |
| Explainability Plot Sentence 6 | 6 seconds |
| Explainability Plot Sentence 7 | 9 seconds |
| Explainability Plot Sentence 8 | 18 seconds |
| Explainability Plot Sentence 9 | 11 seconds |
| Explainability Plot Sentence 10 | 13 seconds |
| **Overall Result Plot** | **112 seconds** |

On average, it took around 12.4 seconds to load the explainability plots for each sentence of the German dataset. The total runtime for the explainability plots was 124 seconds for all sentences. The overall result plot required about 126 seconds. It is noticeable that the average runtime of the German simplified maintenance style dataset differs significantly from the English well-formulated dataset: the average runtime of the explainability plots for each sentence is 2.6 times faster for the simplified German dataset than for the English well-formulated dataset.

The computing power of the hardware on which the application runs is decisive in determining the runtimes. For this evaluation, application runtimes are measured on a 2018 Apple MacBook Pro with a 6-core Intel i7 processor clocked at 2.6GHz, 16GB of DDR4 memory, and a Radeon Pro 560X graphics card with 4GB of GDDR5 memory. The performance metrics suggest that the application performs satisfactorily on this hardware. However, if the scope of the evaluation is broadened or the application is used in industry, it may be necessary to use more powerful computing resources.

## 5.3 Qualitative Evaluation of TEEE Explanations

The qualitative evaluation examines the quality of the explanations provided by the TEEE. Since the evaluation of the transformer model's event extraction capability has already been conducted in Section 4.2, no further assessment is required for the quality of event extraction from the TEEE. Therefore, explainability plots with Shapley values from the evaluation dataset are examined more closely in this section. To evaluate the TEEE explanations, two graphs from the English dataset and two graphs from the German dataset are analyzed in depth. One sentence from each dataset, in which two events were extracted from the TEEE, is analyzed. The sentence from the English dataset for the evaluation is: *"Addressing both the motor's temperature and potential software issues became a priority for the technicians."* The sentence from the German dataset for the evaluation is: *"Kühlsystem defekt, Filtereinheit voll mit Staub."*

Within the given English sentence, the TEEE has identified the events of motor heat and preventive measures. The following Figure 25 and Figure 26 show the explainability plot with the SHAP values of the English sentence.

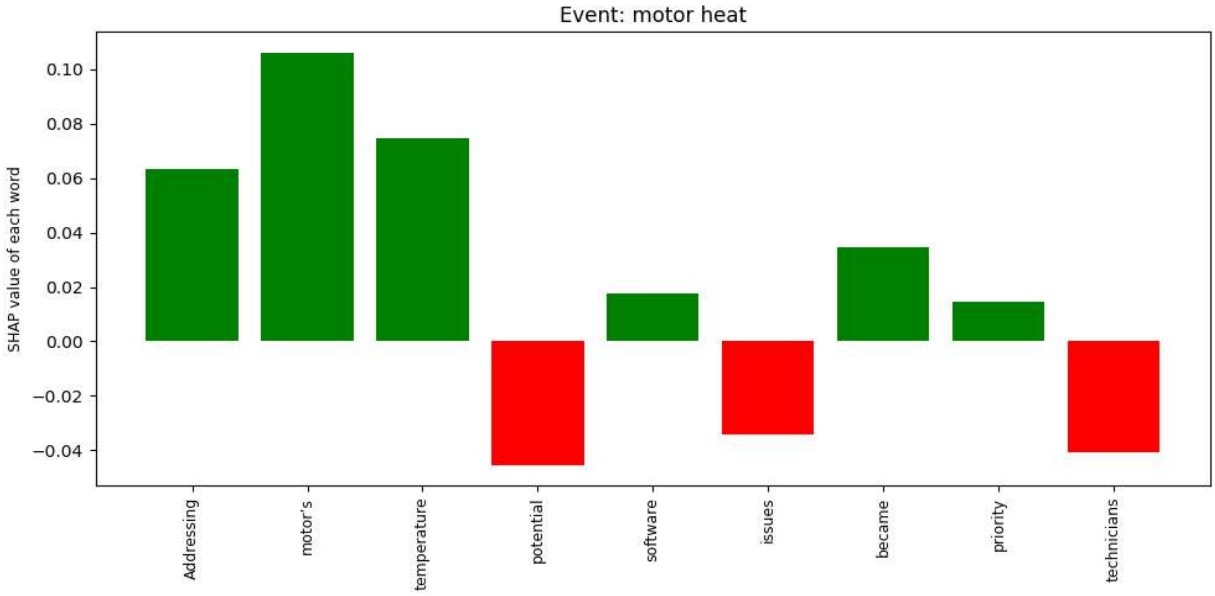



**Figure 25** Explainability Plot Motor Heat (English Dataset)

Figure 25 shows that the words „Addressing“, „motor’s“, and „temperature“ in particular have high SHAP values when predicting the event „motor heat“. The remaining words have a relatively small positive or negative influence on the prediction. High positive SHAP values for words directly related to the predicted event, in this case, motor and temperature, confirm the model’s focus on the relevant words and thus increase the explainability of the TEEE’s predictions.

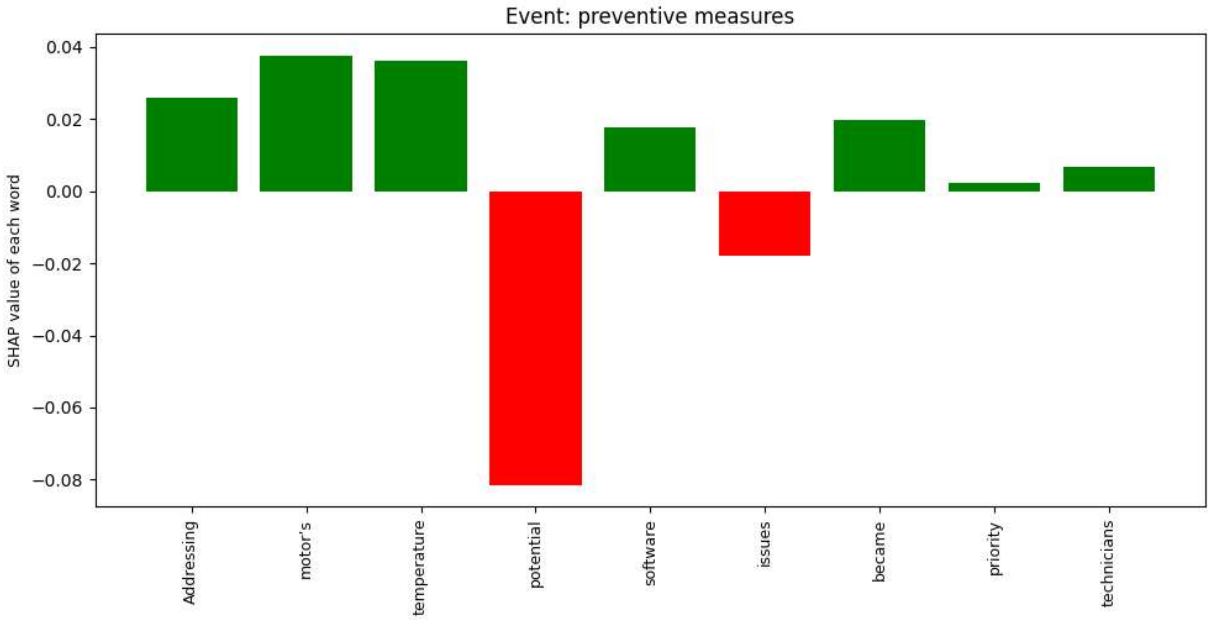



**Figure 26** Explainability Plot Preventive Measures (English Dataset)

The SHAP values shown in Figure 26 for “preventive measures” are not as meaningful as for the event “motor heat”, as a synonym of “preventive measures” does not appear directly in the sentence. The words „Addressing“, „motor’s“ and „temperature“ have a high SHAP value, but a much lower one compared to the event „motor heat“ in Figure 25. More strikingly, the word „potential“ has a high negative SHAP value, indicating that it reduces the likelihood of predicting „preventive measures“ from the sentence. This suggests that in the context of the sentence, „potential“ is less relevant or inversely related to the event of preventive measures.

Upon examination of the two figures, certain peculiarities can be noted. In examining the SHAP value distributions for these two events, it is noticeable that “Addressing”, “software, “issue”, “became” and “priority” have almost similar SHAP values for both events, indicating their consistently low contribution to the likelihood of the predicted events. High consistency of the SHAP values of specific tokens across different events indicate that the contributions of these tokens to the model decision are relatively stable. Based on this observation, it is possible to deduce in advance how certain tokens contribute to the model’s decision, which increases human confidence in the model decisions. Figure 25 clearly shows that the words “motor’s” and “temperature” have high SHAP values for predicting the event “motor heat”. These terms are closely related to the “motor heat” event, making their significant contribution to the model’s prediction intuitively understandable to humans. This explanation increases the transparency of the TEEE, thereby enhancing user trust in its predictions.

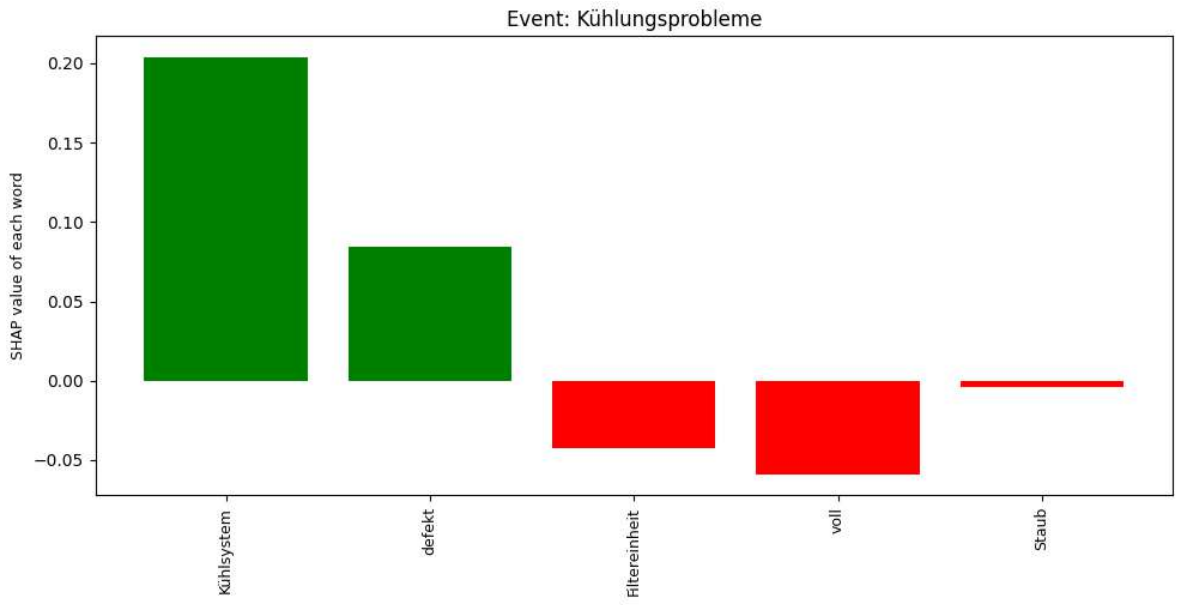



**Figure** 27 Explainability Plot Kühlungsprobleme (German Dataset)

Figure 27 shows the SHAP values from the “Kühlungsprobleme“ event. The German evaluation text was deliberately given to TEEE in a simple manner to assess how the model handles poorly formulated texts. It closely resembles scenarios that can occur within industrial maintenance settings. Figure 27 shows that the words “Kühlsystem“ and “defekt“ in particular have high SHAP values when predicting the event “Kühlungsprobleme“. The high positive SHAP values for words directly related to the predicted event, in this case, “Kühlsystem“ and “defekt“, confirm the model‘s focus on the relevant words and thus increase the explainability of the TEEE’s predictions also for poorly formulated texts that occur in the practical setting of industrial maintenance.

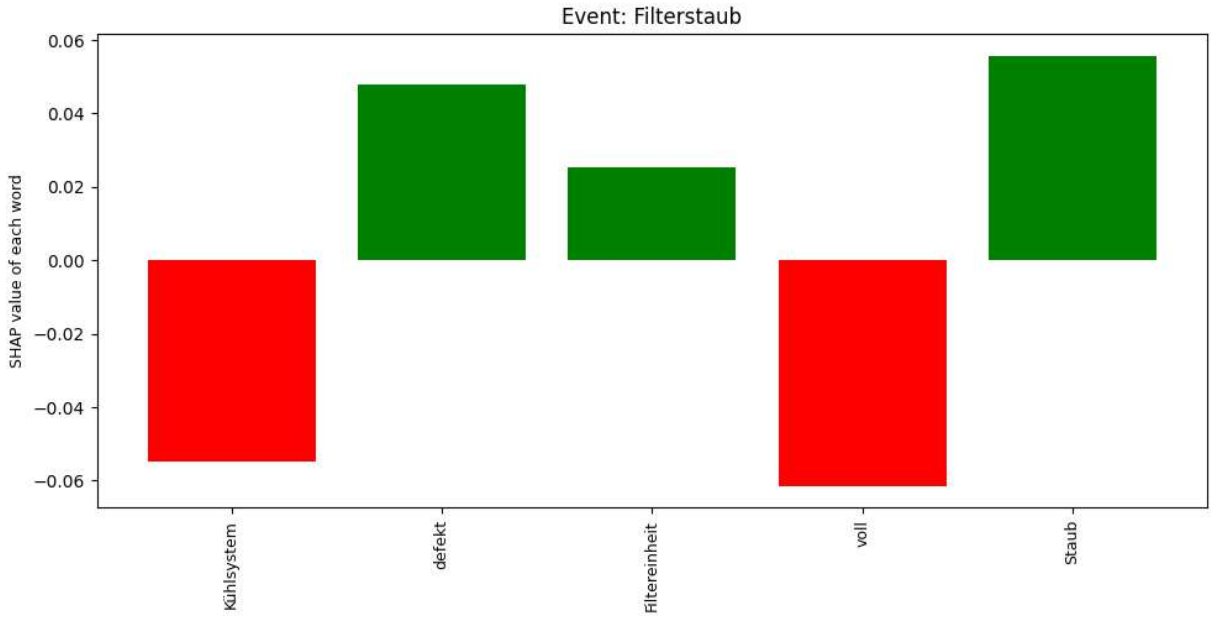



**Figure 28** Explainability Plot Filterstaub (German Dataset)

Figure 28 shows the SHAP values for the “Filterstaub” event. The word “Staub” has the highest positive SHAP value, suggesting it is strongly associated with the prediction of “Filterstaub”. As “Staub” is directly related to the event “Filterstaub”, its presence significantly increases the likelihood of this prediction.

From Figures 27 and 28, it is particularly noticeable that certain words show a high variation in their SHAP values for the two different events. The high variation in SHAP values for the words “Kühlsystem”, “Filtereinheit” and “Staub” in the two diagrams highlight their contextual impact. For example, the word “Filtereinheit” has a high SHAP value when TEEE predicts the event “Kühlungsprobleme“ and a low SHAP value when TEEE predicts the event “Filterstaub“. This variability underscores the importance of individual words for the model predictions. This variance in the SHAP values of

individual tokens can therefore help to increase the explainability of the model. By understanding how the meaning and influence of words change with context, it is possible to understand more easily why the model makes certain decisions.

In contrast, the words “defekt” and “voll” have relatively similar SHAP values when predicting the two different events. These two words cannot be directly associated with either of the two events and are therefore not of primary importance in predicting the events. It is easy to understand from the logical way of thinking of humans that words that are not directly related to the predicted events always make a similarly small contribution to the prediction. This increases the trust of humans in the model and thus in its decisions. Additionally, the analysis of the German dataset shows that TEEE can also be used for poorly formulated texts that occur in practical industrial maintenance environments.

The qualitative analysis of the results reveals important information about the model’s decision-making process. Depending on the context, the model assigns very different SHAP values to the same words, illustrating the model’s sensitivity to individual words. The evaluation of TEEE explanations indicates that they clarify the decision-making process, increasing the explainability of the predictions of transformer models for human interpreters. This leads to increased trust in using transformer models such as TEEE in industrial maintenance scenarios.

# 6 Conclusion and Outlook

In this thesis, an application known as TEEE was developed utilizing the design science research methodology. This work aims to create an application that initially extracts events from industrial maintenance texts using a transformer model and renders the outcomes of this downstream task comprehensible. Consequently, the sub-research questions, defined in Section 1.2, were defined as follows:

**RQ1:** *What explainability methodology can be used to explain transformer models predictions for zero-shot event extraction?*

In recent years, transformer models have been successfully deployed in various sectors. Numerous scientific studies have demonstrated the benefits and accuracy of transformer models. However, the practical application of transformer models poses a significant challenge: their predictions are not easily understandable for humans due to their black-box nature. Their predictions must be both accurate and understandable for transformer models to be used successfully in practical applications, such as industrial maintenance. People working with transformer models must trust their predictions to be effective. Due to the importance of the explainability of transformer models, a systematic literature review was conducted for this sub-research question. The systematic literature review in Section 4.3 initially analyzed the basics of XAI. Subsequently, a taxonomy for the explainability of transformer models was created based on the current state-of-the-art literature. With the help of this taxonomy, the most suitable explainability framework was selected for this thesis. SHAP, which stands for SHapley Additive exPlanations, was identified as this work's most suitable explainability framework. SHAP uses game theory to interpret model predictions and evaluates the importance of each feature towards the model's predictions.

**RQ2:** *What is the utilization and success rate of transformer models when applied to an industrial maintenance dataset for zero-sot event extraction tasks?*

Scientific studies in various areas have already confirmed the benefits and success of transformer models. In industrial maintenance, very little scientific literature was available when this thesis was written. For this reason, various fine-tuned transformer models were presented in Section 4.1 and evaluated using several fictional datasets from industrial maintenance in Section 4.2. Four models for the downstream task of

zero-shot event extraction were compared quantitatively and qualitatively. The models were compared quantitatively based on the accuracy, the F1 score and the model runtime. The qualitative analysis used to compare the models in this section focused on their practical applicability in real-world scenarios, specifically for zero-shot event extraction in industrial maintenance texts. The qualitative analysis involved a more nuanced evaluation of the models beyond raw performance metrics. The model "MoritzLaurer/multilingual-MiniLMv2-L6-mnli-xnli" proved to be the most suitable model for the use case of this thesis, primarily due to its consistent and stable performance across all categories, combined with its superior runtime efficiency in both German and English for single and multiple events.

**RQ3:** *What measurement methods can be used to evaluate the explainability of transformer models when applied to an industrial maintenance dataset for zero-shot event extraction tasks?*

In this work, TEEE was developed as a framework to enhance the explainability of transformer models in industrial maintenance. The methodology for explainability employed in TEEE is based on the literature research presented in Chapter 3, thus addressing the first research question. TEEE's capability for zero-shot event extraction is based on the transformer model selected in Section 4.2, responding to the second research question.

Evaluating the explainability of transformer models is challenging because it requires balancing the model's performance with the quality of explanation provided in terms humans can understand. Chapter 5 of this thesis addresses this issue. One way to assess the explainability of TEEE is to evaluate its ability to explain its predictions in a way understandable to humans. This was achieved through qualitative analysis using a fictional example sentence in the domain of industrial maintenance. Additionally, the model's runtime was evaluated quantitatively, revealing relevant patterns within the example sentence that were meaningful from a human perspective and, thus, understandable.

**Primary research question:** *"How to increase the explainability of transformer models predictions for zero-shot event extraction in industrial maintenance?"*

By answering research questions 1-3, a framework, the TEEE, was developed, which emerges as the artefact of this diploma thesis and simultaneously answers the primary research question in one sentence. The TEEE increases the explainability of transformer model predictions in industrial maintenance.

## 6.1 Limitations and Future Work

The TEEE produces promising results in explaining event extraction tasks within the industrial maintenance domain. However, TEEE, like any established application or model, has limitations, and future work will be necessary.

One advantage of TEEE is its applicability to different domains, as it is not restricted to industrial maintenance applications. The transformer model was not explicitly fine-tuned for an industrial maintenance dataset. While this broad applicability is a strength, it also represents a significant potential for future improvement. At the time of writing this thesis, no suitable dataset was available for event extraction in the industrial maintenance sector. For this reason, an already fine-tuned transformer model was used. If a suitable training dataset is available in the future, it can be used to fine-tune the current transformer and subsequently substitute it within the application. By fine-tuning the current model using a specific data set from the industrial maintenance sector, TEEE can be significantly improved for the industrial maintenance domain.

Evaluating the explainability of the TEEE is a challenging task that requires balancing the model's performance and the quality of its explanations regarding explainability to humans. The qualitative analysis within this study revealed patterns in the fictional sentence from industrial maintenance that were meaningful and understandable from a human viewpoint. However, these patterns are specific to the fictional sentence evaluated in this thesis. A substantial dataset would be required to uncover a broader range of relevant patterns in the transformer model underlying TEEE. This dataset must then be analyzed to identify and understand general patterns behind TEEE comprehensively.

Another limitation of TEEE is the significant delay in processing long texts. This latency, particularly in real-time applications where quick decisions are essential, could affect its practical applicability. Further optimization of TEEE in terms of processing time will be a fundamental part of future developments.

In conclusion, TEEE is already successfully contributing to the explainability of event extraction in industrial maintenance, but its potential has yet to be fully realized. The flexible architecture of TEEE allows it to be applied to a wide range of domains, but a fine-tuned transformer would optimize the application domain specifically. As the field of explainable AI, especially transformer models and event extraction continues to grow and more specialized datasets become available, there will be opportunities in the future to optimize TEEE for successful application in the industry. In the future, developing an even more efficient and specific TEEE will be possible, further closing the gap between transformer explainability and real-world applications.

# References

Adadi, A., & Berrada, M. (2018). Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access*, *6*, 52138–52160. https://doi.org/10.1109/ACCESS.2018.2870052

Akhbardeh, F., Desell, T., & Zampieri, M. (2020). NLP Tools for Predictive Maintenance Records in MaintNet. *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: System Demonstrations*, 26–32. https://aclanthology.org/2020.aacl-demo.5

Alammar, J. (2021). Ecco: An Open Source Library for the Explainability of Transformer Language Models. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, 249–257. https://doi.org/10.18653/v1/2021.acl-demo.30

Alawneh, L., Mohsen, B., Al-Zinati, M., Shatnawi, A., & Al-Ayyoub, M. (2020). A Comparison of Unidirectional and Bidirectional LSTM Networks for Human Activity Recognition. *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 1–6. https://doi.org/10.1109/PerComWorkshops48775.2020.9156264

Alpaydin, E. (1995). *Selective Attention for Handwritten Digit Recognition*.

Ansari, F., Glawar, R., & Nemeth, T. (2019). PriMa: A prescriptive maintenance model for cyber-physical production systems. *International Journal of Computer Integrated Manufacturing*, *32*(4–5), 482–503. https://doi.org/10.1080/0951192X.2019.1571236

Ansari, F., Kohl, L., Giner, J., & Meier, H. (2021). Text mining for AI enhanced failure detection and availability optimization in production systems. *CIRP Annals*, *70*(1), 373–376. https://doi.org/10.1016/j.cirp.2021.04.045

Arya, V., Bellamy, R. K. E., Chen, P.-Y., Dhurandhar, A., Hind, M., Hoffman, S. C., Houde, S., Liao, Q. V., Luss, R., Mojsilović, A., Mourad, S., Pedemonte, P., Raghavendra, R., Richards, J., Sattigeri, P., Shanmugam, K., Singh, M., Varshney, K. R., Wei, D., & Zhang, Y. (2019). *One Explanation Does Not Fit All: A Toolkit and Taxonomy of AI Explainability Techniques* (arXiv:1909.03012). arXiv. https://doi.org/10.48550/arXiv.1909.03012

Atanasova, P., Simonsen, J. G., Lioma, C., & Augenstein, I. (2020). *A Diagnostic Study of Explainability Techniques for Text Classification* (arXiv:2009.13295). arXiv. https://doi.org/10.48550/arXiv.2009.13295

Attanasio, G., Nozza, D., Pastor, E., & Hovy, D. (2022). Benchmarking Post-Hoc Interpretability Approaches for Transformer-based Misogyny Detection. *Proceedings of NLP Power! The First Workshop on Efficient Benchmarking in NLP*, 100–112. https://doi.org/10.18653/v1/2022.nlppower-1.11

Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., & Herrera, F. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, *58*, 82–115. https://doi.org/10.1016/j.inffus.2019.12.012

Benjamins, R., Barbado, A., & Sierra, D. (2019). *Responsible AI by Design in Practice* (arXiv:1909.12838). arXiv. http://arxiv.org/abs/1909.12838

Besinger, P., Vejnoska, D., & Ansari, F. (2023). Responsible AI (RAI) in Manufacturing: A Qualitative Framework. *Procedia Computer Science*.

Bousdekis, A., Lepenioti, K., Apostolou, D., & Mentzas, G. (2019). Decision Making in Predictive Maintenance: Literature Review and Research Agenda for Industry 4.0. *IFAC-PapersOnLine*, *52*(13), 607–612. https://doi.org/10.1016/j.ifacol.2019.11.226

Briceno-Mena, L. A., Arges, C. G., & Romagnoli, J. A. (2022). Machine Learning-Based Surrogate Models and Transfer Learning for Derivative Free Optimization of HTPEM Fuel Cells. In L. Montastruc & S. Negny (Eds.), *Computer Aided Chemical Engineering* (Vol. 51, pp. 1537–1542). Elsevier. https://doi.org/10.1016/B978-0-323-95879-0.50257-5

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020). *Language Models are Few-Shot Learners* (arXiv:2005.14165). arXiv. https://doi.org/10.48550/arXiv.2005.14165

Brundage, M. P., Sexton, T., Hodkiewicz, M., Dima, A., & Lukens, S. (2021). Technical language processing: Unlocking maintenance knowledge. *Manufacturing Letters*, *27*, 42–46. https://doi.org/10.1016/j.mfglet.2020.11.001

Castelvecchi, D. (2016). Can we open the black box of AI? *Nature News*, *538*(7623), 20. https://doi.org/10.1038/538020a

Cheong, L. L., Meharizghi, T., Black, W., Guang, Y., & Meng, W. (2022). *Explainability of Traditional and Deep Learning Models on Longitudinal Healthcare Records* (arXiv:2211.12002). arXiv. https://doi.org/10.48550/arXiv.2211.12002

Cohen, G., Sapiro, G., & Giryes, R. (2019). *DNN or k-NN: That is the Generalize vs. Memorize Question* (arXiv:1805.06822). arXiv. https://doi.org/10.48550/arXiv.1805.06822

Cole, R., Purao, S., Rossi, M., & Sein, M. (2005). *Being Proactive: Where Action Research Meets Design Research.*

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., & Stoyanov, V. (2020). Unsupervised Cross-lingual Representation Learning at Scale. *Proceedings of the 58th Annual*

*Meeting of the Association for Computational Linguistics*, 8440–8451. https://doi.org/10.18653/v1/2020.acl-main.747

Conneau, A., Lample, G., Rinott, R., Williams, A., Bowman, S. R., Schwenk, H., & Stoyanov, V. (2018). *XNLI: Evaluating Cross-lingual Sentence Representations* (arXiv:1809.05053). arXiv. https://doi.org/10.48550/arXiv.1809.05053

Danilevsky, M., Qian, K., Aharonov, R., Katsis, Y., Kawas, B., & Sen, P. (2020). A Survey of the State of Explainable AI for Natural Language Processing. *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, 447–459. https://aclanthology.org/2020.aacl-main.46

Dankar, A., Jassani, A., & Kumar, K. (2023). *Improving Knowledge Distillation for BERT Models: Loss Functions, Mapping Methods, and Weight Tuning* (arXiv:2308.13958). arXiv. https://doi.org/10.48550/arXiv.2308.13958

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* (arXiv:1810.04805). arXiv. https://doi.org/10.48550/arXiv.1810.04805

Di Flumeri, F. (2022, August). *Explainable AI for supporting operators in manufacturing machines maintenance: Evaluating different techniques of explainable AI for a machine learning model that can be used in a manufacturing environment* [Info:eu-repo/semantics/masterThesis]. University of Twente. http://essay.utwente.nl/93244/

Dignum, V. (2017). *RESPONSIBLE ARTIFICIAL INTELLIGENCE: DESIGNING AI FOR HUMAN VALUES*. *1*.

Ding, Y., Jia, M., Miao, Q., & Cao, Y. (2022). A novel time-frequency Transformer based on self-attention mechanism and its application in fault diagnosis of rolling bearings. *Mechanical Systems and Signal Processing*, *168*, 108616. https://doi.org/10.1016/j.ymssp.2021.108616

Do, P., Voisin, A., Levrat, E., & Iung, B. (2015). A proactive condition-based maintenance strategy with both perfect and imperfect maintenance actions. *Reliability Engineering & System Safety*, *133*, 22–32. https://doi.org/10.1016/j.ress.2014.08.011

Dong, J., Guan, Z., Wu, L., Zhang, Z., & Du, X. (2022). *Towards Explainability in NLP: Analyzing and Calculating Word Saliency through Word Properties* (arXiv:2207.08083). arXiv. https://doi.org/10.48550/arXiv.2207.08083

Doshi-Velez, F., & Kim, B. (2017). *Towards A Rigorous Science of Interpretable Machine Learning* (arXiv:1702.08608). arXiv. https://doi.org/10.48550/arXiv.1702.08608

Du, M., Liu, N., & Hu, X. (2019). *Techniques for Interpretable Machine Learning* (arXiv:1808.00033). arXiv. https://doi.org/10.48550/arXiv.1808.00033

Edwards, B., Zatorsky, M., & Nayak, R. (2008). *Clustering and Classification of Maintenance Logs using Text Data Mining*.

Eyzaguirre, C., Rio, F. del, Araujo, V., & Soto, A. (2021). *DACT-BERT: Increasing the efficiency and interpretability of BERT by using adaptive computation time.* https://openreview.net/forum?id=wKfXaxPist

Gandomi, A., & Haider, M. (2015). Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*, *35*(2), 137–144. https://doi.org/10.1016/j.ijinfomgt.2014.10.007

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., & Kagal, L. (2019). *Explaining Explanations: An Overview of Interpretability of Machine Learning* (arXiv:1806.00069). arXiv. http://arxiv.org/abs/1806.00069

Guo, H., Zhang, Y., & Zhu, K. (2022). Interpretable deep learning approach for tool wear monitoring in high-speed milling. *Computers in Industry*, *138*, 103638. https://doi.org/10.1016/j.compind.2022.103638

He, P., Gao, J., & Chen, W. (2023). *DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing* (arXiv:2111.09543). arXiv. http://arxiv.org/abs/2111.09543

Hevner, A. (2007). A Three Cycle View of Design Science Research. *Scandinavian Journal of Information Systems*, *19*.

Hevner, A., R, A., March, S., T, S., Park, Park, J., Ram, & Sudha. (2004). Design Science in Information Systems Research. *Management Information Systems Quarterly*, *28*, 75.

Hossin, M., & M.N, S. (2015). A Review on Evaluation Metrics for Data Classification Evaluations. *International Journal of Data Mining & Knowledge Management Process*, *5*, 01–11. https://doi.org/10.5121/ijdkp.2015.5201

Howard, J., & Ruder, S. (2018). *Universal Language Model Fine-tuning for Text Classification* (arXiv:1801.06146). arXiv. https://doi.org/10.48550/arXiv.1801.06146

Hsieh, C.-Y., Li, C.-L., Yeh, C.-K., Nakhost, H., Fujii, Y., Ratner, A., Krishna, R., Lee, C.-Y., & Pfister, T. (2023). *Distilling Step-by-Step! Outperforming Larger Language Models with Less Training Data and Smaller Model Sizes* (arXiv:2305.02301). arXiv. https://doi.org/10.48550/arXiv.2305.02301

IOT Analytics. (2021, May 4). Predictive Maintenance Market: The Evolution from Niche Topic to High ROI Application. *IoT Analytics*. https://iot-analytics.com/predictive-maintenance-market-evolution-from-niche-topic-to-high-roi-application/

Järvinen, P. (2007). Action Research is Similar to Design Science. *Quality & Quantity*, *41*(1), 37–54. https://doi.org/10.1007/s11135-005-5427-1

Jiao, Z., Pan, L., Fan, W., Xu, Z., & Chen, C. (2022). Partly interpretable transformer through binary arborescent filter for intelligent bearing fault diagnosis. *Measurement*, *203*, 111950.

https://doi.org/10.1016/j.measurement.2022.111950

Kagermann, H., Wahlster, W., & Helbig, J. (2013). Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0. Abschlussbericht des Arbeitskreises Industrie 4.0. *acatech*. https://www.acatech.de/publikation/umsetzungsempfehlungen-fuer-das-zukunftsprojekt-industrie-4-0-abschlussbericht-des-arbeitskreises-industrie-4-0/

Kim, Y., Denton, C., Hoang, L., & Rush, A. M. (2017). *Structured Attention Networks* (arXiv:1702.00887). arXiv. https://doi.org/10.48550/arXiv.1702.00887

Klaise, J., Van Looveren, A., Cox, C., Vacanti, G., & Coca, A. (2020). *Monitoring and explainability of models in production* (arXiv:2007.06299). arXiv. https://doi.org/10.48550/arXiv.2007.06299

Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text Classification Algorithms: A Survey. *Information*, *10*(4), Article 4. https://doi.org/10.3390/info10040150

Lample, G., & Conneau, A. (2019). *Cross-lingual Language Model Pretraining* (arXiv:1901.07291). arXiv. https://doi.org/10.48550/arXiv.1901.07291

Laurer, M., van Atteveldt, W., Casas, A., & Welbers, K. (2022). *BERT-NLI-transfer-learn-laurer.pdf*. https://osf.io/https://osf.io/74b8k

Lee, J., Ni, J., Djurdjanovic, D., Qiu, H., & Liao, H. (2006). Intelligent prognostics tools and e-maintenance. *Computers in Industry*, *57*(6), 476–489. https://doi.org/10.1016/j.compind.2006.02.014

Li, C., Chen, Y., & Shang, Y. (2022). A review of industrial big data for decision making in intelligent manufacturing. *Engineering Science and Technology, an International Journal*, *29*, 101021. https://doi.org/10.1016/j.jestch.2021.06.001

Li, X., Xiong, H., Li, X., Wu, X., Zhang, X., Liu, J., Bian, J., & Dou, D. (2022). Interpretable deep learning: Interpretation, interpretability, trustworthiness, and beyond. *Knowledge and Information Systems*, *64*(12), 3197–3234. https://doi.org/10.1007/s10115-022-01756-8

Liang, D., Gonen, H., Mao, Y., Hou, R., Goyal, N., Ghazvininejad, M., Zettlemoyer, L., & Khabsa, M. (2023). *XLM-V: Overcoming the Vocabulary Bottleneck in Multilingual Masked Language Models* (arXiv:2301.10472). arXiv. http://arxiv.org/abs/2301.10472

Lin, T., Wang, Y., Liu, X., & Qiu, X. (2022). A survey of transformers. *AI Open*, *3*, 111–132. https://doi.org/10.1016/j.aiopen.2022.10.001

Liu, H., Liu, Z., Jia, W., Lin, X., & Zhang, S. (2020). A novel transformer-based neural network model for tool wear estimation. *Measurement Science and Technology*, *31*(6), 065106. https://doi.org/10.1088/1361-6501/ab7282

Liu, J., Min, L., & Huang, X. (2021). *An overview of event extraction and its applications* (arXiv:2111.03212). arXiv. https://doi.org/10.48550/arXiv.2111.03212

Liu, S., Le, F., Chakraborty, S., & Abdelzaher, T. (2021). On Exploring Attention-based Explanation for Transformer Models in Text Classification. *2021 IEEE International Conference on Big Data (Big Data)*, 1193–1203. https://doi.org/10.1109/BigData52589.2021.9671639

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach* (arXiv:1907.11692; Version 1). arXiv. https://doi.org/10.48550/arXiv.1907.11692

Lopez, M. M., & Kalita, J. (2017). *Deep Learning applied to NLP* (arXiv:1703.03091). arXiv. https://doi.org/10.48550/arXiv.1703.03091

Lu, S., Zhao, G., Li, S., & Guo, J. (2022). Explainable document-level event extraction via back-tracing to sentence-level event clues. *Knowledge-Based Systems*, *248*, 108715. https://doi.org/10.1016/j.knosys.2022.108715

Lundberg, S., & Lee, S.-I. (2017). *A Unified Approach to Interpreting Model Predictions*.

Madsen, A., Reddy, S., & Chandar, S. (2023). Post-hoc Interpretability for Neural NLP: A Survey. *ACM Computing Surveys*, *55*(8), 1–42. https://doi.org/10.1145/3546577

Mahlamäki, K., Niemi, A., Jokinen, J., & Borgman, J. (2016). Importance of maintenance data quality in extended warranty simulation. *International Journal of COMADEM*, *19*, 3–10.

Marques, H., & Giacotto, A. (2019). *Prescriptive Maintenance: Building Alternative Plans for Smart Operations*. 231–236. https://doi.org/10.3384/ecp19162027

Martindale, N., & Stewart, S. (2021). TX$^2$: Transformer eXplainability and eXploration. *Journal of Open Source Software*, *6*(68), 3652. https://doi.org/10.21105/joss.03652

Matyas, K. (2022). *Instandhaltungslogistik: Qualität und Produktivität steigern*.

May, M. C., Neidhöfer, J., Körner, T., Schäfer, L., & Lanza, G. (2022). Applying Natural Language Processing in Manufacturing. *Procedia CIRP*, *115*, 184–189. https://doi.org/10.1016/j.procir.2022.10.071

Molnar, C. (2021). *Interpretable Machine Learning*.

Montavon, G., Samek, W., & Müller, K.-R. (2018). Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, *73*, 1–15. https://doi.org/10.1016/j.dsp.2017.10.011

Mylonas, N., Mollas, I., & Tsoumakas, G. (2022). *An Attention Matrix for Every Decision: Faithfulness-based Arbitration Among Multiple Attention-Based Interpretations of Transformers in Text Classification* (arXiv:2209.10876). arXiv. https://doi.org/10.48550/arXiv.2209.10876

Namatēvs, I., Sudars, K., & Dobrājs, A. (2022). Interpretability versus Explainability:

Classification for Understanding Deep Learning Systems and Models. *Computer Assisted Methods in Engineering and Science*, *29*(4), Article 4. https://doi.org/10.24423/cames.518

Naylor, M., French, C., Terker, S., & Kamath, U. (2021). *Quantifying Explainability in NLP and Analyzing Algorithms for Performance-Explainability Tradeoff* (arXiv:2107.05693). arXiv. https://doi.org/10.48550/arXiv.2107.05693

Nemeth, T., Bernerstätter, R., Glawar, R., Matyas, K., & Sihn, W. (2015). Instandhaltung 4.0: Sicherstellung von Produktqualität und Anlagenverfügbarkeit durch einen echtzeitbasierten Instandhaltungsleitstand. *Zeitschrift für wirtschaftlichen Fabrikbetrieb*, *110*(9), 569–573. https://doi.org/10.3139/104.111373

Ouyang, X., Wang, S., Pang, C., Sun, Y., Tian, H., Wu, H., & Wang, H. (2021). *ERNIE-M: Enhanced Multilingual Representation by Aligning Cross-lingual Semantics with Monolingual Corpora* (arXiv:2012.15674). arXiv. http://arxiv.org/abs/2012.15674

Peres, R. S., Jia, X., Lee, J., Sun, K., Colombo, A. W., & Barata, J. (2020). Industrial Artificial Intelligence in Industry 4.0—Systematic Review, Challenges and Outlook. *IEEE Access*, *8*, 220121–220139. https://doi.org/10.1109/ACCESS.2020.3042874

Quarteroni, S. (2018). Natural Language Processing for Industry. *Informatik-Spektrum*, *41*(2), 105–112. https://doi.org/10.1007/s00287-018-1094-1

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). *"Why Should I Trust You?": Explaining the Predictions of Any Classifier* (arXiv:1602.04938). arXiv. https://doi.org/10.48550/arXiv.1602.04938

Rychener, Y., Renard, X., Seddah, D., Frossard, P., & Detyniecki, M. (2023). On the Granularity of Explanations in Model Agnostic NLP Interpretability. In I. Koprinska, P. Mignone, R. Guidotti, S. Jaroszewicz, H. Fröning, F. Gullo, P. M. Ferreira, D. Roqueiro, G. Ceddia, S. Nowaczyk, J. Gama, R. Ribeiro, R. Gavaldà, E. Masciari, Z. Ras, E. Ritacco, F. Naretto, A. Theissler, P. Biecek, … S. Pashami (Eds.), *Machine Learning and Principles and Practice of Knowledge Discovery in Databases* (pp. 498–512). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-23618-1_33

Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, *18*(11), 613–620. https://doi.org/10.1145/361219.361220

Sarica, S., & Luo, J. (2021). Stopwords in technical language processing. *PLOS ONE*, *16*(8), e0254937. https://doi.org/10.1371/journal.pone.0254937

Schwenke, L., & Atzmueller, M. (2021). *Show Me What You're Looking For: Visualizing Abstracted Transformer Attention for Enhancing Their Local Interpretability on Time Series Data*.

Sen, C., Hartvigsen, T., Yin, B., Kong, X., & Rundensteiner, E. (2020). Human Attention Maps for Text Classification: Do Humans and Neural Networks Focus on the Same Words? *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 4596–4608. https://doi.org/10.18653/v1/2020.acl-main.419

Serre, T., Kreiman, G., Kouh, M., Cadieu, C., Knoblich, U., & Poggio, T. (2007). A quantitative theory of immediate visual recognition. *Progress in Brain Research*, *165*, 33–56. https://doi.org/10.1016/S0079-6123(06)65004-8

Sexton, T., & Fuge, M. (2019). *Using Semantic Fluency Models Improves Network Reconstruction Accuracy of Tacit Engineering Knowledge*. https://doi.org/10.1115/DETC2019-98429

Shah, D., Campbell, W., & Zulkernine, F. H. (2018). A Comparative Study of LSTM and DNN for Stock Market Forecasting. *2018 IEEE International Conference on Big Data (Big Data)*, 4148–4155. https://doi.org/10.1109/BigData.2018.8622462

Shapley, L. S., & Roth, A. E. (Eds.). (1988). *The Shapley value: Essays in honor of Lloyd S. Shapley*. Cambridge University Press.

Siener, M., & Aurich, J. C. (2011). Quality oriented maintenance scheduling. *CIRP Journal of Manufacturing Science and Technology*, *4*(1), 15–23. https://doi.org/10.1016/j.cirpj.2011.06.014

Simon, H. A. (2008). *The sciences of the artificial* (3. ed., [Nachdr.]). MIT Press.

Sokolova, M., Japkowicz, N., & Szpakowicz, S. (2006). Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation. In A. Sattar & B. Kang (Eds.), *AI 2006: Advances in Artificial Intelligence* (Vol. 4304, pp. 1015–1021). Springer Berlin Heidelberg. https://doi.org/10.1007/11941439_114

Szczepański, M., Pawlicki, M., Kozik, R., & Choraś, M. (2021). New explainability method for BERT-based model in fake news detection. *Scientific Reports*, *11*(1), Article 1. https://doi.org/10.1038/s41598-021-03100-6

Tamekuri, A., Nakamura, K., Takahashi, Y., & Yamaguchi, S. (2022). Providing Interpretability of Document Classification by Deep Neural Network with Self-attention. *Journal of Information Processing*, *30*, 397–410. https://doi.org/10.2197/ipsjjip.30.397

Tang, Z., Hahn-Powell, G., & Surdeanu, M. (2020). Exploring Interpretability in Event Extraction: Multitask Learning of a Neural Event Classifier and an Explanation Decoder. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, 169–175. https://doi.org/10.18653/v1/2020.acl-srw.23

*The AI Index Report 2022 – Artificial Intelligence Index*. (2022, December). https://aiindex.stanford.edu/ai-index-report-2022/

Tunstall, L. (2022). *Natural Language Processing with Transformers*.

Turbé, H., Bjelogrlic, M., Lovis, C., & Mengaldo, G. (2022). *InterpretTime: A new approach for the systematic evaluation of neural-network interpretability in time series classification* (arXiv:2202.05656). arXiv. https://doi.org/10.48550/arXiv.2202.05656

Vashishth, S., Upadhyay, S., Tomar, G. S., & Faruqui, M. (2019). *Attention Interpretability Across NLP Tasks* (arXiv:1909.11218). arXiv. https://doi.org/10.48550/arXiv.1909.11218

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention Is All You Need* (arXiv:1706.03762). arXiv. https://doi.org/10.48550/arXiv.1706.03762

Wang, L., Shen, Y., Peng, S., Zhang, S., Xiao, X., Liu, H., Tang, H., Chen, Y., Wu, H., & Wang, H. (2022). *A Fine-grained Interpretability Evaluation Benchmark for Neural NLP* (arXiv:2205.11097). arXiv. https://doi.org/10.48550/arXiv.2205.11097

Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., & Zhou, M. (2020). *MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers* (arXiv:2002.10957). arXiv. http://arxiv.org/abs/2002.10957

Williams, A., Nangia, N., & Bowman, S. (2018). A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 1112–1122. https://doi.org/10.18653/v1/N18-1101

Xue, H., Chu, W., Zhao, Z., & Cai, D. (2018). A Better Way to Attend: Attention with Trees for Video Question Answering. *IEEE Transactions on Image Processing*, *27*(11), 5563–5574. https://doi.org/10.1109/TIP.2018.2859820

Yan, J., Meng, Y., Lu, L., & Li, L. (2017). Industrial Big Data in an Industry 4.0 Environment: Challenges, Schemes, and Applications for Predictive Maintenance. *IEEE Access*, *5*, 23484–23491. https://doi.org/10.1109/ACCESS.2017.2765544

Yi, H., Shiyu, S., Xiusheng, D., & Zhigang, C. (2016). A study on Deep Neural Networks framework. *2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, 1519–1522. https://doi.org/10.1109/IMCEC.2016.7867471

Zhang, H., Yao, W., & Yu, D. (2022). *Efficient Zero-shot Event Extraction with Context-Definition Alignment* (arXiv:2211.05156). arXiv. https://doi.org/10.48550/arXiv.2211.05156

Zhang, Z., Farnsworth, M., Song, B., Tiwari, D., & Tiwari, A. (2022). Deep Transfer Learning With Self-Attention for Industry Sensor Fusion Tasks. *IEEE Sensors Journal*, *22*(15), 15235–15247. https://doi.org/10.1109/JSEN.2022.3186505

Zini, J. E., & Awad, M. (2022). On the Explainability of Natural Language Processing

Deep Models. *ACM Computing Surveys*, *55*(5), 103:1-103:31. https://doi.org/10.1145/3529755

Zonta, T., da Costa, C. A., da Rosa Righi, R., de Lima, M. J., da Trindade, E. S., & Li, G. P. (2020). Predictive maintenance in the Industry 4.0: A systematic literature review. *Computers & Industrial Engineering*, *150*, 106889. https://doi.org/10.1016/j.cie.2020.106889

IDC, & Statista. (June 7, 2021). Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2020, with forecasts from 2021 to 2025 (in zettabytes) [Graph]. In Statista. Retrieved June 01, 2024, from https://www.statista.com/statistics/871513/worldwide-data-created/

# List of Figures

Figure 1 Volume of worldwide Data (Statista, 2024) ..... 1
Figure 2 Design Science Research Cycle based on (Hevner, 2007) ..... 6
Figure 3 Overall Architecture of TEEE ..... 8
Figure 4 Overall Architecture of PriMa (Ansari et al., 2019) ..... 13
Figure 5 Text Classification Pipeline (Kowsari et al., 2019) ..... 15
Figure 6 Zero-Shot Information Extraction Task Demonstration ..... 17
Figure 7 Structure of a DNN with Depth 1 (Lopez & Kalita, 2017) ..... 18
Figure 8 The Transformer Model Architecture (Vaswani et al., 2017) ..... 20
Figure 9 Steps of Systematic Literature Review ..... 25
Figure 10 Search String ..... 27
Figure 11 Screening Process ..... 29
Figure 12 Breakdown of Maintenance-Related Screening Results and Non-Maintenance-Related Screening Results ..... 33
Figure 13 Single Event Accuracy and F1 Score ..... 43
Figure 14 Multiple Event Accuracy and F1 Score ..... 44
Figure 15 Runtime Single- and Multiple Event ..... 44
Figure 16 Number of AI Publications in the World 2010-2021 (The AI Index Report 2022 – Artificial Intelligence Index, 2022) ..... 46
Figure 17 Responsible AI for the Domain of Manufacturing (Besinger et al., 2023) . 47
Figure 18 Black-box Model and White-box Model ..... 49
Figure 19 TEEE User Interface ..... 59
Figure 20 Overview Event Results Graph ..... 60
Figure 21 SHAP Values for the first Sentence ..... 61
Figure 22 Event Extraction Pipeline Architecture ..... 62
Figure 23 Explainability Plot with Shapley Values ..... 63
Figure 24 Overall Result Plot ..... 64
Figure 25 Explainability Plot Motor Heat (English Dataset) ..... 69
Figure 26 Explainability Plot Preventive Measures (English Dataset) ..... 69
Figure 27 Explainability Plot Kühlungsprobleme (German Dataset) ..... 70
Figure 28 Explainability Plot Filterstaub (German Dataset) ..... 71
Figure 29 Code of the Event Extraction Pipeline ..... 90

# List of Tables

Table 1 Problems P1, P2 and P3 ..... 4
Table 2 Sub Research Questions for P1, P2 and P3 ..... 5
Table 3 Objectives of this Thesis ..... 5
Table 4 Exclusion Criteria ..... 27
Table 5 Selected Publications from Screening ..... 30
Table 6 Example Entries with Corresponding Events from the Various Datasets ..... 38
Table 7 Quantitative Evaluation Transformer Models 1 & 2 ..... 41
Table 8 Quantitative Evaluation Transformer Models 3 & 4 ..... 41
Table 9 Feature Matrix of Perturbation-, Gradient-, Propagation-, and Attention-Based Explainability Methods ..... 52
Table 10 Taxonomy for the Explainability of Transformer Models ..... 54
Table 11 Fictional Example Sentence with Evaluation Events ..... 62
Table 12 English Evaluation Text and Events (English Dataset) ..... 66
Table 13 German Evaluation Text and Events (German Dataset) ..... 66
Table 14 Quantitative Evaluation of TEEE Runtime (English Dataset) ..... 67
Table 15 Quantitative Evaluation of TEEE Runtime (German Dataset) ..... 67
Table 16 Additional Publications ..... 89

# Appendix

**Table 16** Additional Publications

| Article | Type | First selection source |
|---|---|---|
| (Castelvecchi, 2016) | Journal Article | Google Scholar |
| (Doshi-Velez & Kim, 2017) | Preprint | Google Scholar |
| (Lundberg & Lee, 2017) | Preprint | Google Scholar |
| (Ribeiro et al., 2016) | Preprint | Google Scholar |
| (Shapley & Roth, 1988) | Book | Google Scholar |
| (Dignum, 2017) | Journal Article | Google Scholar |
| (Benjamins et al., 2019) | Preprint | Google Scholar |
| (Ansari et al., 2019) | Journal Article | Google Scholar |
| (Marques & Giacotto, 2019) | Journal Article | Google Scholar |
| (Dankar et al., 2023) | Preprint | Google Scholar |
| (Howard & Ruder, 2018) | Preprint | Google Scholar |
| (Sarica & Luo, 2021) | Journal Article | Google Scholar |
| (Ansari et al., 2021) | Journal Article | Science Direct |
| (Besinger et al., 2023) | Journal Article | Science Direct |
| (Gandomi & Haider, 2015) | Journal Article | Science Direct |
| (Brundage et al., 2021) | Journal Article | Science Direct |
| (C. Li et al., 2022) | Journal Article | Science Direct |
| (Gilpin et al., 2019) | Preprint | IEEE Xplore |
| (Yan et al., 2017) | Journal Article | IEEE Xplore |

```python
# Constants
MODEL_NAME = 'MoritzLaurer/multilingual-MiniLMv2-L6-mnli-xnli'
count_clicks = 0

# Load Hugging Face model and tokenizer
model = AutoModelForSequenceClassification.from_pretrained(MODEL_NAME)
tokenizer = AutoTokenizer.from_pretrained(MODEL_NAME, use_fast=False)

# Creation of individual EventExtractionPipeline, based on the HuggingFaceZeroShotClassificationPipeline
class EventExtractionPipeline(ZeroShotClassificationPipeline):
    def __call__(self, *args):
        call_super = super().__call__(args[0], self.new_labels)[0]
        return [[{'label': x[0], 'score': x[1]} for x in zip(call_super['labels'], call_super['scores'])]]

    def set_new_labels(self, labels):
        self.new_labels = labels


# Processing of text and labels
def process_text_and_labels(stored_text, stored_events_list):
    # Update the model's label2id and id2label configurations based on the given labels
    model.config.label2id.update({v: k for k, v in enumerate(stored_events_list)})
    model.config.id2label.update({k: v for k, v in enumerate(stored_events_list)})

    # Create an instance of the EventExtractionPipeline with the defined HuggingFace model and tokenizer
    event_extraction_pipe = EventExtractionPipeline(model=model, tokenizer=tokenizer)
    event_extraction_pipe.set_new_labels(stored_events_list)

    # Split the example text by periods to get individual sentences and clean them
    temp = [s.strip() + '.' for s in stored_text.split('.') if s]

    # Lists to store events and their associated scores for all sentences
    event_list_all = []
    score_list_all = []

    # Process each sentence to predict events
    for sentence in temp:
        prediction = event_extraction_pipe([sentence])
        event_score_list = prediction[0]
```

**Figure 29** Code of the Event Extraction Pipeline