CIRP Annals - Manufacturing Technology 73 (2024) 361–364

Contents lists available at ScienceDirect

CIRP Annals - Manufacturing Technology

journal homepage: https://www.editorialmanager.com/CIRP/default.aspx

Ontology-integrated tuning of large language model for intelligent maintenance

Peng Wang[a], John Karigiannis[b], Robert X. Gao (1)[c,*]

[a] Department of Electrical and Computer Engineering, Department of Mechanical and Aerospace Engineering, University of Kentucky, Lexington, KY, USA [b] GE Research, Niskayuna, NY, USA
[c] Department of Mechanical and Aerospace Engineering, Case Western Reserve University, Cleveland, OH, USA

ARTICLE INFO

Article history:
Available online 6 June 2024

Keywords:
Maintenance
Machine learning
Large language models

ABSTRACT

As new AI technologies such as Large Language Models (LLM) quickly evolve, the need for enhancing generalpurpose LLMs with physical knowledge to better serve the manufacturing community has been increasingly recognized. This paper presents a method that tailors GPT-3.5 with domain-specific knowledge for intelligent aircraft maintenance. Specifically, aircraft ontology is investigated to curate maintenance logs with encoded component hierarchical structure to fine-tune GPT-3.5. Experimental results demonstrate the effectiveness of the developed method in accurately identifying defective components and providing consistent maintenance action recommendations, outperforming general-purpose GPT-3.5 and GPT-4.0. The method can be adapted to other domains in manufacturing and beyond.

1. Introduction

Over the past few years, the field of Artificial Intelligence (AI) has undergone a significant transformation enabled by Large Language Models (LLMs). Examples include the Large Language Model Meta AI by Meta [1], Generative Pre-trained Transformer 3 (GPT-3) and the subsequent ChatGPT (GPT-3.5) and GPT-4.0 by OpenAI [2], and Bidirectional Encoder Representations from Transformers (BERT) and Gemini by Google [3]. These models are predominantly built upon the transformer architecture and drastically advanced the field of Natural Language Processing (NLP) by enabling models to be pre-trained on extensive datasets from the Internet, and to learn language structures and nuances without explicitly labelled data through self-supervised learning [4]. After meta-training, the pre-trained models can be aligned to human preferences to enhance their relevance and applicability across a broad spectrum of language-based tasks.

As an example, LLM models are widely recognized for their text-generation capabilities, making them advantageous in content creation from drafting news articles to generating creative fictions. Their proficiency in understanding languages has been leveraged for developing chatbots and virtual assistants, enhancing customer services and interactive experiences [5]. With a foundation in deep learning, LLMs have shown effectiveness in tackling diverse, complex tasks such as information extraction, summarization, and coding assistance [6,7], further highlighting their pivotal in accelerating the evolution of general-purpose AI.

With their increasing popularity, the limitations of LLMs in performing domain-specific tasks, such as predictive maintenance and automation, have also been increasingly noted. This is because LLMs are trained on broad-based datasets that do not necessarily cover specialized technical data pertinent to manufacturing [8]. The lack of domain knowledge in general-purpose LLMs can lead to difficulty in proper contextual understanding, which is a prerequisite for correctly interpreting the nuances of domain-specific terminologies and processes. For example, in predictive maintenance, if a machine contains multiple components with the same part name (e.g., Seal), these components must be correctly associated with the machine′s hierarchical structure to avoid confusion for reliable performance analysis and maintenance decisions. This requires the LLMs to understand the component hierarchical structure and correctly interpret the maintenance logs. The issue is exacerbated by the fact that the description of the same component recorded in the maintenance logs may vary from worker to worker, leading to erroneous analysis and maintenance recommendations based on the reasoning of general-purpose LLMs. To bridge this gap, using domain knowledge to fine-tune LLMs has emerged as a solution.

The process of fine-tuning involves training the existing LLMs on datasets that are curated for the target domain, allowing maximizing domain-specific performance while retaining the reasoning capability of the initial model. For example, BERT has been fine-tuned with climate data, resulting in ClimateBERT that has improved BERT′s performance in climate-related tasks [9]. Also, GPT has been tailored to KAI-GPT, a language model for transparent, accurate, and safe customer banking service [10].

This study investigates LLMs for manufacturing by fine-tuning GPT3.5 and converting it to an intelligent maintenance assistant for aircraft (see Fig. 1). Towards this end, the ontology of aircraft structure is first investigated to curate the original maintenance logs into conversational data. The goal is to alleviate the constraints of typical LLM fine-tuning

* Corresponding author.
E-mail address: robert.gao@case.edu (R.X. Gao).

https://doi.org/10.1016/j.cirp.2024.04.012

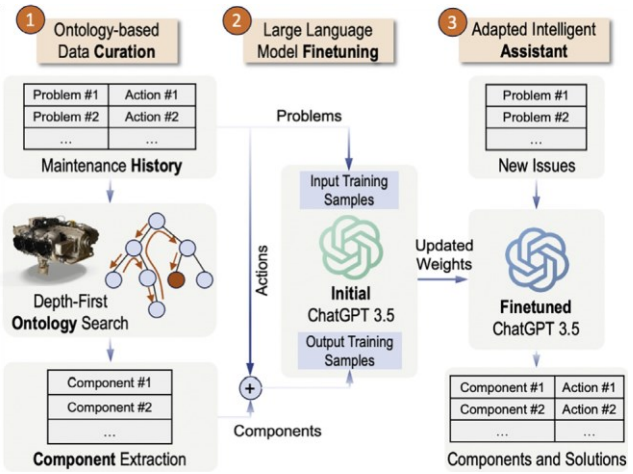


Fig. 1. Ontology-integrated tuning of LLM for predictive maintenance.

approaches from overly reliant on structured data that limits model generalizability and adaptivity. The conversational data is used as inputs to fine-tune the GPT-3.5 model, achieving contextual understanding of components' hierarchical structure. This sets the technological basis for pinpointing defective components and recommending consistent maintenance actions. Results from a case study using aircraft have shown that the finetuned GPT-3.5 model outperforms both the general-purpose GPT-3.5 and GPT-4.0 models in effectively identifying defective components, thereby better supporting predictive aircraft maintenance. The developed method holds significant promise beyond aviation, extendable to various manufacturing scenarios where maintenance and production systems share similar challenges and requirements.

2. Ontology-integrated tuning of LLM

Central to the developed approach is the ontology of the aircraft, which plays a crucial role in enabling maintenance log curation and domain-specific fine-tuning of the LLMs.

2.1. Aircraft structure ontology

To realize ontology-integrated LLM tuning as illustrated in Fig. 1, a tree structure representing a segment of an aircraft's ontology is developed, with a top-level concept of the ontology shown in Fig. 2. This hierarchical organization is crucial to curating the maintenance logs and enhancing LLMs' comprehension of aircraft-specific issues, for two reasons. First, it addresses ambiguity when the same type of components appears in different branches of the structure. For instance, the component "Seal" is found not only in the engine cylinder baffle but also in the engine cylinder intake. Through the process of learning to describe each component within its hierarchical context, LLMs acquire the ability to distinguish and correctly identify these components, thereby resolving the ambiguity. Second, the structure aids in reducing the impact of inconsistencies commonly found in maintenance logs that are created by multiple workers. Depending on their training, personal choice, and company preference, variation may occur when the logs are entered. A common cause is the use of abbreviated descriptions, which omits the full hierarchical chain of a component. For example, references to "engine" and "cylinder" are often left out when mentioning related

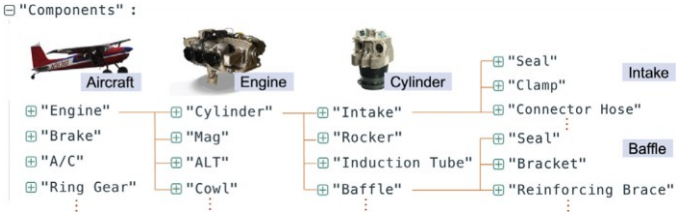
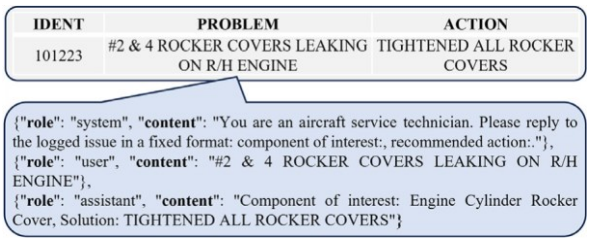


Fig. 2. Aircraft ontology in a hierarchical structure.

sub-components. This leads to inaccurate interpretations by LLMs when not trained with knowledge of the hierarchical structure, and consequently, unreliable analysis. Furthermore, incorporating additional domain-specific relationships like component-wear mechanism-maintenance triplets into the LLM is envisioned to enable causal reasoning underlying the maintenance recommendation. For this purpose, comprehensive domain knowledge collection is needed and will constitute part of future research.

2.2. GPT fine-tuning

In this work, GPT-3.5 (containing 175 billion parameters) instead of the state-of-the-art GPT-4.0 (containing 1 trillion parameters) is selected as the foundational LLM model for fine-tuning, due to its demonstrated performance in general-purpose NLP task handling and user-friendly access to its fine-tuning resources through OpenAI's Application Programming Interface (API) [11]. Central to the fine-tuning effort is the preparation of domain-specific data and iterative refinement of the fine-tuning hyperparameters such that the tailored optimization of GPT-3.5 ensures reliable maintenance actions. Finetuning GPT-3.5 requires the training data to be prepared in a specific conversational format where each conversation sample contains three messages, and each message specifies a role (system, user, or assistant) and the related content. For example, the system's message specifies the purpose of fine-tuning, while the user's message simulates the questions/messages asked by human users. Finally, the assistant's message indicates the responses generated by the fine-tuned model. In Fig. 3a conversation sample used for model fine-tuning is illustrated.



IDENT	PROBLEM	ACTION
101223	#2 & 4 ROCKER COVERS LEAKING ON R/H ENGINE	TIGHTENED ALL ROCKER COVERS

{"**role**": "system", "**content**": "You are an aircraft service technician. Please reply to the logged issue in a fixed format: component of interest:, recommended action:."},
{"**role**": "user", "**content**": "#2 & 4 ROCKER COVERS LEAKING ON R/H ENGINE"},
{"**role**": "assistant", "**content**": "Component of interest: Engine Cylinder Rocker Cover, Solution: TIGHTENED ALL ROCKER COVERS"}

Fig. 3. Sample fine-tuning system, user, and assistant message.

Domain-specific knowledge is typically organized in a tree-structure and has been extracted from historical data using a similarity measure [12]. In the presented work, historical maintenance logs have been explored to provide the basis for developing an aircraft ontology, which is subsequently verified by domain experts. The ontology is then incorporated into the training data for fine-tuning GPT-3.5 by curating the original maintenance log into a conversational format, where a new content "component of interest" that pinpoints the hierarchical structure of the defective component is added, as illustrated in Fig. 3. The ground truth "component of interest" is first generated from "in-order traversal" algorithm-based depth-first search operations [13], which compare the problem description to full hierarchical ontology structure. The outcome is then examined by a domain expert to remove the ambiguities. Compared to other options of incorporating ontology into training data, e.g., decomposing the entire ontology into conversational forms to indicate the hierarchical structural organization of the components and mixing them with maintenance log conversations, the presented approach is advantageous in implicitly integrating ontology into conversation samples that iteratively reinforce the GPT-3.5 model's understanding of the aircraft structural composition.

The process of fine-tuning GPT-3.5 primarily utilizes a variant of the cross-entropy loss function, which is used in training LLMs based on the transformer architecture. In the context of language models, the cross-entropy loss function quantifies how well the model's predicted probability distribution over the next word qðxÞ aligns with the actual word pðxÞ that appears in the training data:

H pð ; qÞ ¼ X p xð Þlogq xð Þ ð1Þ

x

One-hot encoding is used for calculating the true distribution pðxÞ when fine-tuning GPT-3.5. The actual next word is set with a probability of 1 whereas all other words are set with a probability of 0. Three hyperparameters for GPT-3.5 fine-tuning are provided by the OpenAI API: number of epochs, learning rate, and batch size. Selection of proper values of these hyperparameters depends on the size and quality of the fine-tuning dataset, as well as the complexity of the domain-specific applications. Due to the lack of advanced options for fine-tuning and overfitting mitigation such as early stop of model tuning, these hyperparameters are empirically determined during the fine-tuning process in this study.

2.3. Evaluation of fine-tuned LLM model

To evaluate the quality of the fine-tuned GPT-3.5 in extracting the component of interest from the problem description, the Intersection over Union (IoU) score is used, which quantifies the degree of word matching between the ground truth and model response. The IoU score is suited for evaluating structured outputs when constrained to a limited number of vocabularies. The IoU score ranges from 0 to 1, where 0 indicates complete irrelevancy of the extracted information and 1 denotes perfect extraction.

To quantify the performance of the fine-tuned GPT-3.5 in predicting recommended maintenance actions given a problem description, the semantics of the output must be considered, rather than relying on word matching only. The potential large variation in describing the same actions makes it difficult to predetermine a constrained list of candidates. To overcome this challenge, BERTScore is investigated. BERTScore leverages the contextual embedding from BERT [3], which is a special transformation that maps language vocabularies into a highdimensional space such that semantically similar words are clustered together while words with little semantic similarity are separated [14]:

$$P = \frac{1}{|C|}\sum_{c \in C} \max_{g \in G} \cos\left(\mathrm{BERT}_{emb}(c), \mathrm{BERT}_{emb}(g)\right)$$

$$R = \frac{1}{|G|}\sum_{g \in G} \max_{c \in C} \cos\left(\mathrm{BERT}_{emb}(c), \mathrm{BERT}_{emb}(g)\right)$$

$$\mathrm{BERTScore} = 2\frac{P \cdot R}{P + R}$$

where C and G can be intuitively considered as the list of words in the predicted text and the reference ground truth, respectively. BERT_{emb} is the function to compute the embedding of each entry in C and G. $\cos(\mathrm{BERT}_{emb}(c), \mathrm{BERT}_{emb}(g))$ refers to the cosine similarity between the embeddings from the predicted and the reference texts. P and R are analogous to precision and recall, which average the maximum cosine similarities for each entry in the predicted text over the entries in the reference text, and vice versa, leading to the final calculation of BERTScore in (4).

In addition to the BERTScore, BLEU (Bilingual Evaluation Understudy) [14] is considered for evaluating how the LLM-recommended maintenance action follows similar vocabulary as the referenced ground truth. BLEU examines the frequency of n-grams (sequential groups of n words), which appear in both the prediction and the ground truth. The metric calculates a score based on the matches of the n-grams as [15]:

$$\mathrm{BLEU} = \mathrm{BP}\exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

where p_n is the ratio of the number of n-grams in the predicted text that matches the ground truth to the total number of n-grams in the predicted text, w_n is the weight assigned to each n-gram (set uniformly in this study), and N denotes the maximum length of n-grams used. Brevity Penalty (BP) is designed to penalize predicted texts that are too short compared to the reference, ensuring that shorter texts don't unfairly receive higher scores due to a higher likelihood of n-gram matching [15]. BERTScore and BLEU range from 1 to 1 and 0 to 1, respectively, with higher values indicating higher similarity between the predicted and the reference texts.

3. Experimental evaluation and results

The developed method is evaluated using a publicly available aviation maintenance dataset [16]. The dataset contains 6169 maintenance logs, each is represented by a triplet of problem identification number (IDENT), problem description (PROBLEM), and maintenance action that has been taken (ACTION) (see Fig. 3 for an example). Upon examination, significant repetitions are identified after the first 2000 logs. As a result, the first 2000 logs were chosen for this study. Among the 2000 logs, 1500 were randomly selected to fine-tune the GPT 3.5 model while the remaining 500 logs were reserved for testing. Considering the generative nature of GPT, each testing log is evaluated five times when evaluating the performance of the fine-tuned GPT3.5 in terms of randomness in its response generation.

3.1. GPT 3.5 fine-tuning

Restricted by the fine-tuning API, only three hyperparameters (i.e., epoch, batch size, and learning rate) are tuneable, and no advanced training mechanisms (e.g., early stopping) are provided. Iterative hyperparameter refinement has been conducted, and a combination of default batch size, learning rate and one training epoch has yielded the most satisfactory performance, as shown in Fig. 4. Beyond the 1st training epoch, severe overfitting is observed, as reflected in the divergence between training and validation losses. The fast convergence indicates that the structure ontology as formulated in this study are relatively straightforward to learn by GPT-3.5, which was pretrained using a large dataset. This highlights the advantage of leveraging a general-purpose LLM for specific problem-solving in manufacturing.

Fig. 4. Progression of GPT-3.5 fine-tuning.

3.2. Results and discussions

To evaluate the performance of the fine-tuned GPT-3.5 on generating domain-specific responses to airplane maintenance logs, three LLMs are compared: fine-tuned GPT-3.5 (GPT-3.5 FT), non-fine-tuned GPT-3.5 (GPT-3.5 NFT), and non-fine-tuned GPT-4.0 (GPT-4.0 NFT). Shown in Fig. 5 are sample responses from these 3 LLMs. Each response contains two parts: component of interest and recommended maintenance actions. It is noted qualitatively that GPT-3.5 FT outperforms both GPT-3.5 NFT and GPT-4.0 NFT in general, especially in identifying which component the maintenance log was referring to. This is

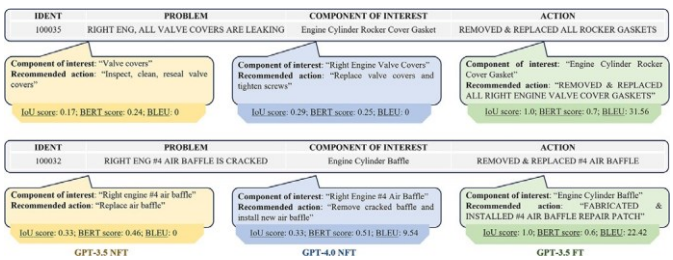


Fig. 5. Sample responses from GPT-3.5 NFT, GPT-4.0 NFT, and GPT-3.5 FT, and their evaluation scores.

because the non-fine-tuned GPTs use the keywords directly from the original problem description to output their responses. For example, they extracted "Right Engine #4 Air Baffle" from the problem description "RIGHT ENG#4 AIR BAFFLE IS CRACKED" only, whereas GPT-3.5 FT is able to trace back the entire hierarchical structure for improved accuracy in defective component identification.

To quantitively assess the LLMs' performance, IoU score is first calculated to evaluate the models' response in determining component of interest. Subsequently, BERTScore and BLEU are computed to quantify the models' performance on recommending actions. Considering the generative nature of the LLM models, each metric is computed five times to evaluate the consistency and variation of the models' responses.

As shown in Fig. 6(a), GPT-3.5 FT achieves a mean IoU score of 0.88, whereas the IoU scores of GPT-3.5 NFT and GPT-4.0 NFT are 0.43 and 0.40, respectively. The good match with the ground truth achieved by GPT-3.5 is particularly noteworthy given the more sophisticated architecture of GPT-4.0 as compared to GPT-3.5. The result further highlights the importance of fine-tuning of generalpurpose LLM models in adaptation to manufacturing applications.

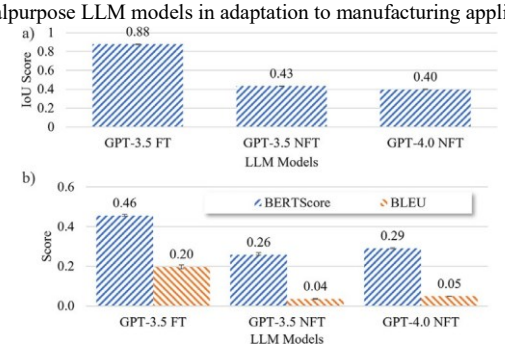


Fig. 6. Comparison among GPT-3.5 FT, GPT-3.5 NFT, and GPT-4.0 NFT: a) IoU scores for extraction of component of interest, b) BERTScore and BLEU for prediction of recommended maintenance actions.

The effectiveness of model fine-tuning is further substantiated by the mean BERTScore and BLEU of GPT-3.5 FT for predicting recommended actions, as seen in Fig. 6(b). The mean BERTScore of 0.46 indicates robust consistency between the predicted actions and reference ground truth in terms of the text semantics, whereas the BLEU metric, at 0.20, reflects a reasonable n-gram overlap with the reference texts. In comparison, both non-fine-tuned models have shown less favorable predictions for recommended maintenance actions.

The small error bars observed across all three models indicate a high level of consistency in the models' performance across the five different tests conducted. This aspect is critical to predictive maintenance for manufacturing, where reliability and repeatability of performance are essential. Given the consistency demonstrated by all model variants, the development of LLM architectures provides a potentially stable foundation for predictive maintenance tasks.

4. Conclusions

This paper introduced an innovative approach to transforming general-purpose LLMs into a domain-specific tool for intelligent aircraft maintenance. Incorporating an aircraft structure ontology into the fine-tuning process of GPT-3.5 enhances the model's performance in identifying aircraft components of interest and recommending maintenance actions. The enhanced performance of the fine-tuned GPT-3.5 over GPT-3.5 and GPT-4.0 in maintenance log analysis, e.g., 0.88 vs. 0.43 and 0.40 in identifying components of interests, not only demonstrates the feasibility of tailoring LLMs for enhanced operations in manufacturing because of the similarities in maintenance activities across different domains, but also sheds light on their continued evolution and roader applications in other fields of interest.

Future research will explore fine-tuning of LLMs with expanded domain knowledge (including both ontology and domain-specific relationships and attributes as represented by knowledge graphs), and further investigate topics such as data bias and interpretability to facilitate transfer learning and domain generalization across multiple industrial sectors and more effectively integrate LLM into the existing digital manufacturing platform for more comprehensive and versatile AI-enhanced applications.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Peng Wang: Writing review & editing, Visualization, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. John Karigiannis: Resources, Data curation, Conceptualization. Robert X. Gao: Writing review & editing, Visualization, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization.

Acknowledgments

The authors would like to thank Dr. Jianjing Zhang from Case Western Reserve University for his valuable contributions. Support from the National Science Foundation under awards CMMI-2237242, CNS-2125460, and EEC-2133630 (Engineering Research Center on Hybrid and Autonomous Manufacturing Moving from Evolution to Revolution) is sincerely appreciated.

References

[1] Touvron H, Lavril T, Izacard G, Martinet X, Lachaux MA, et al. (2023) Llama: Open And Efficient Foundation Language Models. arXiv:2302.13971.
[2] Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, et al. (2020) Language Models Are Few-Shot Learners. Advances in Neural Information Processing Systems 33: 1877–1901.
[3] Devlin J, Chang MW, Lee K, Toutanova K (2018) Bert: Pre-Training Of Deep Bidirectional Transformers For Language Understanding. arXiv:1810.04805.
[4] Min B, Ross H, Sulem E, Veyseh A, Nguyen TH, et al. (2023) Recent Advances In Natural Language Processing Via Large Pre-Trained Language Models: A Survey. ACM Computing Surveys 56(2):1–40.
[5] Ray PP (2023) ChatGPT: A Comprehensive Review On Background, Applications, Key Challenges, Bias, Ethics, Limitations And Future Scope. Internet of Things and Cyber-Physical Systems 3:121–154.
[6] Akay H, Kim SG (2021) Reading Functional Requirements Using Machine Learning-Based Language Processing. CIRP Annals 70(1):139–142.
[7] Addepalli S, Weyde T, Namoano B, Oyedeji OA, Wang T, et al. (2023) Automation of Knowledge Extraction For Degradation Analysis. CIRP Annals 72(1):33–36.
[8] Zhang S, Dong L, Li X, Zhang S, Sun X, et al. (2023) Instruction Tuning For Large Language Models: A Survey. arXiv:2308.10792.
[9] Webersinke N, Kraus M, Bingler JA, Leippold M (2021) Climatebert: A Pretrained Language Model For Climate-Related Text. arXiv:2110.12010.
[10] Evanini K KAI-GPT: The First Large Language Model Purpose-Built for Banking, https://kasisto.com/blog/kai-gpt-the-first-large-language-model-purpose-builtfor-banking/, May 2023.
[11] https://platform.openai.com/docs/guides/fine-tuning, August 2023.
[12] Chen M, Qu R, Fang W (2022) Case-Based Reasoning System For Fault Diagnosis Of Aero-Engines. Expert Systems with Applications 202:117350.
[13] Tarjan R (1972) Depth-First Search And Linear Graph Algorithms. SIAM Journal on Computing 1(2):146–160.
[14] Zhang T, Kishore V, Wu F, Weinberger KQ, Artzi Y (2019) Bertscore: Evaluating Text Generation With Bert. arXiv:1904.09675.
[15] Papineni K, Roukos S, Ward T, Zhu WJ (2002) Bleu: A Method For Automatic Evaluation Of Machine Translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, 311–318.
[16] Akhbardeh F, Desell T, Zampieri M (2020) MaintNet: A Collaborative Open-Source Library for Predictive Maintenance Language Resources. In: Proceedings of the 28th International Conference on Computational Linguistics: System Demonstrations, 7–11.