SPEECH-TO-JOBSHOP: AN ONTOLOGY-DRIVEN DIGITAL ASSISTANT FOR SIMULATION MODELING

Heiner Ludwig[1], Vincent Betker[1], Thorsten Schmidt[1], and Mathias Kühn[1]

[1]Institute of Material Handling and Industrial Engineering, TUD Dresden University of Technology, e-mail: *firstname.lastname*@tu-dresden.de

KEYWORDS

Simulation Modeling; Digital Assistant; Large Language Models; Ontology; Job Shop Scheduling

ABSTRACT

This paper introduces a novel method utilizing speech-based digital assistants and large language models (LLMs) to streamline the creation of simulation models for Job Shop Scheduling Problems (JSSP). The system simplifies the process by allowing natural language interactions for ontology-based model generation. The study evaluates the performance of various LLMs in ontology-based simulation modeling by benchmarking their ability to extract and assign semantical entities and relations. We found that *ChatGPT-4-Turbo* is able to correctly identify all model elements given in descriptions of the production scenarios we tested, while less resource-intensive and open source models like *Mixtral-8x7b* and *Zephyr-beta* perform well in a less complex scenario. The findings demonstrate the potential of integrating LLMs and natural language processing in simulation modeling, significantly enhancing efficiency and reducing the need for manual modeling.

INTRODUCTION

The JSSP as a classical optimization problem from the field of production planning is a widely studied area of research. The aim is to determine the optimal sequence in which the orders are to be processed on the machines in order to optimize various target parameters. These include various time- and cost-based, human- and environment-centered parameters (Destouet et al., 2023). The development of an adequate simulation model is often complex and time-consuming and requires specially qualified staff or external experts. In particular, the modeling of different scenarios often involves a large amount of manual work (Agalianos et al., 2020). Destouet et al. (2023) emphasize the need for quickly adaptable simulation models in production systems in order to incorporate unforeseen deviations and thus increase the resilience of the overall production process. Especially in brownfield applications there is no homogeneous, digital infrastructure for automatically generating or updating the simulation models, so that the information needed has to be inserted manually. To reduce complexity, merge heterogeneous data and quickly adjust model parameters, Xiaochen Zheng and Kiritsis (2022) and May et al. (2022) recommend the use of semantic data structures such as knowledge graphs and ontologies, which serve as the foundation for digital twins of production systems. However, Khadir et al. (2021) argue that ontology creation is "time-consuming, error-prone and the maintenance is laborious" and refer to the methods of "ontology learning", which deal with the automatic creation of ontologies using various data sources. In order to minimize the effort required to create and maintain production simulation models and the underlying ontology, we investigate the suitability of a speech-based digital assistant for creating a job shop simulation in this paper. For this purpose, we present a new method based on LLMs to formalize the natural language input into a job shop ontology, which is validated and fed into a simulation tool.

BACKGROUND

Simulation modeling involves domain experts and simulation experts. While the former can articulate the inner workings of real-world systems, the latter develops a conceptual model from this knowledge and translates it into a simulation model. While this synergy aims to replicate and analyze real systems within simulated environments accurately, it is not only time and resource-intensive, but also a recurrent task due to the iterative nature of the simulation modell development process (see Fig. 1). Integrating LLM-based voice assistants to streamline and enhance this process through natural language interactions could reduce the effort required in the model development phase of a simulation study, enabling faster model validation cycles and thus faster simulation studies.

The use of voice-based digital assistants in the production environment is a growing field of research (Ludwig et al., 2023). The intuitive, flexible use and the high information density of natural language make

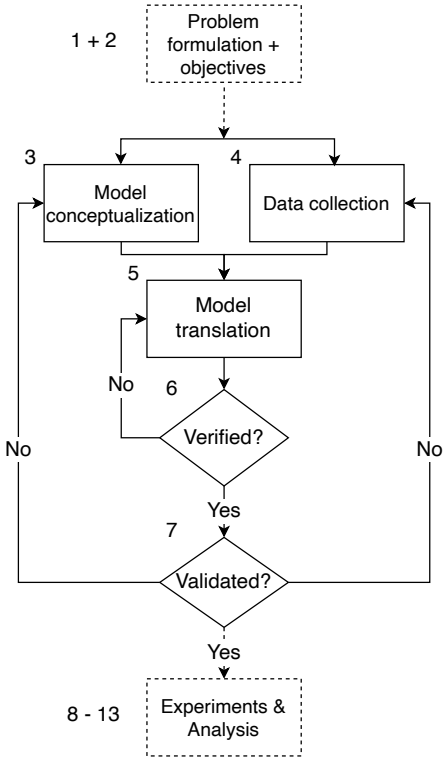

Figure 1: The modelling process as part of the steps needed to conduct a simulation study according to Banks (2010, p. 15).

the potential of a voice assistant obvious. The employee can move around the store floor and, in dialog with the digital assistant, describe the structure that he or she sees in real life, significantly reducing the abstractness of the model description. LLMs, such as *ChatGPT* or *Mixtral-8x7b* (Jiang et al., 2024), allow the comprehensive evaluation of the content of language input, making it possible to evaluate and formalize complex system descriptions. Automatically enriched text inputs (so-called "prompts") incorporate already defined entities from the ontology into the context of the employee's speech input, so that the application focus is predetermined. The synergetic combination of LLMs and semantic knowledge structures combines the respective advantages and provides a data structure that can be further processed by other system components. In this way, the language understanding and generalizability of the LLMs are combined with the clearly defined form, visibility and extensibility of semantical data structures (Pan et al., 2024).

We connect the research field "Ontology-based Production Simulation" for the integration of ontologies into simulation models with the field of "Ontology Learning", to create and extend ontologies automatically (e.g. based on text input) using the *Web Ontology Language* (OWL).

Ontology-based Production Simulation

The use of ontologies as a basis for setting up and operating production simulation systems offers potential in terms of flexibility, transferability, reusability and expressiveness. Salman Saeidlou and Jules (2019) show a multi-agent job store scheduling system that validates and exchanges information between different system components through a manually created ontological data structure. Zhu et al. (2019) also show the potential of a semantic graph for modeling a JSSP with regard to the definition of simulation-relevant entities and restrictions. May et al. (2022) synchronize the states of the simulation model with the stored ontology and thus enable the replicability of individual situations in order to use them to simulate different scenarios. Serrano-Ruiz et al. (2022) point out the advantage of creating a clear terminology with the help of ontologies in an industrial context and use the flexibility to model faults in the simulation model.

Ontology Learning

In order to exploit the aforementioned advantages of using ontologies to address the JSSP, but with minimal manual modeling effort and improved accessibility, we use ontology learning methods for an automatic, dialog-like construction of the simulation models. The automatic generation and population of ontologies has been studied for decades (Wong et al., 2012). Khadir et al. (2021) distinguish between two main approaches: *Linguistic and statistical approaches* and *Machine learning approaches*. In the former method, statistical and stochastic models are combined with linguistic properties. This includes common methods, such as *term frequency-inverse document frequency* to assess the importance of individual words, or the use of *lexico-syntactic patterns* to analyze sentence structures. The second approach builds on this and uses machine learning methods to accurately evaluate language. These include modern transformer-based deep learning models that are suitable for creating semantic graphs, such as generalized models *like GPT-3* (Brown et al., 2020) and specialized models like *REBEL* (Huguet Cabot and Navigli, 2021). Further studies show the potential of LLMs for generating semantic graphs, although for other contexts (Bellan et al., 2023). Trajanoska et al. (2023) show the advantage of generalized over specialized language models, as these also take implicitly expressed information into account in the modeling. For this reason, we decide to use generalized LLMs and compare them in terms of ontological modeling abilities using natural language input.

Web Ontology Language

We use OWL as the semantic data structure, as it provides well-defined entities and relationship types as well as restriction definitions (W3C, 2004). Due to the widespread use of OWL, we assume that the concepts occur extensively in the training data of the LLMs. Therefore simple prompting instructions are sufficient for extracting OWL entities and relations from the natural language input. Formally, an OWL ontology O

consists of the following concepts:

$$O = (TBox; ABox)$$

$$where$$

$$TBox = (C; R_O; R_D; A)$$
$$ABox = (I; R_{O;A}; R_{D;A})$$

The *Terminological Box* (TBox) contains the concepts of the domain and specifies description rules. C is a set of classes that represent the concepts of entities (e.g. Machine, Job, Task). R_O is a set of object properties that describe relations between entities (e.g. Job has Task). R_D are relations that assign attributes to entities (e.g. Task hasDuration integer). A are axioms for creating constraints (e.g. a Job has at least one Task) and can be evaluated with the help of reasoning algorithms.

The *Assertional Box* (ABox) contains the concrete individuals I (e.g. "Drilling Machine 2000") and their assertions $R_{O;A}$ and $R_{D;A}$, which refer to the concepts of C, R_O and R_D of the TBox.

METHOD

We present a new method for language-based creation of simulation models that combines the capabilities of LLMs with those of ontologies. This involves going through four different stages for the iterative enrichment and validation of a given simulation ontology, which serves as the basis for the simulation application (see Fig. 2). We describe the individual components in more detail below:

Dialog Handling

The dialog handling module is the dialog interface for the employee. It translates the voice input into text and uses synthesized voice output to ask questions if information is unclear or missing. We use *Whisper* (Radford et al., 2023) as a speech-to-text converter, the *SpeechSynthesis-Web API* (MozDevNet, 2023) generates the speech output.

Entity And Relation Extraction

The extraction of OWL elements is a central component of the overall system. Based on the input text and the existing ontology content, the system creates prompts that contain instructions for extracting and classifying spoken information. Simon et al. (2023) has already shown the basic suitability for formalizing data based on text sources using the example of *ChatGPT3*. However, since there is a wide range of LLMs that have various advantages and disadvantages (in terms of cost, availability, resource requirements, data protection, etc.), we compare various models in the evaluation chapter with regard to their performance in entity and relation extraction. We use an ontology reasoner to infer types of individuals that only occur in data property assertions or object property assertions

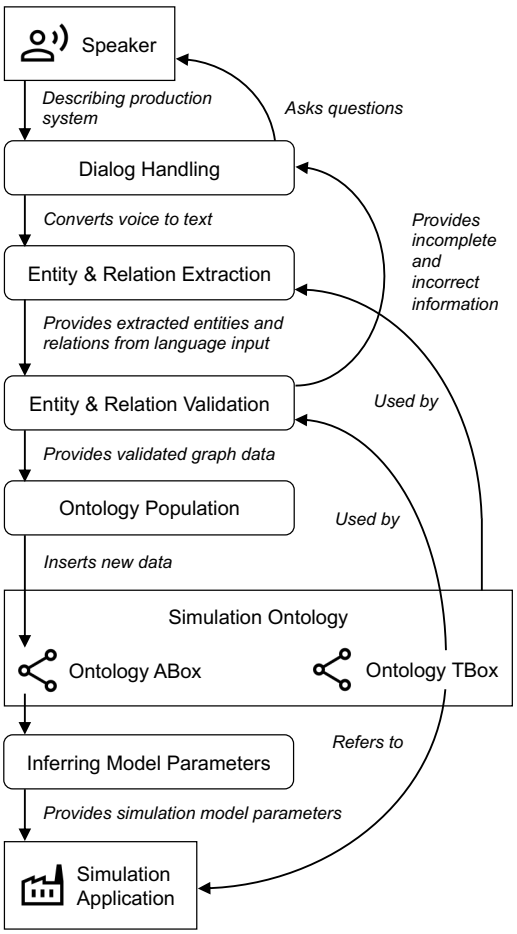

Figure 2: Structure of the system

in the extraction step, but are not explicitly assigned by the LLM.

Entity And Relation Validation

The third step checks the extracted entities and relations for completeness and plausibility. For this purpose, we again use an ontology reasoner that performs the validation based on constraints previously defined in the TBox. If contradictions or missing information are detected, the dialog module is notified, which algorithmically formulates queries and sends them to the employee. For example, the ontology specifies that each machine group must consist of at least one machine by using a cardinality restriction. If the employee declares a new machine group to be simulated but forgets to enter the specific machine, the reasoner recognizes the invalid restriction and the system specifically asks for the missing information to be added.

Ontology Population

The final processing step inserts the complete, valid entities and relations into the ontology ABox ("population"). This includes the detection and correct filtering of duplicated data.

Inferring Model Parameters

Based on the knowledge stored in the ontology, this step extracts relevant entities and relations and map them to the expected input data of the simulation application. This can include the modeled systems configuration, its state and its order backlog as provided by the speech input. For our evaluation, we set up a JSSP model using the *Python*-based discrete-evemt-simulation framework *salabim* (van der Ham, 2018) as an example of a simulation software.

EVALUATION

The use of LLMs plays a decisive role in the functionality of the overall system. However, especially in an industrial context, organizational and technical framework conditions must be taken into account, which may exclude the use of certain models. We have selected 7 different LLMs: *openAI's ChatGPT4-turbo* and *ChatGPT3.5* as well as *Alphabet's Gemini Pro* are LLMs with a very large number of parameters and are only available via the providers' cloud systems. As the description of the production system and it's production processes involves sensitive company data, compatibility with the company's data protection regulations must be considered. ChatGPT4-turbo currently leads the *MT-Bench* score and the *MMLU* score and is therefore considered the most powerful LLM available (as of january 2024). *Llama-2 70b* from *Meta* and *Mixtral-8x7b* from *Mistral AI* are freely available. However, due to their size, they require powerful workstations to operate, which may result in additional initial acquisition and operating costs. We decided to use CodeLlama 34B instruct as a model tuned for coding tasks in order to evaluate the performance of OWL modeling. *Zephyr-beta* as fine tuned version of *Mistral AI's Mistral 7B* is the smallest tested model and is able to run on common personal computer systems.

We evaluate two simulation scenarios S1 and S2 with different sizes and compare the performance of each model measuring correctly extracted and assigned OWL entities and relations by calculating *Precision* and *Recall*. To minimize the randomness of the output and thus obtain repeatable results, we set the parameter *temperature* and the parameter *Top-p-value* to 0 for all models. The prompt input initially consists of the instruction to extract individuals, data property assertions and object property assertions based on the (previously spoken) text in the context of a production simulation. Element labels should be formulated in the singular. All existing classes, individuals, data properties and object properties from the simulation ontology are specified, with the note that not all of them have to be included. All experiments are conducted in English.

S1 describes a production system with a lathe and a drilling machine, with three jobs with respective duedates, to which 4 tasks with respective durations are assigned. This results in 9 individuals, 7 data property assertions and 8 object property assertions. S2 contains two machine parks, one with 2 turning machines and one with 2 drilling machines. There are 5 orders with respective due dates, which include 7 tasks with respective durations. This leads to 18 individuals, 11 data property assertions and 18 object property assertions that need to be extracted. The full speech input in S2 reads as follows:

> "Our production system consists of two machine groups, the drilling machines and the milling machines. There are two milling machines and two drilling machines. There are five production orders consisting of production tasks. The "series drilling" order must be drilled for half an hour. The order for the Hansens company must be milled for 2 hours. The "cuboid" order must first be milled for 1 hour and then drilled for another 20 minutes. The order from the housing construction department must be drilled for 10 minutes. The hold-downs must first be milled for half an hour and then drilled for another half an hour. The order for the Hanses company is particularly important, it has to be finished in 4 hours. The orders of cuboids, series drilling and housing construction should leave our production in eight hours. The hold-down devices have until 16 hours."

ChatGPT-4 is the only model that performs flawlessly in both scenarios and shows a human-like ontology modeling performance, choosing meaningful, self-explanatory labels for the elements. Zephyr as the smallest model extracts almost all concepts in a meaningful way and beats significantly larger models.

We evaluate two metrics for individuals. The *Detection* measurement describes the number of correctly recognized individuals, *Assignment* indicates the correct type assignment of the correctly detected individuals. The "implicit" detection defines the identification of individuals based on assigned data property assertions or object property assertions using predefines domain or range definitions. In the case of data property assertions, *Detection* also indicates all correctly recognized relations; the measurement of the respective correctness of the *Subject* and the *Value* relates to correctly recognized data property assertions. Similarly, we apply the same metric to the specification of correctly extracted object property statements, measuring the assignment of the correct *Object* instead of the *Value* according to the OWL specification. We calculate the arithmetic mean to summarize *Precision* ("Prec.") and *Recall* ("Rec.") for each model in both scenarios.

Table 1 shows the different extraction performances of the models for both scenarios. We have listed the respective model size in parameters under the model name. In general, we found no correlation between model size and modeling performance. Most models perform better in the less complex S1. The average precision of the analyzed LLMs for S1 is 0.97, the recall is 0.94.

Table 1: Comparison of text-to-owl generation of different LLMs.

			ChatGPT4-turbo 1.76 T[M]		Gemini Pro 600 B		ChatGPT3.5 175 B		Llama-2 70b 70 B		Mixtral-8x7b 46.7 B[M]		CodeLlama 34B instruct 34 B		Zephyr-beta 7 B	
			Prec.	Rec.	Prec.	Rec.	Prec.	Rec	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
Scenario S1	I	Explicit Detection	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0.56	**1**	**1**
		Implicit Detection	-	-	-	-	-	-	-	-	-	-	**1**	0.44	-	-
		Assignment	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	$R_{D,A}$	Detection	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0.85	**1**	0.71	**1**	**1**	**1**	0.85
		Subject	**1**	**1**	0.43	0.43	0.43	0.43	**1**	**1**	0.6	0.6	**1**	**1**	**1**	**1**
		Value	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	$R_{O,A}$	Detection	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0.88	**1**	0.5	**1**	0.5	**1**	0.88
		Subject	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
		Object	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	Average		**1**	**1**	0.93	0.93	0.93	0.93	**1**	0.97	0.95	0.85	**1**	0.94	**1**	0.97
Scenario S2	I	Explicit Detection	**1**	**1**	**1**	0.39	**1**	0.39	0.88	0.39	**1**	0.61	**1**	0.39	0.5	0.5
		Implicit Detection	-	-	**1**	0.61	**1**	0.61	-	-	**1**	0.39	**1**	0.22	-	-
		Assignment	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	$R_{D,A}$	Detection	**1**	**1**	**1**	**1**	**1**	**1**	0.83	0.45	0.91	0.91	0.83	0.45	**1**	0.36
		Subject	**1**	**1**	**1**	0.82	0.45	0.45	0	0	0.5	0.5	**1**	**1**	**1**	**1**
		Value	**1**	**1**	0.82	0.78	0.85	**1**	0.8	0.8	0.8	0.8	**1**	**1**	0.25	0.25
	$R_{O,A}$	Detection	**1**	**1**	**1**	**1**	**1**	**1**	0.29	0.11	**1**	0.94	**1**	0.61	0.27	0.17
		Subject	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
		Object	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0.36	0.36	**1**	**1**
	Average		**1**	**1**	0.98	0.95	0.91	0.93	0.72	0.59	0.9	0.89	0.9	0.75	0.75	0.66

[M]Mixed expert model: not all parameters will be used for processing a prompt.

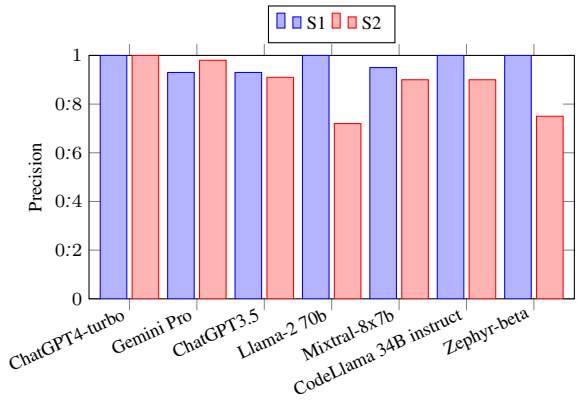

Figure 3: Precision score of the LLMs for S1 and S2

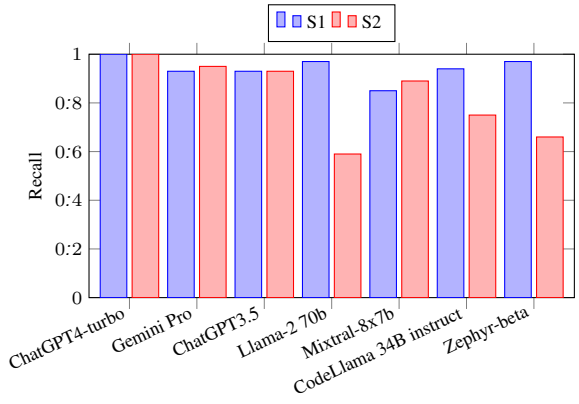

Figure 4: Recall score of the LLMs for S1 and S2

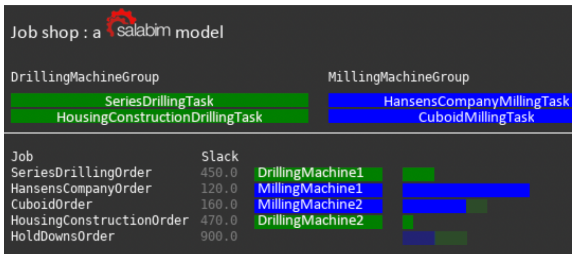

Figure 5: Screenshot of the simulation model instantiated by description S2 using the *salabim* framework

In the more comprehensive S2, CodeLlama 34B instruct shows that fine-tuning with code brings significant advantages in OWL modeling, so that it performs better than the Llama-2 70b, which shares the same model architecture but is twice as large. Behind Llama-2 70b, Zephyr performs worst, hallucinating several additional entities and relations and choosing abstract descriptions for the elements (e.g. it uses “job1” instead of “series milling job”). Another problem is the high precision in finding the subjects in combination with the low precision of the (apparently) extracted values of the data property assertions. In this way, the model extracts relationships that conform to the OWL TBox, but are incorrect in terms of content, which is not noticeable during the algorithmic reasoning process. Using ChatGPT4-turbo’s output, we are able to successfully instantiate a simulation model based on textual input (see fig. 5)

CONCLUSION

In this paper, we combine findings from the research areas of "Ontology-based Production Simulation" and "Ontology Learning" and present a flexible digital assistant for the simple construction of production simulation models based on natural language input. The evaluation focuses on the extraction step, in which the ontological data structure is generated from the spoken text. For this purpose, we compare the performance of different LLMs using two production scenarios. ChatGPT4-turbo consistently shows error-free modeling. For less extensive descriptions, significantly smaller, freely available models also deliver good results (cf. Zephyr-beta, CodeLlama 34B instruct in S1).

The generic, ontology-based system architecture makes it possible to use it for other types of production simulations. The integration of additional data sources is a further object of investigation, so that the ontology uses existing (company) data to record information that is to difficult by using voice input (e.g. the shopfloor layout). The use of smaller LLMs is desirable as they are freely available and allow local execution due to lower resource consumption. However, as performance decreases with more extensive scenarios, large descriptions require a split into smaller parts and the extracted ontological elements to be subsequently merged. Furthermore, CodeLlama 34B instruct shows that fine-tuning on code tasks has a positive effect on the understanding of OWL modeling. The fine-tuning of LLMs specifically for OWL modeling tasks is promising and represents a research gap.

References

K. Agalianos, S.T. Ponis, E. Aretoulaki, G. Plakas, and O. Efthymiou. Discrete event simulation and digital twins: Review and challenges for logistics. *Procedia Manufacturing*, 51:1636–1641, 2020. doi:10.1016/j.promfg.2020.10.228. 30th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2021).

Jerry Banks, editor. *Discrete-Event System Simulation*. Pearson Education, Upper Saddle River, N.J. ; London, 5th ed., international version edition, 2010.

Patrizio Bellan, Mauro Dragoni, and Chiara Ghidini. Assisted process knowledge graph building using pre-trained language models. In Agostino Dovier, Angelo Montanari, and Andrea Orlandini, editors, *AIxIA 2022 – Advances in Artificial Intelligence*, pages 60–74, Cham, 2023. Springer International Publishing.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

Candice Destouet, Houda Tlahig, Belgacem Bettayeb, and Bélahcène Mazari. Flexible job shop scheduling problem under industry 5.0: A survey on human reintegration, environmental consideration and resilience improvement. *Journal of Manufacturing Systems*, 67:155–173, 2023. doi:10.1016/j.jmsy.2023.01.004.

Pere-Lluís Huguet Cabot and Roberto Navigli. REBEL: Relation extraction by end-to-end language generation. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi:10.18653/v1/2021.findings-emnlp.204.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024.

Ahlem Chérifa Khadir, Hassina Aliane, and Ahmed Guessoum. Ontology learning: Grand tour and challenges. *Computer Science Review*, 39:100339, 2021. doi:10.1016/j.cosrev.2020.100339.

Heiner Ludwig, Thorsten Schmidt, and Mathias Kühn. Voice user interfaces in manufacturing logistics: a literature review. *International Journal of Speech Technology*, 26 (3):627–639, September 2023. doi:10.1007/s10772-023-10036-x.

Marvin Carl May, Lars Kiefer, Andreas Kuhnle, and Gisela Lanza. Ontology-based production simulation with ontologysim. *Applied Sciences*, 12(3), 2022. doi:10.3390/app12031608.

MozDevNet. Speechsynthesis - web apis: Mdn, 2023. URL `https://developer.mozilla.org/en-US/docs/Web/API/SpeechSynthesis`.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, page 1–20, 2024. doi:10.1109/tkde.2024.3352100.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 28492–28518. PMLR, 23–29 Jul 2023.

Ebrahim Amini Sharifi Salman Saeidlou, Mozafar Saadat and Guiovanni D. Jules. Agent-based distributed manufacturing scheduling: an ontological approach. *Cogent Engineering*, 6(1):1565630, 2019. doi:10.1080/23311916.2019.1565630.

Julio C. Serrano-Ruiz, Josefa Mula, and Raúl Poler. Toward smart manufacturing scheduling from an ontological approach of job-shop uncertainty sources. *IFAC-PapersOnLine*, 55(2):150–155, 2022. doi:10.1016/j.ifacol.2022.04.185. 14th IFAC Workshop on Intelligent Manufacturing Systems IMS 2022.