

Integrating LLMs with Knowledge Graphs-enhanced Task-Oriented Dialogue Systems

Vasile-Ionut-Remus Iga^{1,*}

¹ Babes-Bolyai University, Business Informatics Research Center, str. Th Mihali, nr. 58-60, Cluj-Napoca, Romania

Abstract

Large Language Models (LLM) have become the state-of-the-art natural language processing systems. Their emergent abilities paved the way for dialogue systems capable of understanding and solving users' specific tasks, ranging from arithmetic problems to simple chatting, all expressed in natural language. However, for specific domains, research has shown that LLMs cannot directly substitute Task-Oriented Dialogue Systems (TOD). TOD Systems aims to master a specific domain or company, enabling communication by natural language. Thus, this research project focuses on building personalized TODS with the help of artificial intelligence, using LLMs grounded with Temporal Knowledge Graphs. We assess the temporal validity of facts in the KG through temporal timestamps. To capture the dynamics of a company or domain, business processes are modeled with BPMN, offering the possibility of converting them to KGs. Finally, the TOD System will be able to grow a domain-specific KG and reason over it, leveraging LLMs capabilities of solving KG-related tasks.

Keywords

Large Language Model, Knowledge Graph, Task-Oriented Dialogue System, Business Process Modeling.

1. Introduction

Nowadays, dialogue systems aim to possess the ability to converse freely using natural language. Scientific research implemented it in two main ways: task-oriented dialogue systems and chatbots. A task-oriented dialog system (TOD) aims to assist the user in fulfilling certain tasks in a specific domain, such as restaurant booking, weather query, and flight booking, which makes it valuable for real-world business [1]. Chatbots are mainly for entertainment, mimicking chit-chatting, and making dialogues more natural.

TODs could be built in various ways, ranging from classical rule-based approaches to neural network ones. The latter supposes either creating a processing pipeline by connecting four dialogue processing tasks, namely natural language understanding, dialogue state tracking, dialogue policy, and natural language generation, or a single end-to-end model that can solve all the above-mentioned tasks at once, this being the state-of-the-art (SOTA) approach [1].

A certain use case for TOD systems is solving diverse tasks of a specific company, by leveraging internal features, while mapping its knowledge. In our prior research [2], we created

CAiSE 2024 Doctoral Consortium

* Corresponding author.

✉ vasile.iga@ubbcluj.ro (V.I.R. Iga)

ORCID 0009-0001-4568-929X (V.I.R. Iga)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

a TODs, leveraging an ontology and a static Knowledge Graph (KG) to contextualize conversations and store pertinent information. This advancement offers significant benefits, including the ability to have multiple discussion threads within the same conversation and the KG serving as a proxy for data validation. The system ultimately aids the company in constructing and managing its specific knowledge base, facilitating Create-Retrieve-Update-Delete (CRUD) operations. The maintenance of the KG is done through our TOD system by solving the Knowledge Graph Completion (KGC) task, and the Knowledge Graph Reasoning (KGR) task to address CRUD operations.

However, our system relied on text template-matching rules, limiting the natural flow of dialogues and the adaptation to new concepts beyond the provided ontology. Consequently, in a subsequent study [3], we employed neural networks to discern user intent and identify relevant entities from input text. While promising, this approach did not completely overcome the previously mentioned drawbacks. Another constraint is the use of static KGs, which cannot capture the time-related validity of facts, unlike temporal KGs.

Regardless of their architecture type, these systems are usually limited by their coverage of the domain's knowledge. For example, a neural network-based TOD is bounded by its training data, leaving a gap for newer use cases. Moreover, the quality and size of the used datasets heavily impact its performance. Therefore, a solution is the synergy between two important and appealing technologies, Large Language Models (LLMs) and Temporal Knowledge Graphs (KGs), assisted by the versatility of Business Process Model Notation (BPMN).

LLMs took the world by storm with their emergent capabilities of solving a wide range of tasks, from mathematical to text-processing ones, formulated in natural language (i.e. prompts). At their core, they are neural networks-based complex architectures with billions of parameters, trained with terabytes of data on computationally expensive supercomputers, that try to model the generative likelihood of word sequences, to predict the probabilities of the next tokens [4]. ChatGPT[†] is the best-known one, with others being Llama[‡], Grok[§], etc. LLMs cannot be used instead of TODs on their own [5], due to their limited domain-specific knowledge, and tendency to *hallucinate* facts [6]. Therefore, aligning them with KGs may alleviate these shortcomings.

Knowledge Graphs (KG) can be defined as graphs of data intended to accumulate and convey knowledge of the real world, whose nodes represent entities of interest and edges represent potentially different relations between these entities [7]. Their capacity to store the time-related context of the facts divides them into Static and Temporal KGs. Meanwhile, they are difficult to construct and evolving by nature [8], thus an LLM-enhanced TODS can reduce these drawbacks by construction through processing natural language text and leveraging pre-trained knowledge.

Building upon our previous work, while leveraging emergent technologies synergized with well-established ones, we formulate the following research objective:

Does the integration of LLMs, Temporal KGs and auxiliary tools such as BPMN *mitigate the shortcomings of limited-knowledge and robotic-like TOD systems?*

To answer it, we propose a system that will be able to (but not limited to) (i) grow a specific company or domain's knowledge graph, and (ii) execute certain related tasks in a traceable

[†] <https://openai.com/blog/chatgpt>

[‡] <https://llama.meta.com/>

[§] <https://x.ai/model-card/>

manner, both while engaging in natural language dialogues. To the best of our knowledge, we are among the first to study the aforementioned research objective using the specified technologies.

The rest of the proposed research article evolves as follows: Section 2 presents the related work, section 3 provides an in-depth analysis of our previous work from where we start, and further details about the desired system’s architecture, while Section 4 concludes the proposal.

2. Related Work

The possibility of grounding TOD systems with knowledge graphs is a well-studied domain. By mapping the dialogue history into graphs to capture its semantics, Yang et al. [9] further exploits the connections between entities in the KG and the mapped graph to enhance reasoning. Tuan et al. [10] proposes an end-to-end model-agnostic method that does symbolic reasoning on any scale of KGs that can be incorporated into dialogue systems to enhance response generation. A somehow similar idea is presented in [11], where a smaller, task-relevant KG is extracted and fed to a BERT model, to guide the answers with the elements from the KG, in an end-to-end manner. Andreas et al. [12] maps each user utterance into a local graph of actions, that is further evaluated and executed similarly to a program, to decide for the next action to take. Our previous work [2] builds on the aforementioned research. However, none of them take into consideration the use of LLMs.

Both LLMs and TODs share the same underlying trait, being conversational agents. Current research aims at enhancing a TOD with an LLM, rather than substituting it. For example, Hu et al. [5] uses ChatGPT to offer user-simulated satisfaction feedback of the TOD output, to further optimize it. Chiu et al. [13] transforms the user’s input text into executable code, using an LLM. Next, the code is executed, while each result updates the dialogue state, based on which an action is selected. Other works shift to using LLMs for TOD-related tasks, such as Dialogue State Tracking (DST) or Natural Language Generation (NLG). For example, Gao et al. [14] designs an adaptive prompt generation framework, to create DST- or NLG-specific prompts for a black-box LLM. Somehow similar to our proposed research, Braunschweiler et al. [15] and Shen et al. [16] try to ground LLMs response, using external knowledge, leading to a more specialized and factual-based TOD. Rather differently, none of them use KGs to store the additional information.

Although the interconnection between LLMs and TODS was not studied too much through the lenses of KGs, large language models can solve KG-related tasks, such as KGC or KGR. Pan et al. [8] proposes a unified roadmap that focuses on the synergy between LLMs and KGs. Zhu et al. [17] experimented with GPT4 and ChatGPT for KGC and KGR, concluding that they are below SOTA models for KGC in Zero- and One-Shot training paradigms, while for KGR they can achieve close or above performances. Han et al. [18] trains a smaller Pre-Trained Language Model (PLM) to check and complete the output of an LLM, for the task of KGC, in an iterative manner. Wei et al. [19] guides ChatGPT to extract relevant information from the input text in a multi-stage dialogue approach, given a certain schema. Despite being relevant work, none of the mentioned studies tackle the tasks regarding Temporal KGs. The use of LLMs for TKG is rather limited, while early experiments suggest that ChatGPT performs poorly, having trouble keeping consistency during temporal inference and failing in actively long-dependency temporal inference [20].

In our work, we plan on using auxiliary tools to help the system become more interpretable, thus one such example is modeling business processes through the use of BPMN. The connection between BPMN and LLMs finds itself at an incipient stage, while not many studies are available yet. One notable research work analyses ChatGPT’s capacity to model different phases of a Business Process Modeling lifecycle [21]. It concludes that it is fit to solve preliminary tasks, such as “gathering information” or “process modeling” in as-is scenarios, but yet unprepared for “selecting and applying redesign methods”, “comparing and assessing models”, or “querying and refactoring models”. Through their work, we find the opportunity to further test and adapt LLMs for BPM.

The research investigated above mainly focuses on the interplay between LLMs and KGs/TODS/BPMN, or TODS and KGs, but does not take into account the possibility of combining them all into a system that benefits their advantages, complementing each other weaknesses. Therefore, our work is keen on developing a powerful TOD that can easily communicate via natural language, while building a company or domain’s specific knowledge and leveraging it, in a straightforward, interpretable, and traceable manner.

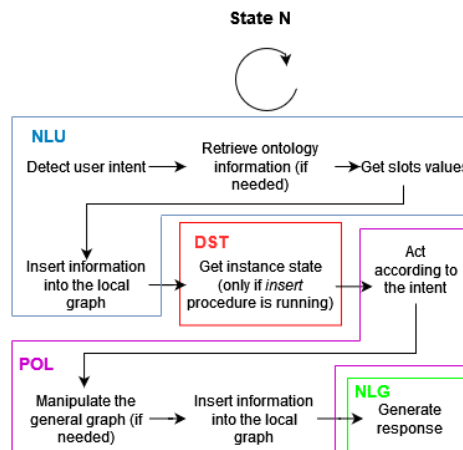
3. Research Methodology

In line with the Design Science approach [25], this section presents the past knowledge that we build upon, the relevance of our work and the design of the proposed system.

3.1. Preliminary Results

The preliminary state-of-the-art in the application domain of our work was extensively presented in the Related Work section, thus here we emphasize our past work relevant to our current research.

In our previous work, we focused on developing a rule-based TOD^{**}, that maps its knowledge on a provided ontology, uses a local KG to map the conversation’s context, and a global KG to store relevant information [2]. Figure 1 presents the overall architecture of the TOD system. It comprises four modules, each focusing on a certain task.



^{**} <https://github.com/IonutIga/TOD-System>

Figure 1: The pipeline architecture of the TOD system [2].

We continued with the integration of neural network components, mainly substituting the rule-based NLU component with a fine-tuned BERT instance^{††} [3]. Its overall architecture was kept simple, adding two classifier layers on top of BERT, for intent and slots detection. The results were satisfactory but not good enough to use in practice.

However, to train such models, one needs adequate datasets, that are not off the shelf. To overcome this problem, we designed a machine-to-machine dialogue simulator^{‡‡} [22]. Figure 2 presents a simplified version of the overall architecture, where the TOD system is the main actor. It starts by requesting a prompt from the generator, based on which it engages in a conversation with a probabilistic rule-based user simulator while annotating each conversation for the specified task (here, NLU).

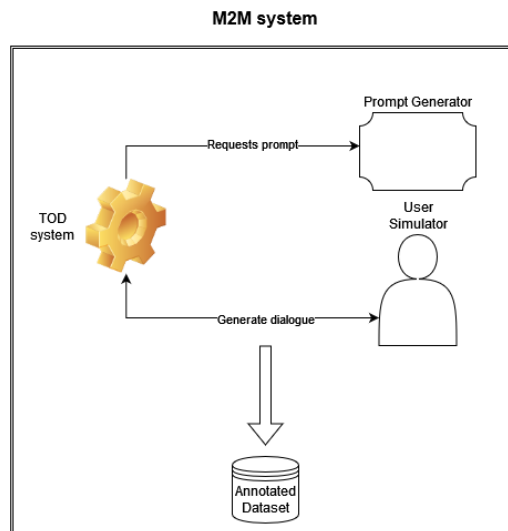


Figure 2: Summarized diagram of the M2M system architecture [22].

All of the above systems were based on a simple ontology, comprising three classes: *Project*, *Employee*, and *Status*, and six relationships either between themselves (*hasManager*, *hasStatus*), or with literal values (*hasName*, *hasClass*, *hasCode*, *hasRole*). Figure 3 depicts the ontology. More details can be found in [22].

^{††} <https://github.com/IonutIga/Domain-Specific-NLU-BERT>

^{‡‡} <https://github.com/IonutIga/Dialogue-Simulator>

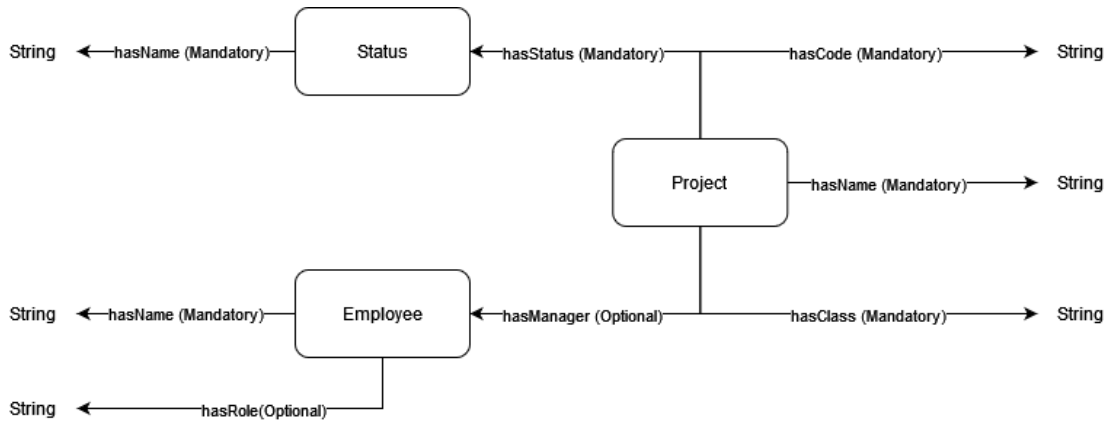


Figure 3: Domain ontology example used throughout experiments.

The system was developed in Python 3.9, the ontology and KGs were described in RDF using the Turtle syntax and managed with the RDFLib library, while the KG queries were written in SPARQL.

As mentioned in the introduction, our approach had several limitations, including the use of template-matching rules for the dialogue flow, or static KGs that suffer from time-validity issues of the stored facts, unlike temporal variants. Thus, we aim to extend the knowledge base (specific to the Rigor Cycle in Design Science) with a new artifact (the proposed system) and all the resulted conclusions after testing the software against automatic and human-centric metrics.

3.2. Application Context

Building a new software must target a specific group of people or organization, such that its development will contribute to the welfare of them. Our main goal is to target the shortcomings of limited-knowledge, robotic-like TOD systems in *any* domain.

However, we are aware that testing such hypothesis is difficult to achieve, thus we narrowed our application context to the usage of such assistants by university students to do their bureaucratic duties (load contract/scholarship forms, choose exam dates, send feedback, request documents etc.), replacing the need to use a variety of platforms or interact with secretaries. From personal and other students' experiences, this tasks can quickly become tedious, having to search through different regulations, call the secretary or discuss with fellow students about how they solved them. These shortcomings should not be part of a modern university, thus they can be alleviated by a central, well-instructed and human-like TOD system, as proposed by our research.

Ultimately, acceptance criteria for the ultimate evaluation of the research results should be defined, able to answer the question: "*Does the design artifact improve the environment and how can this improvement be measured?*". Therefore, the final system will be evaluated **directly** by students, by interacting with it to solve the desired tasks. After a certain interaction, each student will be asked to complete a form regarding the truthfulness, coherency, fluency, understanding, success rate or other relevant metrics. Their responses will be centralized, thus obtaining a global view of the TODS efficiency, and whether we improved the environment or

not. Alongside human metrics, automatic ones will be employed at each stage of the system, to ensure only performant components are added to it. Example of such metrics are: accuracy, recall, precision, f1 score, joint goal accuracy, inform/success rate, hits@k etc. of clearly detecting the intent, choosing the right action, building the correct graph, executing a valid BPMN process, generating a valid response etc.;

3.3. Proposed System

Constructing the artifact is the heart of any design science research project, known as the Design Cycle. The application context leads the development and evaluation of the system, while the preliminary results and related work ground our approach.

Based on our previous findings, culminating with the current research directions, we decided to shift the neural network and rule-based components of the TOD system to LLMs, while keeping the KG-related features with the addition of the time dimension. On top of that, we plan on adding BPMN to trace the execution of the tasks. The expected system should be able to discuss with ease in natural language with the users, understand their needs, extract relevant information and store them in a KG, and provide KG-grounded responses, while executing their specific tasks, in a traceable manner.

LLMs are the main engine of the proposed TOD system. Their task is to execute the four natural language modules presented in Figure 1. As we are aware of their NLG capabilities, the focus shifts to the other three. To achieve them, LLMs should solve the KGC and KGR tasks. Therefore, we aim to start by using prompting techniques to guide LLMs in solving the mentioned tasks. Prompting is the most direct and least demanding manner to obtain desirable results from these models. Different techniques are considered, such as direct prompting (DP), in-context learning (ICL), chain of thought (COT), or planning (P). Pitfalls may be encountered, one of them being the inability of LLMs to perform such tasks by only using prompting techniques. To overcome this, we start by considering fine-tuning the model. Two main techniques are to be used, classical supervised fine-tuning (SFT), and reinforcement learning with human or AI feedback (RLHF/RLAIF). Both of them require training and testing datasets. The first one is straightforward, while the second one requires either a human or another model to guide the reward model into ranking positive answers higher than the negative ones, which is going to be used to fine-tune the LLM. If the LLM is still underperforming, we move to train a smaller specialized PLM that either corrects the LLM or is corrected by it. Finally, one last solution is to consider dividing the main tasks into smaller ones, iterating through all the mentioned approaches. Figure 4 presents the alternative routes for specializing LLMs on KGC and KGR. Each route is depicted with a certain color. We start with prompting (black route), then try fine-tuning (red route), finally testing using specialized PLMs (green route). When a PLM corrects the LLM, prompting happens first. If the LLM corrects the output of the PLM, prompting happens after. The two stages are numbered accordingly, to highlight the possible scenarios. Finally, we can consider the same approach when dividing each of the main tasks into sub ones.

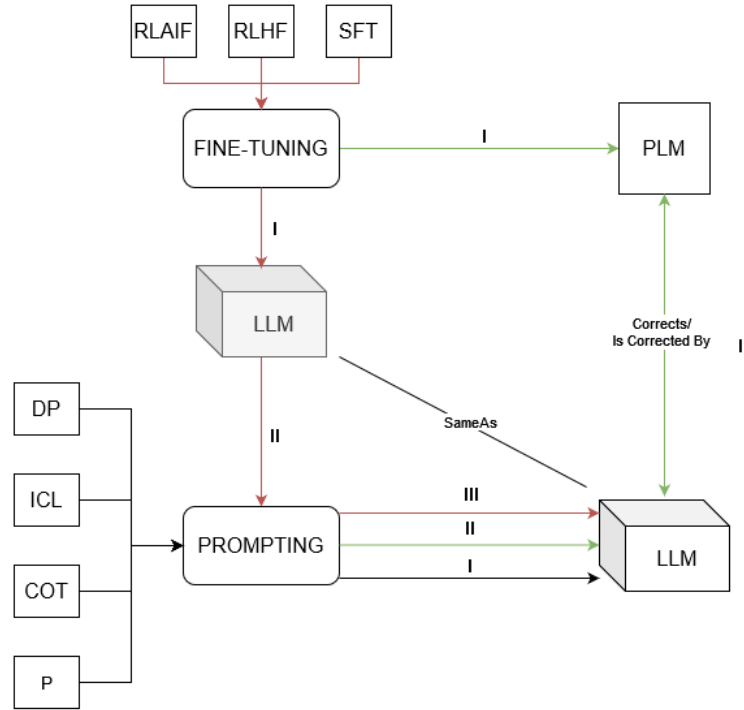


Figure 4: Alternative routes for specializing LLMs on KGC, KGR or sub-tasks of each.

Regarding the temporal dimension of the KG, we plan on starting with a simple, yet powerful method following Nguyen et al. [23], instantiating each relationship, and adding two new temporal relationships to it, namely startDate and endDate. The start date is either today's date (when the input text is encountered) or a provided date. The end date is either a provided date or marked with a label to highlight the fact's continuous validity. It can also be omitted, thus signaling the model that the fact still holds. Regardless of the labeling choice of time, scalability and understanding issues may be encountered. Scalability refers to the number of new relationships added with each new fact, while understanding focuses on the LLMs capability of processing the labels' meaning. If such issues occur with the proposed labeling method, we may consider changing it, following the guidelines of Zhang et al. [24].

Finally, to trace and guide the execution of certain tasks by the TOD system, we emphasize the use of BPMN to model business processes. As BPMN models can be converted into RDF graphs, the connection to our proposed system is evident. Therefore, we plan on designing such models by human experts, that can be later interpreted by the TODS, mainly the LLM component. Thus, we need to ensure LLMs can understand the models. To overcome the possible pitfall, the system will either be directly fed with guidelines about the necessary information to be obtained from the user (by providing human-designed questions to be asked and the target parameters to be collected) or by instructing it to reason over the graph version of the process, where the ontology also describes the nature of encountered facts.

Figure 5 depicts the above-mentioned details of the overall expected TOD. For the BPMN process, we provide both possible solutions that were mentioned above.

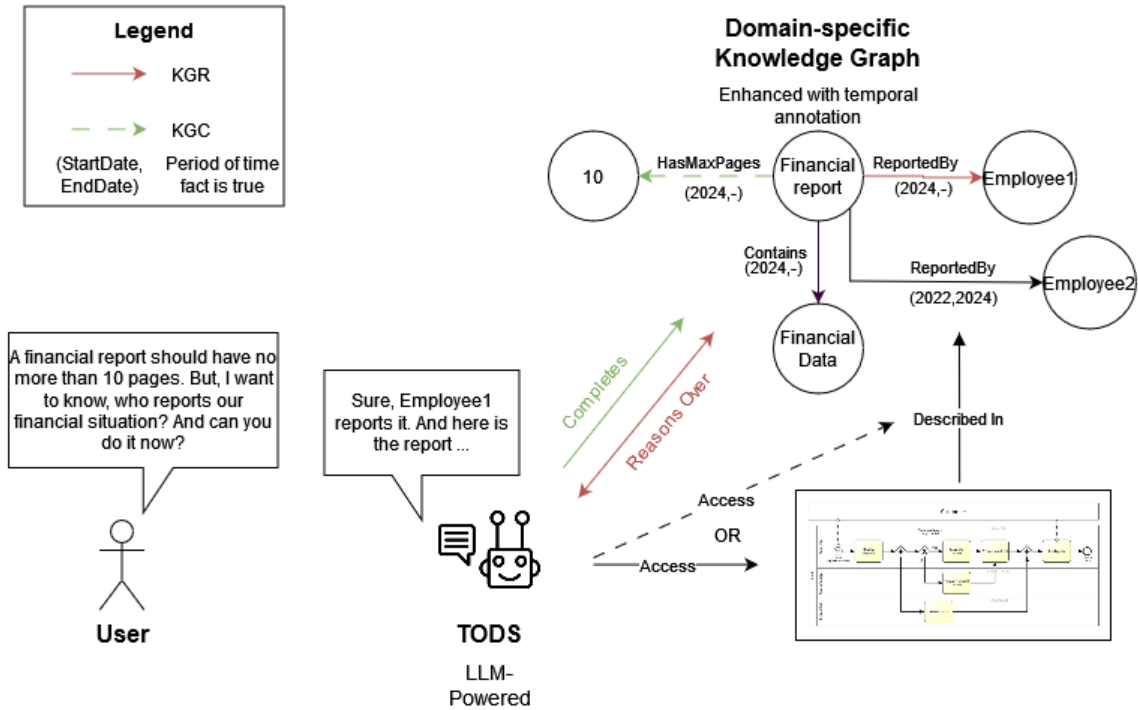


Figure 5: The expected TOD system behaviour.

In line with the application context, the addition of LLMs should increase the naturalness of the dialogue flows and add new information to the system knowledge. Temporal KGs store the university-specific bureaucracy information to ensure the TOD system knows its domain. The temporal dimension assures validity of the processes that are stored inside it. For example, if a task is no longer supported by the university, the system will know to disable its usage. Describing the processes with BPMN increases the interpretability of our system, thus they will only execute what we expect from them. We can view the obtained system as an automatization tool with a natural language interface and self-managing knowledge.

4. Conclusions

Task-Oriented Dialogue Systems focus on solving specific-domain tasks. Such systems can either assist humans in solving their duties or replace repetitive work that can be automated.

However, TODs usually lack naturalness in dialogue flows, while also being reluctant to outside-of-domain information. Therefore, other technologies may be used to overcome these shortcomings. For example, LLMs are fit for conversational paradigms, while also possessing strong background information that can be leveraged for OOD situations. KGs are a great way to store information in a more humanized way, leading to the possibility of reasoning over it. Also, adding the temporal context to it alleviates the problem of time-validity of facts.

To this day, LLMs-KGs, LLMs-TODS, or TODS-KGs synergy has been studied, leading to promising results. However, the connection between all three at once is understudied and should receive more attention.

Building on top of previous work, with respect to the Design Science approach, we focus on developing a TOD system enhanced by LLMs, KGs, and auxiliary tools such as BPMN, to overcome their specific limitations: lack of naturalness and OOD adaptation (TODS), hallucinations, lack of interpretability and domain-specific knowledge (LLMs), and the difficulty of keeping KGs up to date. Each component's advantages can be used to compensate for the other's shortcomings. Our main plan of execution is backed by solutions for each component, reducing the effects of possible pitfalls. Therefore, we aim to obtain a TOD system capable of growing one domain/company knowledge, executing different tasks in a traceable manner, grounded with valid facts existent in a continuously growing specialized Temporal KG.

Acknowledgements

I want to thank Prof. Dr. Gheorghe-Cosmin Silaghi (gheorghe.silaghi@ubbcluj.ro) from Babes-Bolyai University for the supervision of our previous work and the current thesis.

References

- [1] Z. Zhang, T. Ryuichi, Z. Qi., H. MinLie, and Z. XiaoYan, Recent Advances and Challenges, Task-Oriented Dialog Systems, *Science China Technological Sciences* 63 (10): 2011-2027 (2020), <https://doi.org/10.1007/s11432-016-0037-0>
- [2] V.I. Iga, G.C. Silaghi, Ontology-based dialogue system for domain-specific knowledge acquisition, in: da Silva, A.R., da Silva, M.M., Estima, J., Barry, C., Lang, M., Linger, H., Schneider, C. (eds.) *Information Systems Development: Organizational Aspects and Societal Trends (ISD2023 Proceedings)*, Lisboa, Portugal. Association for Information Systems (2023), <https://doi.org/10.62036/ISD.2023.46>
- [3] V.I. Iga, G.C. Silaghi, Leveraging bert for natural language understanding of domain-specific knowledge, in: 25th Intl. Symp. on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2023, Nancy, France. IEEE (2023), to appear
- [4] W.X. Zhao, K. Zhou, et al, A survey of large language models. *arXiv preprint arXiv:2303.18223* (2023), <https://doi.org/10.48550/arXiv.2303.18223>
- [5] Z. Hu, Y. Feng, A.T. Luu, B. Hooi, A. Lipani, Unlocking the Potential of User Feedback: Leveraging Large Language Model as User Simulators to Enhance Dialogue System, in: *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23)*. Association for Computing Machinery, New York, NY, USA, 3953–3957, (2023), <https://doi.org/10.1145/3583780>
- [6] Y. Zhang, Y. Li, L. Cui, D. Cai, L. Liu, T. Fu, X. Huang, E. Zhao, Y. Zhang, Y. Chen, L. Wang, A.T. Luu, W. Bi, F. Shi, S. Shi, Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models. *ArXiv*, [abs/2309.01219](https://arxiv.org/abs/2309.01219) (2023), <https://doi.org/10.48550/arXiv.2309.01219>
- [7] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. de Melo, C. Gutierrez, S. Kirrane, J.E.L. Gayo, R. Navigli, S. Neumaier, A.N. Ngomo, A. Polleres, S.M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, A. Zimmermann, *Knowledge Graphs. Synthesis Lectures on Data, Semantics, and Knowledge*, Morgan & Claypool Publishers, 2021. <https://doi.org/10.2200/S01125ED1V01Y202109DSK022>

- [8] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, X. Wu, Unifying large language models and knowledge graphs: A roadmap, CoRR abs/2306.08302 (2023), <https://doi.org/10.48550/ARXIV.2306.08302>
- [9] S. Yang, R. Zhang, S.M. Erfani, Graphdialog: Integrating graph knowledge into end-to-end task-oriented dialogue systems, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 1878–1888, (2020), <https://doi.org/10.18653/v1/2020.emnlp-main.147>
- [10] Y. Tuan, S. Beygi, M. Fazel-Zarandi, et al., Towards Large-Scale Interpretable Knowledge Graph Reasoning for Dialogue Systems, in: Findings of the Association for Computational Linguistics: ACL 2022, (2022) pp. 383–395, <https://doi.org/10.18653/v1/2022.findings-acl.33>
- [11] D. Chaudhuri, M.R.A.H. Rony, J. Lehmann, Grounding Dialogue Systems via Knowledge Graph Aware Decoding with Pre-trained Transformers, in: Verborgh, R., *et al.* The Semantic Web. ESWC 2021. Lecture Notes in Computer Science, vol 12731. Springer, Cham, (2021), https://doi.org/10.1007/978-3-030-77385-4_19
- [12] J. Andreas, et al., Task-oriented dialogue as a dataflow synthesis, in: Transactions of the Association for Computational Linguistics, vol. 8, pp. 556–571, (2020), https://doi.org/10.1162/tacl_a_00333
- [13] J. Chiu, W. Zhao, D. Chen, S. Vaduguru, A. Rush, D. Fried, Symbolic Planning and Code Generation for Grounded Dialogue, in: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 7426–7436, Singapore. Association for Computational Linguistics, (2023), <https://doi.org/10.18653/v1/2023.emnlp-main.460>
- [14] J. Gao, L. Xiang, H. Wu, H. Zhao, Y. Tong, Z. He, An Adaptive Prompt Generation Framework for Task-oriented Dialogue System, in: Findings of the Association for Computational Linguistics: EMNLP 2023, pages 1078–1089, Singapore. Association for Computational Linguistics, (2023), <https://doi.org/10.18653/v1/2023.findings-emnlp.76>
- [15] N. Braunschweiler, R. Doddipatla, S. Keizer, S. Stoyanchev, Evaluating Large Language Models for Document-grounded Response Generation in Information-Seeking Dialogues, in: Proceedings of the 1st Workshop on Taming Large Language Models: Controllability in the era of Interactive Assistants!, pages 46–55, Prague, Czech Republic. Association for Computational Linguistics, (2023), <https://aclanthology.org/2023.tllm-1.5>
- [16] W. Shen, Y. Gao, C. Huang, F. Wan, X. Quan, W. Bi, Retrieval-Generation Alignment for End-to-End Task-Oriented Dialogue System, in: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 8261–8275, Singapore. Association for Computational Linguistics, (2023), <https://doi.org/10.18653/v1/2023.emnlp-main.514>
- [17] Y. Zhu, X. Wang, J. Chen, S. Qiao, Y. Ou, Y. Yao, S. Deng, H. Chen, N. Zhang, Llms for knowledge graph construction and reasoning: Recent capabilities and future opportunities, CoRR abs/2305.13168 (2023), <https://doi.org/10.48550/ARXIV.2305.13168>
- [18] J. Han, N. Collier, W.L. Buntine, E. Shareghi, PiVe: Prompting with Iterative Verification Improving Graph-based Generative Capability of LLMs, CoRR abs/2305.12392 (2023), <https://doi.org/10.48550/ARXIV.2305.12392>
- [19] X. Wei, X. Cui, N. Cheng, X. Wang, X. Zhang, S. Huang, P. Xie, J. Xu, Y. Chen, M. Zhang, Y. Jiang, W. Han, Zero-shot information extraction via chatting with chatgpt, CoRR abs/2302.10205 (2023), <https://doi.org/10.48550/ARXIV.2302.10205>

- [20] C. Yuan, Q. Xie, S. Ananiadou, Zero-shot Temporal Relation Extraction with ChatGPT, in: The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks, pages 92–102, Toronto, Canada. Association for Computational Linguistics, (2023), <https://doi.org/10.18653/v1/2023.bionlp-1.7>
- [21] N. Klietsova, J.V. Benzin, T. Kampik, J. Mangler, S. Rinderle-Ma, Conversational Process Modelling: Can Generative AI Empower Domain Experts in Creating and Redesigning Process Models?, arXiv preprint arXiv:2304.11065, (2023), <https://doi.org/10.48550/arXiv.2304.11065>
- [22] V.I.R. Iga, Ontology-driven dialogue simulator for generating task-oriented dialogue datasets, Master's thesis, Babes-Bolyai University, Cluj-Napoca, Romania, 2023, <https://github.com/IonutIga/Dialogue-Simulator>
- [23] V. Nguyen, O. Bodenreider, A.P. Sheth, Don't like RDF reification? Making statements about statements using singleton property, in: Chung, C., Broder, A.Z., Shim, K., Suel, T. (eds.) 23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7-11, 2014. pp. 759_770. ACM (2014), <https://doi.org/10.1145/2566486.2567973>
- [24] F. Zhang, Z. Li, D. Peng, et al., RDF for temporal data management – a survey. *Earth Sci Inform* 14, 563–599 (2021), <https://doi.org/10.1007/s12145-021-00574-w>
- [25] A. R. Hevner, A Three Cycle View of Design Science Research, in: *Scandinavian Journal of Information Systems: Vol. 19: Iss. 2, Article 4* (2007), Available at: <https://aisel.aisnet.org/sjis/vol19/iss2/4>