





Towards Bridging the Gap Between Knowledge Graphs and Chatbots

Annemarie Wittig¹, Aleksandr Perevalov^{1,2}, and Andreas Both^{2,3}

¹ Anhalt University of Applied Sciences, Köthen, Germany

² Leipzig University of Applied Sciences, Leipzig, Germany

`andreas.both@htwk-leipzig.de`

³ DATEV eG, Nuremberg, Germany

Abstract. Chatbots are nowadays being applied widely in different life domains. One major reason for this trend is the mature development process that is supported by large companies and sophisticated conversational platforms. However, the required development steps are mostly done manually while transforming existing knowledge bases into interaction configurations, s.t., algorithms integrated into the conversational platforms are enabled to learn the intended interaction patterns. However, already existing domain knowledge may get vanished while transforming a structured knowledge base into a “flat” text representation without references backwards. In this paper, we aim for an automatic process dedicated to generating interaction configurations for a conversational platform (Google Dialogflow) from an existing domain-specific knowledge base. Our ultimate goal is to generate chatbot configurations automatically, s.t., the quality and efficiency are increased.

Keywords: Dialog systems · Chatbots · Knowledge graphs · Synthetic data generation · Natural-language interfaces · Software generator · Human-computer interactions

1 Introduction

Chatbots and other natural-language user interfaces have become a major driver for interactive systems. It is not hard to predict a very important role of such systems for user interaction in the future. The technology for creating chatbots is becoming more powerful and robust (e.g., [1, 7]). Platforms like Amazon Alexa¹, Google Dialogflow², and Microsoft Bot Framework³ as well as powerful open-source frameworks (like Rasa⁴) provide a rich set of features to build (novel) Web-based dialog systems without strong technical skills.

¹ cf., <https://developer.amazon.com/alexa/>.

² cf., <https://cloud.google.com/dialogflow>.

³ cf., <https://dev.botframework.com/>.

⁴ cf., <https://rasa.com/> and <https://github.com/RasaHQ/rasa>.

However, creating a chatbot using one of the well-known conversational platforms might become time-consuming while doing the configuration process manually. This process demands the alignment of a domain-specific knowledge base with the possible user-chatbot interaction patterns (or intents). Typically, this process is done manually and cannot take advantage of pre-existing knowledge. A few research initiatives have recently addressed this automation challenge and are therefore considered as related work. For example, in [8] BPMN models are used as input for a chatbot generator. Another approach uses HTML pages, annotated with specific information, to create a specific chatbot automatically, as described in [3].

To the best of our knowledge, a pre-existing knowledge graph (KG) [6] cannot be directly used for configuring chatbot platforms, although it might already perfectly define the domain knowledge in a machine-readable format. From this observation, we conclude the demand for a process that will enable usage of the domain-specific knowledge bases for creating the configurations, s.t., a chatbot can be generated automatically while preserving the modeled domain knowledge. This approach follows the same goal as Question Answering over KGs [4]: to make structured domain-specific data accessible by natural-language input.

Our long-term goal is to establish a generalized, robust engineering approach to create a chatbot configuration based solely on an existing standardized domain-specific knowledge base. In this paper, we consider a special type of knowledge bases – knowledge graphs (KGs). We hypothesize that from a KG, the training data for interaction patterns of a chatbot (typically: questions and its intents; in this paper, we generally use the term *questions*) can be generated. Typically, for a KG, *natural language verbalizations* of triples as a whole are not available. Therefore, in this paper, we manually established *fragment templates* for such verbalizations. We have done so, by defining templates that can be combined with actual questions. Additionally, replacing and combining abstract concepts (e.g., Employee) and relations (e.g., hasEmail) in the KG leads to usable questions. While doing so, a question fragment such as “What’s the *<hasEmail>* of *<Employee>*?” pointing to the concept **Email** can be transformed to the real question “What’s the *mail* of *[Andreas Both]*?” or “What’s the *email address* of the *employee*, who *teaches* *[Question Answering and Chatbots lecture]*?” etc.

Given this scenario, we derived the following research questions: *Research question 1 (RQ₁)*: “Is it possible to automatically generate a chatbot configuration from a given knowledge graph, s.t., the chatbot answer quality is comparable to a manually generated system?”; *Research question 2 (RQ₂)*: “Is the quality of such an automatic process sufficient for real user interaction?”.

To start the discussion with the scientific community regarding the research questions and to preliminarily validate our approach, we used an exemplary KG describing a department of a university including: timetable of the offered courses, courses (and their instances), appointments, lectures, and the employees of the university considering their general information. For executing the experiments, we use Google Dialogflow as a platform for creating a chatbot from the configuration. Hence, the whole setting can only be influenced by the data provided to the Google Dialogflow, especially since the exact processing of the data

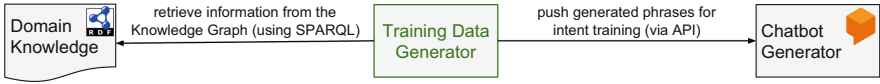


Fig. 1. Big picture of KG-based chatbot generation process

by it is encapsulated. The experimental results show overall good results for the training and testing with the generated data.

Although the approach is not yet generalized, our experimental analyses show great potential. Hence, we propose to the research community the future directions of generating Web-based natural-language user interfaces from domain-specific knowledge bases.

The paper is structured as follows. In the next section, we will describe our approach. Our experiments are described in Sect. 3. In Sect. 4, we will discuss our findings and sketch a future end-to-end process for generating chatbots automatically from a KG. The paper is concluded in Sect. 5.

2 Approach

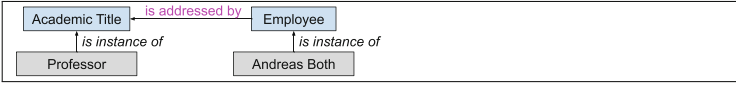
Our process is driven by the domain knowledge represented as a knowledge graph and will create the data required for configuring a chatbot using the Dialogflow platform (cf., Fig. 1). The main idea is driven by the observation that an available information modeling of a domain is already providing well-suited knowledge representation (as it is already done in many companies/industries⁵). Typically, RDF-based knowledge graphs are used for technical implementation (cf., [5]). Hence, it is also used here. Consequently, our *approach is aiming to automatically generate the textual training data (natural-language questions) for a conversational platform from a given knowledge graph*. In the following, we will describe the requirements for the two main tasks of the training data generator. A chatbot is based on the interaction patterns or, more precisely – intents. They are the essential part of a dialogue and are activated depending on the input of a user. On activation, the answer, predefined in the configuration, is provided by the system. The input questions might contain specific parts which are reflecting a particular intention and therefore are used by the underlying intent-detection algorithm to compute the correct response. All this information needs to be provided to the conversational platform.

In this work, the domain-specific knowledge base is represented as an RDF-based⁶ knowledge graph. Hence, the data has a common-sense knowledge (e.g., a lecturer is teaching courses) and the concrete instances are also represented within the KG (e.g., the instance with the label “Andreas Both” is a lecturer, “Andreas Both” is teaching the course “Web Engineering”). Given this information, we assume that for each intent at least one textual representation of a

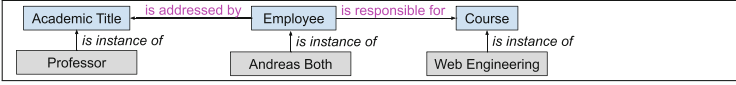
⁵ cf., <https://iirds.org/>, <https://blog.cambridgesemantics.com/merck-kгаа-bosch-and-deloitte-share-their-knowledge-graph-stories>, <http://internationaldataspaces.org>.

⁶ cf., <https://www.w3.org/TR/rdf11-primer/>.

E_1 contains *simple questions*, e.g., “With which **academic title** do I *address* the **employee** **Andreas Both**?” that can be derived from a KG like:



E_2 contains *mid-size questions*, e.g., “With which **academic title** do I *address* the **employee** who *is responsible for* the **course** **Web Engineering**?” that can be derived from a KG like:



E_3 contains *long questions*, e.g., “With which **academic title** do I *address* the **employee** who *is responsible for* the **course** that *is part of* the **study program** **computer science**?” that can be derived from a KG like:

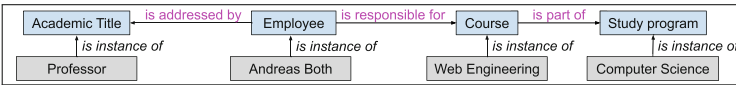


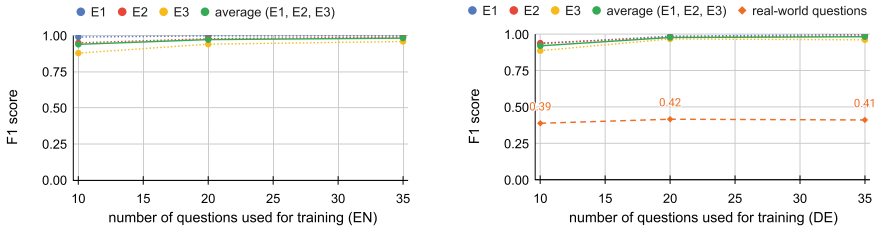
Fig. 2. Knowledge evaluation patterns.

question can be generated. For example, the question “*Who teaches courses?*” can be generated from the described common-sense knowledge, and the question “*Is Andreas Both teaching Web Engineering?*” from the described instance. There are several options available to mix terminology and instances, e.g., “*Who teaches Web Engineering?*”. Obviously, substrings like “Andreas Both” or “Web Engineering” reflect the required parameters of the user’s input and can be used to compute the expected chatbot response. For a completely automated process, we assume that such textual questions are generated automatically from the KG and the instances are highlighted within the questions using framework-specific markup. Figure 2 shows the examples of how training data can be generated. There, E_1 contains verbalizations that are generated using a simple pattern that is based on a predicate. Correspondingly, E_2 uses 2 edges of the given KG and E_3 uses 3 edges to generate verbalizations.

3 Experiment

To validate the approach, an ad hoc ontology of the Anhalt University using the domain knowledge of the authors was used. For the evaluation, 3 experiments (Exp_1 , Exp_2 , and Exp_3) regarding different verbalization types were designed. For each experiment, we create three types of input based on the complexity of the question (E_1 , E_2 , E_3). The complexity depends on how many triples are integrated into the question. Considering the KG, these facts are directly correlating with the KG edges that would be used to compute an answer (cf., Fig. 2).

In all experiments, the Google Dialogflow conversational platform was used. The experiments were performed using the API to ensure uniformity of execution. We trained a model for each verbalization type (Exp_1 , Exp_2 , and Exp_3)



(a) Exp₁^{EN}: Evaluation of English dataset (b) Exp₁^{DE}: Evaluation of German dataset

Fig. 3. Exp₁: Evaluation of label-based generation of verbalizations of training data

and evaluated the quality for E₁ (simple), E₂ (medium), and E₃ (long) questions (and also the average quality) separately for English and German. Additionally, we used a randomly selected subset of *503 real-world German questions*⁷ to evaluate the German model. These questions were collected through an integration of the chatbot into the live learning management system (LMS) of the Anhalt University. The users, Anhalt University’s bachelor students of different years, were provided with a general description of the supported topics and instructed to create related questions that are used as a dataset. However, the underlying ontology was *not* changed relating to the collected real-world input from actual users. All models were evaluated using a 5-fold cross-validation where N randomly selected questions are used for the training with $N \in \{10, 20, 35, 50, 100, 150, 200, 250, 500\}$ and the quality evaluation was concluded by using F1 scores. Hence, the knowledge representation needs to be considered static. In the following, we will evaluate three different training data generations (the data is available in our online appendix) and their quality regarding the real-world questions.

Exp₁: Verbalization Using Only Concept and Entity Labels. In this experiment, we used only the labels of concepts and entities to generate the training data for the chatbot (cf., Fig. 3). A simple verbalization could be “*academic title* *Andreas Both* ?” (cf., the example in Fig. 2). The structure of such data can only roughly be described as a natural language. Due to the usage of only labels, only test sets of 10, 20, and 35 questions per intent were generated and evaluated for E₁, E₂, and E₃ questions (cf., Fig. 3a and 3b). However, even this comparatively low number of training data is sufficient. As Fig. 3 demonstrates, the F1 score is increasing w.r.t. the number of the provided training data. In general, the quality of the chatbot model is acceptable (leading to the assumption that the named entities and concepts are dominating features of the Dialogflow’s intent detection model). Surprisingly, even the evaluation of the real-world questions (cf., Fig. 3b) is reasonable (between 0.39 and 0.42).

Exp₂: Verbalization Using Predefined Patterns. In our second evaluation, we used predefined templates to simulate the creation of natural-language questions. They use all defined labels (cf., Fig. 2), vastly increasing the number

⁷ The data is available in our online appendix at <https://doi.org/hmb3>.

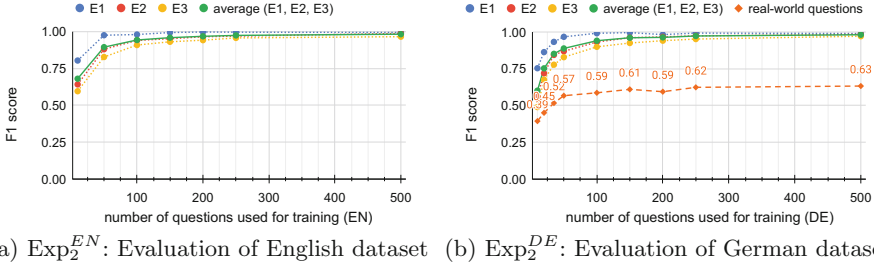


Fig. 4. Exp_2 : Evaluation of sentence-based generation of verbalizations of training data

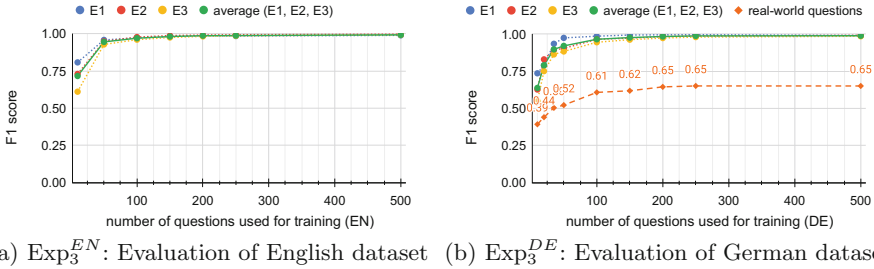
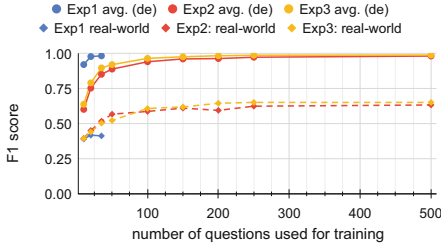


Fig. 5. Exp_3 : Evaluation of clause-based generation of verbalizations of training data

of questions generated to up to 500. The generated E_1 verbalizations can be considered to reflect well-formed natural language (e.g., “With which academic title *address* the employee Andreas Both?”). For E_2 and E_3 questions, we simply replaced the addressed entity or concept with a question that is pointing to it. For example, the entity “Andreas Both”, contained in the previously mentioned E_1 question, could be addressed using questions aiming for an answer of type Employee, e.g., “Who *is responsible for* Web Engineering?”. Combining both questions results in a E_2 question such as “With which academic title *address* the employee *Who is responsible for* Web Engineering?”. Obviously, the natural-language quality of the mid-size (E_2) and long (E_3) questions will not always be high. Nevertheless, the Exp_2 ’s evaluation quality is increased in comparison to Exp_1 (cf., Fig. 4). In particular, the generated German model shows improved quality regarding the real-world questions (cf., Fig. 4b).

Exp₃: Verbalization Using Subordinate Clause. The final evaluation was done with additional templates. The generation mechanism is the same as in Exp_2 . However, the templates were improved, s.t., the combinability of templates is increased. As the simple questions (E_1) are not created by combining question templates, they are equal to the ones of Exp_2 . However, we intentionally created additional templates to extend questions with subordinate clauses. They lead to more natural sentences, e.g., “With which academic degree *do I* *address* employee, *who is responsible for* Web Engineering?”. The results of the evaluation are shown in Fig. 5. It shows a very similar model quality as Exp_2



Evaluation	Correlation
Exp ₁	0.971
Exp ₂	0.984
Exp ₃	0.936

(a) Comparison of trained German models' quality vs. real-world questions.

(b) Correlation of German model quality and German real-world questions

Fig. 6. Comparison of results.

for both languages. However, the evaluation of the German real-world questions shows a significant improvement in comparison to Exp₂, which is also achieved with smaller training sets (with 200 questions Exp₂: 0.59 vs. Exp₃: 0.65). Hence, we can assume that a better natural-language representation of the generated training data is leading to an improved chatbot quality.

4 Discussion

Our experiments show that the approach for automatic training data generation along with the Google Dialogflow intent detection module demonstrates decent results. Despite the approach having significant limitations, as only pre-defined templates were used in the experiments, the Dialogflow's models were still capable of providing reasonable quality, as summarized in Fig. 6a.

Even while using such an unideal process, we are capable to highlight the potential advantages of our approach by the conducted experiments: (1) An automatic process is capable of generating more training data than a manual process, which might improve the quality towards a very high level; (2) Our approach is also able to create multilingual conversational interfaces, leading to higher chatbot generation efficiency and better maintainability of web applications, as they are often built for multilingual environments.

Given our results, the automatic generation of chatbots is possible (i.e., our research question RQ_1 is answered). The obvious advantage is complete coverage of the modeled knowledge domain in the training data for the intents of the chatbot. In addition, our approach enables the efficient provision of significantly larger training data than a human chatbot maintainer would like to generate manually. The correlations of the real-world questions and the average model quality is also very high (cf., Fig. 6b). Hence, the RQ_2 is answered too.

We identified the automatic training data generation as a crucial but missing component for actually achieving the end-to-end automation for creating chatbots based on a given KG. Consequently, our research results point to the fact that scientific investment into establishing robust methods to automatically generate natural-language questions from a KG is required (cf., [2,9]). Hence, we would propose to the research community to develop such a component.

5 Conclusions and Future Work

In this paper, we proposed an end-to-end process for automatically configuring a chatbot by generating training data. The proposed process is based on a domain-specific knowledge base, represented by a knowledge graph, which is a common approach for representing semantic data and is providing terminology (concepts and predicates) as well as concrete data instances. The process was implemented and evaluated while simulating the intent detection task. The experimental results show that it is possible to achieve reasonable quality for real-world questions. Nevertheless, fine-tuning the results and iterative extension of the verbalization templates is required.

However, such an automated process might have a very positive impact on the time and costs (i.e., *efficiency*) for establishing chatbots. Additionally, indicated by our experiments, we assume that *higher quality* can be achieved as more training data can be generated automatically with much higher efficiency in comparison to a manual process. This would foster the generation of future NL-driven Web applications, as the domain-specific knowledge model is typically available (because it is also used for other applications).

References

1. Abdellatif, A., Badran, K., Costa, D., Shihab, E.: A comparison of natural language understanding platforms for chatbots in software engineering. *IEEE Transactions on Software Engineering* (2021)
2. Bouayad-Agha, N., Casamayor, G., Wanner, L.: Natural language generation in the context of the semantic web. *Semantic Web* **5**, 493–513 (2014)
3. Chittò, P., Baez, M., Daniel, F., Benatallah, B.: Automatic generation of chatbots for conversational web browsing. In: Dobbie, G., Frank, U., Kappel, G., Liddle, S.W., Mayr, H.C. (eds.) *ER 2020*. LNCS, vol. 12400, pp. 239–249. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-62522-1_17
4. Diefenbach, D., Both, A., Singh, K.D., Maret, P.: Towards a question answering system over the semantic web. *Semantic Web* **11**, 421–439 (2020)
5. Galkin, M., Auer, S., Scerri, S.: Enterprise knowledge graphs: a backbone of linked enterprise data. In: *2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pp. 497–502. IEEE (2016)
6. Hogan, A., et al.: Knowledge graphs. *Synthesis Lectures on Data. Semant. Knowl.* **12**(2), 1–257 (2021)
7. Janarthanam, S.: Hands-on chatbots and conversational UI development: build chatbots and voice user interfaces with Chatfuel, Dialogflow, Twilio, and Alexa Skills. Packt Publishing Ltd., Microsoft Bot Framework (2017)
8. López, A., Sánchez-Ferreres, J., Carmona, J., Padró, L.: From process models to chatbots. In: Giorgini, P., Weber, B. (eds.) *CAiSE 2019*. LNCS, vol. 11483, pp. 383–398. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21290-2_24
9. Seyler, D., Yahya, M., Berberich, K.: Generating quiz questions from knowledge graphs. In: *Proceedings of the 24th International Conference on World Wide Web* (2015)