A Novel Multi-agent-based Chatbot Approach to Orchestrate Conversational Assistants

Jan Felix Zolitschka(✉)

University Ulm, Helmholtzstraße 22, 89081 Ulm, Germany
jan.zolitschka@uni-ulm.de

Abstract. Nowadays, chatbots have become more and more prominent in various domains. Nevertheless, designing a versatile chatbot, giving reasonable answers, is a challenging task. Thereby, the major drawback of most chatbots is their limited scope. Multi-agent-based systems offer approaches to solve problems in a cooperative manner following the “divide and conquer” paradigm. Consequently, it seems promising to design a multi-agent-based chatbot approach scaling beyond the scope of a single application context. To address this research gap, we propose a novel approach orchestrating well-established conversational assistants. We demonstrate and evaluate our approach using six chatbots, providing higher quality than competing artifacts.

Keywords: Conversational agent · Chatbot · Multi-agent-based system · Orchestration · Collaboration · Mediation · Divide and conquer

1 Introduction

In recent years, conversational agents, also called chatbots, are becoming an increasingly pervasive means of conceptualizing a diverse range of applications in various domains [1–4]. Furthermore, it is forecast that due to the usage of chatbots the annual cost savings in organizations will grow from $48.3 million in 2018 up to $11.5 billion until 2023 [5]. In the area of conversational artificial intelligence, many studies already tap the potential of chatbots and provide well-known approaches for chatbots interacting with humans as well as answering questions regarding open-domain (non-task-oriented) and closed-domain (task-oriented) topics [1, 2, 6–8]. Following a survey from Ramesh et al. [8], chatbot approaches have evolved from simple pattern matching (e.g. ALICE [9]) to modern complex knowledge- and retrieval-based approaches (e.g. MILABOT [10], ALQUIST [11], EVORUS [12] or ALANA [13]) with the aim of giving conversations more human-like shape in order to pass the Turing test. Nevertheless, designing a versatile chatbot, giving reasonable answers to a variety of possible requests, is a challenging task. In particular, the task of simultaneously being robust regarding various domains and answering domain-specific questions is extremely demanding [6]. As a consequence, the major drawback of most chatbots is their limited scope [14]. In order to increase the capabilities of a single chatbot, most approaches in literature manually build and add skills to existing chatbots [15, 16]. Although this works reasonably well, it leads to a

W. Abramowicz and G. Klein (Eds.): BIS 2020, LNBIP 389, pp. 103–117, 2020.
https://doi.org/10.1007/978-3-030-53337-3_8

high effort in generating and coordinating various and complementary skills [14]. Other approaches based on current state-of-the-art knowledge and retrieval models (e.g. deep neuronal networks [10, 11]), require massive datasets, skilled human resources as well as a huge amount of time and hardware to be trained, enhanced and optimized [10]. Besides, as chatbots have become more prominent, the size and complexity of chatbot systems is increasing, which, as for any software system, cannot increase indefinitely [17]. The area of multi-agent-based systems offers a wide range of approaches to solve problems in a cooperative manner following the "divide and conquer" paradigm [18, 19]. In other domains, multi-agent-based approaches are commonly used to model complex and emergent phenomena inspired by human behavior as, for instance, expert collaborations in organizations interacting with the aim of solving a customer request. Consequently, as existing chatbots already provide sound results regarding the scope of their application, it seems promising to design a multi-agent-based chatbot approach, inspired by expert collaboration in practice, scaling beyond the scope of a single chatbot. However, there is still a lack of chatbot approaches giving reasonable answers to a variety of possible requests. To address this research gap, we orchestrate multiple chatbots and propose a multi-agent-based chatbot approach, which is able to learn chatbots' capabilities and identify relevant chatbots capable of giving answers.

Guided by the Design Science Research (DSR) process due to Peffers et al. [20], the remainder of this paper is structured as follows: In the next section, we provide an overview of the related work and identify the research gap. In Sect. 3, we propose a multi-agent-based chatbot approach relying on capability-based middle-agents to orchestrate chatbots in a single conversational agent. In Sect. 4, we demonstrate and evaluate our approach based on six different chatbot datasets on which the approach could be successfully applied. Finally, we conclude with a brief summary, limitations and an outlook on future research.

2 Related Work and Research Gap

In the area of conversational artificial intelligence, many studies already tap the potential of conversational agents and provide well-known chatbots approaches. Nevertheless, there is still a lack of approaches solving user requests in a cooperative manner by reusing multiple well-established chatbots. In the following, informed by the related literature regarding interactions between humans and multiple conversational agents as well as agent collaboration in multi-agent-based systems, we identify the research gap.

2.1 Interactions Between Humans and Multiple Conversational Agents

There has been recent work on analyzing different kinds of interactions between humans and multiple chatbots. First of all, the communication between a user and multiple chatbots can be conducted by multiple *single-bot chats* (a user interacting with a single chatbot in a single chat, e.g. [9–11]), *multi-bot chats* (a user interacting with multiple chatbots in a shared chat [3, 14, 21]) or a *single bot chat orchestrating hidden chatbots* (a user interacting with a single chatbot in a single chat, which in turn interacts with multiple hidden chatbots [12, 13]). At a first glance, interacting with multiple chatbots in multiple single-bot chats or a shared chat seems to increase complexity. Thus, researchers have investigated the user experience of single- versus multi-bot chats [3, 14, 21]. Chaves and Gerosa [3] conducted a Wizard-of-Oz study, where subjects are deluded into thinking that they are interacting with chatbots. As a result, the participants reported more confusion using a multi-bot chat in comparison to single-bot chats. Besides, they identified no significant difference between conversations in single- and multi-bot chats. This is in line with Pinhanez et al. [14], who point out that there is no increase in collaboration and coordination costs while interacting in a multi-bot chat. In contrast, Maglio et al. [21] investigated interactions in an office setting and found out that participants needed less effort to control multiple hidden chatbots within a single chat compared to conversing with chatbots in an individual or shared chat, respectively. Hence, regarding different kinds of interactions, literature reveals promising potential in approaches orchestrating multiple hidden chatbots [3, 14, 21].

In this context, only a few researchers focus on human collaboration with multiple chatbots in terms of a single chatbot orchestrating hidden chatbots [12, 13, 22]. To do so, Papaioannou et al. [13], one of the top competitors of Amazon's Alexa prize, developed a chatbot named ALANA. Their approach is based on a contextual bot priority list and a ranking function trained on user feedback to choose a response from one out of seven (2017) or rather nine (2018) chatbots. Cui et al. [22] solely base their approach on a static priority list in order to choose responses out of four chatbots without any consideration of the customer's intent. In contrast, a chatbot called EVORUS [12] used not only six different chatbots but also crowd-sourced human workers when their approach was unable to answer. To do so, their approach collects feedback from the crowd and learns to select chatbots, which are most likely to generate high-quality responses depending on the given context [12]. Nevertheless, all mentioned approaches orchestrating hidden chatbots solely rely on static priority lists or require a huge amount of user-provided ratings and feedback, which is often not available, time-consuming or expensive to collect. To the best of our knowledge, besides these few examples, further literature does not focus on orchestrating the capabilities of multiple already existing chatbots. In particular, the collaboration between open- and closed-domain chatbots without the use of feedback is not yet addressed by existing literature.

2.2 Agent Collaboration in Multi-agent-based Systems

In particular, as the size and complexity of most systems cannot increase indefinitely, research in the area of multi-agent-based systems offers a wide range of approaches to address the challenge of jointly acting agents [17–19]. In general, multi-agent-based approaches are used to design complex and emergent phenomena inspired by human behavior by using a collection of autonomous and distributed entities, called agents, with individual decision-making. Each agent is designed as an individual software agent with well-defined and limited scope, which perceives its environment (e.g. prospective user messages) and determines its actions accordingly (e.g. reasonable answers) [18]. Several previous studies have proposed agent theories and architectures to provide multi-agent-based systems with a strong formal basis. Hence, different types of agent architectures can be applied depending on the complexity of the agents' deliberation process. The most widely applied types are reactive agents, which are determined by static action rules [23, 24], in contrast to deliberative agents, which use symbolic reasoning for planning their actions [25]. Furthermore, the predominantly used agent architecture is derived from the physical and internet economy [26] and subdivides agents into requester-agents demanding a service (e.g. a visitor requests an apple on a physical market), provider-agents supplying a service (e.g. an orchardist supplies apples to a physical market) and middle-agents as intermediaries (e.g. a salesman mediates fruits on a physical market) [19, 27]. Obviously, regarding the context of our study, particularly designing the user as requester-agent, existing chatbots as reactive provider-agents and middle-agents as intermediaries between users and chatbots seems promising to cope with the task of orchestrating multiple conversational agents.

2.3 Research Gap

Despite emerging scientific work in the field of chatbots [3, 10, 12–14, 28], we still observe a lack of research on how multiple well-established chatbots could be reused in a jointly coordinated manner to scale beyond the scope of a single chatbot application. Regarding related research in distributed artificial intelligence, the area of multi-agent-based systems already offers a wide range of approaches to solve problems in a cooperative manner [18, 19, 23–25]. However, orchestrating hidden well-established chatbots by means of multi-agent-based technology is a novel approach for the area of conversational agents and not yet investigated by previous literature so far. Merely, Hettige and Karunananda [28] take a first step by modeling the components (e.g. graphical interface, natural language processing or data access) of a chatbot as a multi-agent-based system called OCTOPUS, but do not integrate or rather orchestrate single hidden chatbots. Thus, we assume that investigating chatbots based on multi-agent-based architecture without relying on a huge amount of feedback, harbors enormous potential for research and copes with the current challenges in the context of collaborative chatbots. Indeed, to the best of our knowledge, so far none of the recent studies in conversation agents has considered orchestrating hidden chatbots in a single conversational agent while at the same time taking an integrated perspective by not only ranking chatbot answers based on priority lists or human feedback but rather combining research streams by embedding chatbots as provider-agents into a multi-agent-based architecture. Thus, we aim at designing a novel multi-agent-based chatbot approach combining conversational agents and multi-agent-based methods in a well-founded way which improves the versatility of chatbots giving reasonable answers.

3 Novel Multi-agent-based Chatbot Approach

Having stated the solution's objectives, following the DSR process by Peffers et al. [20], we set out to develop an approach to answer human questions with a single chatbot regarding open-domain and closed-domain topics by orchestrating and coordinating hidden chatbots (cf. Fig. 1). Since our research is concerned with the development of a novel approach, it constitutes a contribution of nascent design theory [29] and represents an example of work in interior mode [30]. In our research, we mainly employ a deductive, iterative knowledge creation strategy and develop our approach based on a series of hypotheses that we test and validate. In its entirety, the design and search process described in the following forms a single iteration of the DSR process [20, 31].

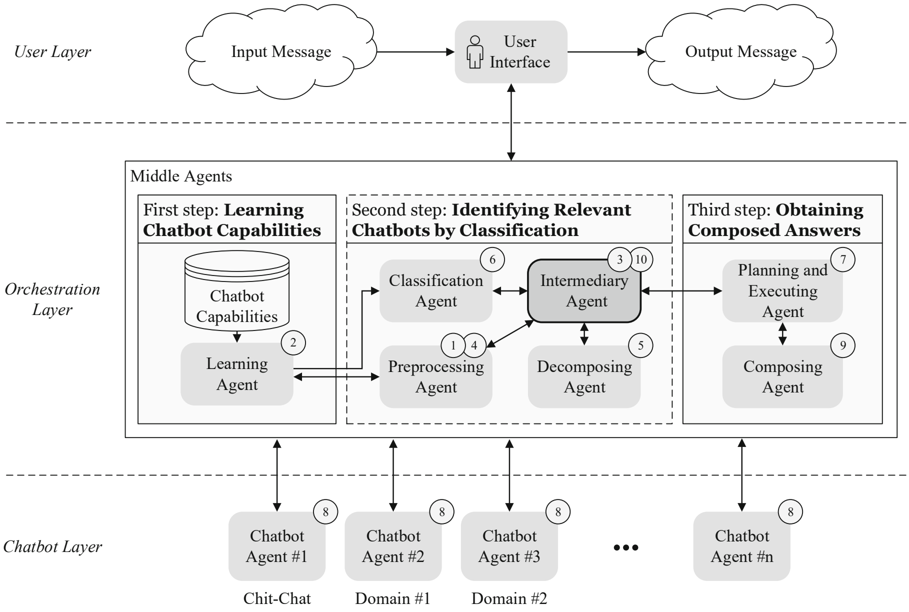

Fig. 1. Multi-agent-based chatbot approach to orchestrate conversational assistants

3.1 Basic Idea and Overview

In literature, multi-agent-based approaches are used to design complex and emergent phenomena inspired by human behavior, such as expert collaboration in organizations interacting with each other to solve a customer request [17–19]. Consequently, we base the architecture of our novel approach on the "divide and conquer" paradigm of multi-agent-based systems containing requester-, provider- and middle-agents as intermediaries [19].

Following the basic process of collaboration through mediation between requester-, provider- and middle-agents in multi-agent-based systems by Klusch and Sycara [19], the response of a user request can be identified by first, advertising the capabilities of all provider-agents to middle-agents; second, requesting responses from middle-agents; third, mediating requests against the knowledge on capabilities of registered provider-agents; fourth gathering responses from provider-agents; and fifth *composing* and *returning* the results to the requesting users. As outlined in Sect. 2.3, the main challenge of creating a versatile chatbot is the reusage of multiple well-designed chatbots in a jointly coordinated manner to scale beyond the scope of a single chatbot application. Therefore, in this paper, we focus on the integration of multiple chatbots in a multi-agent-based architecture by designing reasonable capability-based middle-agents as intermediaries orchestrating hidden chatbots in a single conversational agent (cf. Orchestration Layer in Fig. 1). Against this backdrop, we take an integrated perspective by not only ranking chatbot answers based on priority lists or human feedback [12, 13] but rather combining research streams by embedding chatbots as provider-agents into a multi-agent-based architecture with middle-agents as intermediaries (cf. Fig. 1).

Focusing on the design of reasonable capability-based middle-agents, we take a standardized agent communication language (e.g. KQML or FIPA ACL [32]) for granted to automatically process requests by enabling communication among different agents. Furthermore, we state a requester-agent with a user interface as given, which receives textual requests from and provides responses to a real user (cf. User Layer in Fig. 1). Beyond that, we also state multiple chatbots as given, whereby each chatbot is designed as reactive provider-agent by responding to an answer, when receiving a request (cf. Chatbot Layer in Fig. 1).

Focusing on the orchestration of multiple chatbots, our approach comprises three steps that are sequenced in ten substeps (cf. Fig. 1). In the first step of our approach, we design a middle-agent to learn chatbot capabilities or requests they are able to answer, respectively. In order to determine which of the chatbots is most appropriate for a prospective request, different supervised machine-learning models are trained (learning agent). In the second step of our approach, an intermediary agent identifies all chatbots, which have the potential to answer a user request. To do so, this agent coordinates other middle-agents to preprocess requests, decomposes them into topics and thus, classifies these topics based on the learned classification model of the first step. The third step of our approach builds upon the classified topics of the second step and determines an execution plan, which in turn is used to request appropriate chatbot(s) and gather responses. Subsequently, all gathered responses are composed by the composing agent and published by the intermediary agent as an answer to the user (cf. Fig. 1). In the following subsections, we present our three-step-approach to orchestrate hidden chatbots in a single conversational agent in more detail.

3.2 First Step: Learning Chatbot Capabilities

The aim of the first step is to learn the capabilities of all available chatbots by using a matching mechanism that supports the intermediary middle-agent in identifying an appropriate chatbot regarding prospective user requests (cf. first step in Fig. 1). To do so, literature in multi-agent-based systems states that the choice of a suitable matching mechanism between request and agent capabilities certainly depends on the structure and semantics of the descriptions to be matched [19, 27]. In our context, solvable requests and all prospective similar requests of a chatbot can be treated as capabilities of a chatbot. Therefore, both, user requests and capabilities of the chatbots, in terms of requests they are able to answer, are represented as free-text (e.g. “Is there an airport in Nairobi?”). Thus, we base the identification of appropriate chatbots on text analysis consisting of the common substeps *data preprocessing*, *classification*, and *classifier evaluation*, which have been proven to deliver reliable results in identifying relevant and similar content in text-based requests [33, 34].

More precisely, the task of identifying the chatbot with the highest probability to answer a request is framed as a problem of supervised learning, where a machine-learning model is trained to generate the desired output from training data, which consists of pairwise training inputs (chatbot capabilities) and expected outputs (appropriate chatbot). Following the "divide and conquer" paradigm of multi-agent-based systems and in line with Hettige and Karunananda [28], we propose to split the text analysis into two separate reactive agents performing natural language *preprocessing* (cf. preprocessing agent in Fig. 1), *classifier learning* as well as *classifier evaluation* (cf. learning agent in Fig. 1). In particular, as the preprocessing has to be reused in the subsequent step to apply the classification model, it is reasonable to separate natural language preprocessing from classifier learning in different agents.

In order to learn a suitable model, the *learning agent* is triggered, whenever the available chatbots' capabilities change (e.g. due to adding, removing or changing a chatbot). Following this, the *preprocessing agent* is triggered by the learning agent and applies natural language preprocessing to clean textual data (e.g. eliminate irrelevant and redundant information) and to reduce the number of terms in order to obtain the minimum of relevant terms to improve speed as well as accuracy of classification algorithms [35]. Subsequently, the learning agent applies text classification algorithms (e.g. SVMs or artificial neural networks [36]) to the preprocessed chatbots' capabilities to train supervised machine-learning models. Based on common classifier evaluation measures (e.g. Precision, Recall, Accuracy and F1 score [37]), the learning agent aims at finding the most accurate classifier whenever the set of available classifier models changes. To sum up, the first step constitutes a necessary preparation for solving new requests triggered by changed chatbots capabilities. By doing so, the first step preprocesses chatbot capabilities from all available chatbots by means of the preprocessing agent (first substep). Following this, the learning agent trains a set of classifiers and chooses the model with the highest evaluation measure in order to determine which chatbot is most appropriate for a prospective request (second substep). Thus, a sound capability-based classification of requests in the following step is enabled.

3.3 Second Step: Identifying Relevant Chatbots by Classification

The aim of the second step is to identify all chatbots, which are necessary for solving a new request (cf. second step in Fig. 1). As outlined in Sect. 3.1, we base our architecture on the "divide and conquer" paradigm of multi-agent-based systems, where intermediaries receive requests and cope with the task of orchestrating multiple hidden conversational agents to solve the request. According to literature, capability-based (also known as skill-based or service-oriented) middle-agents can be categorized into matchmaker, mediators and broker [35] depending on the extent of providing mediation services. As matchmaker middle-agents just provide a ranked list of relevant provider-agents and give the choice of selecting a provider-agent to the user, they are not appropriate in our setting. In contrast, mediator and broker middle-agents actively answer the request by forwarding requests to the most relevant provider-agents. However, as mediators are based on a fixed number of agents and need a static pre-integrated global model of the application scope, they seem to be inappropriate as an intermediary in a multi-agent-based chatbot approach. Accordingly, we define the intermediary agent as a reactive broker agent interfacing between the user, middle-agents and chatbot agents (cf. intermediary agent in Fig. 1). More precisely, after receiving a new request, the intermediary agent first contacts the preprocessing agent to apply natural language preprocessing. As one request could refer to more than one topic and thus, could require responses from more than one chatbot, it seems reasonable to identify all contained topics prior to the capability-based classifications of the requests (e.g. "Which species of zebra is known as the common zebra (topic of chatbot one) and does an occupational disability insurance pay in case of an accident in Nairobi (topic of chatbot two)?"). Furthermore, it is necessary to maintain the structure and dependencies of the topics in order to enable the determination of an execution plan in the following step. This is in line with Skorochod'ko [38], who suggested to divide texts into subsentences regarding their semantic overlap and represented the resulting structure as a connected graph. Hence, we define the decomposing agent (cf. decomposing agent in Fig. 1) as a reactive agent, which automatically partitions preprocessed requests into coherent topics in a graph structure based on a well-established statistical text segmentation algorithm (e.g. based on lexical cohesion, decision trees or semantic networks) [38, 39]. Afterward, the reactive classification agent (cf. classification agent in Fig. 1) uses the trained classifier of the first step to identify the most relevant chatbot for each topic. To sum up, the intermediary broker-agent is triggered by a new request from the user (third substep) and coordinates the preprocessing agent (fourth substep), decomposing agent (fifth substep) and classification agent (sixth substep). As a result of the second step, the new request is decomposed into topics and a corresponding graph structure. Finally, the most relevant chatbot is identified for each topic.

3.4 Third Step: Obtaining Composed Answers

The aim of the third step is to plan and execute topics in order to get responses from all relevant chatbots as well as composing and publishing a final answer to the user (cf. third step in Fig. 1). Regarding multi-agent-based literature, deliberative agents are well-established to reason and plan actions based on a symbolic representation of a planning problem [25, 40]. Moreover, literature has also proven approaches for symbolic representation to be successful as a formal basis for deliberation, such as the well-known representation formalism named STRIPS (Stanford Research Institute Problem Solver) [19, 40, 41]. Therefore, we define the planning and executing agent (cf. planning and executing agent in Fig. 1) as a deliberative middle-agent based on STRIPS as planning formalism, which creates an execution plan based on the graph structure and the classified topics of the second step. Obviously, requests in conversational systems are characterized by short texts, which are usually comprised of only a few sub-sentences or topics, respectively. These topics could be independent and therefore parallelizable in an execution plan (cf. zebra and insurance example in Sect. 3.3) or sequentially dependent in their graph structure (e.g. "What is the currency of Nairobi (first topic) and what are $200 in that currency (second dependent topic)?"). However, in the case of sequential dependency, the dependent part of the second topic has to be replaced with the answer received by the execution of the first topic (e.g. answer of the first topic: "Kenyan Shilling"; reformulated second topic: "What are $200 in Kenyan Shilling"). According to literature, the task of reformulating requests by substituting or adding words or phrases to the original request is framed as a problem of query reformulation (e.g. substituting and/or adding words or phrases to the original query or topic, respectively) [42]. Furthermore, answers ought to be combined to create a final answer, which can be determined by the conjunction of all gathered answers. Consequently, we define the composing agent (cf. composing agent in Fig. 1) as a reactive agent to reformulate sequentially dependent topics based on a query reformulation approach as well as to combine answers based on a conjunction of all gathered answers. Summing up, the third step is triggered by the intermediary agent and builds upon the graph structure and the classified topics of the second step. To obtain a final answer, the planning and execution agent determines an execution plan (seventh substep), which in turn is used to request appropriate chatbot(s) (eighth substep). The composing agent supports this process by reformulating topics and answers (ninth substep), and finally, the intermediary agent publishes the composed answer to the user (tenth substep).

4 Demonstration and Evaluation of the Novel Approach

As an essential part of the DSR process [20, 29, 31], we demonstrate and evaluate the practical applicability of our approach. First, we describe the chatbot datasets considered. Then, we demonstrate how the three steps of our novel multi-agent-based chatbot approach could be applied. Finally, we present the results of our application and compare them to baselines and a competing artifact.

4.1 Chatbot Datasets

In order to demonstrate the practical applicability and evaluate the effectiveness of our approach, we use six different chatbot datasets comprised of 116.060 distinct questions and answers. The chatbots can roughly be divided into three main categories with two representatives in each category. First, we use datasets from two popular chitchat chatbots: the Loebner Prize winner MITSUKU [43] (26.78% of all distinct questions) and a successor of ALICE [9] called ROSIE [44] (15.29%). Second, we use chatbots based on question-answer datasets derived from the well-known knowledge bases Quora [45] (13.43%) and Wikipedia [46] (8.28%). These chatbots are capable of answering general knowledge questions beyond the capabilities of the two chitchat chatbots. Third, we use two domain-specific chatbots based on a dataset of algebraic word problems provided by Ling et al. [47] (34.39%) and based on an insurance-specific dataset provided by a major German insurer [48] (1.83%).

As these chatbots are specialized in their scope in terms of distinct capabilities (e.g. only insurance-specific questions or questions available on Quora) and could be mutually enriched by each other's contributions, they foster the need for a chatbot approach composed of jointly acting chatbots. Moreover, the requests of all six chatbots can be easily combined by the conjunction of two different topics. Thus, our demonstration dataset is comprised of 116.060 single requests and 6.73 billion combined requests each comprised of two independent topics. While focusing on the conjunction of independent topics, the considered chatbot datasets provide an appropriate setting to demonstrate and evaluate the applicability of our novel approach orchestrating hidden chatbots in a single conversational agent.

4.2 Demonstration

In order to demonstrate the applicability of our approach, we use Aimpulse Spectrum [49] as a runtime environment for large multi-agent systems providing a platform to design and execute reactive as well as deliberative agents based on FIPA ACL [32] as standardized agent communication language. In order to apply our approach, we first create six reactive chatbot provider-agents, which are able to receive requests and provide an answer based on the chatbots and their datasets described in the previous section. Moreover, we implement the user as reactive requester-agent, forwarding requests to the intermediary agent and receiving answers from the intermediary agent. Based on the chatbot agents and the requester-agent in the multi-agent environment, we apply our approach. Following the first step of our approach, we use the requests of six chatbots to train a set of classifiers and thus learn the capabilities of each chatbot. Therefore, we rely on the vector space representation and apply the preprocessing agent insofar as terms of the requests had been cleared from stopwords and punctuations, transformed to lower case, reduced to their word stems, and weighted based on the relevance of individual terms by applying the well-established frequency measure tf-idf [33, 34]. Following this, the learning agent uses the most common classification algorithms decision tree classification, support vector machine (SVM), k-nearest neighbor (KNN), naïve-Bayes and artificial neural network (ANN) [36] to learn a set of classifier models. To do so, the learning agent applies 10-fold cross-validation [50] for each classifier by using 90% of the 116.060 solvable requests for training and the remaining requests for classifier evaluation.

In order to ensure a rigorous evaluation and to quantify the quality of the results, the agent chooses the best available model based on the performance measure F1 score as this measure is widely used to assess the results of text analysis and condenses as well as balances the information entailed in the measures recall and precision [37]. The results of our first step reveal that the SVM classifier model constitutes the best available model in our setting to learn chatbot capabilities as it shows the highest average values regarding all evaluation metrics of the 10-fold cross-validation (cf. Table 1).

Table 1. Results of chatbot classification (maximum values are marked bold)

Approach	Recall (%)	Precision (%)	Accuracy (%)	F1 score (%)
KNN	80.15	83.83	81.33	79.76
Naïve Bayes	88.78	78.67	85.69	82.14
Decision Tree	88.72	90.14	92.04	88.64
ANN	94.53	95.06	96.38	94.78
SVM	**94.99**	**95.07**	**96.42**	**94.97**

For the second step, we base the text segmentation of the decomposing agent on the occurrence of question marks as the majority of topics contain a question mark at the end of their textual description. Even though using a question mark can be considered as a simple parametrization of our decomposing agent, in our setting, it seems to be a suitable indicator identifying the boundaries of topics. Consequently, the preprocessing prior to the decomposing agent must not perform a removal of punctuations, which contrasts with the preprocessing of the classification agent, which need to be identical to the preprocessing described in the first step to apply the classification model. Concerning the results of the text segmentation, we confirm that question marks are a suitable indicator in our context to identify the boundaries of topics. The majority of the 6.73 billion combined requests are well segmented into topic by the decomposing agent based on the occurrence of question marks (recall 100%, precision 98.33%, accuracy 98.33%, F1 score 99.16%). We found that only a few topics mislead our text segmentation approach by containing more than one question marks or a question mark followed by textual descriptions. In order to perform the third step and thus to determine a request-specific plan, an automatic deliberation is required. As the requests in our dataset are based on the conjunction of independent topics (cf. Sect. 4.1), all topics can be executed in parallel without any dependencies. Furthermore, and to create a final answer, the composing agent combines all independent answers based on a conjunction of all gathered answers as described in Sect. 3.4. Summing up, to create a composed answer regarding a new user request, the intermediary agent orchestrates other middle- and chatbot agents. Hence, multiple chatbots are reused in a jointly coordinated manner to scale beyond the scope of a single chatbot application.

4.3 Evaluation

To evaluate our approach as demanded by the DSR process [20], we follow the Framework for Evaluation in Design Science (FEDS) put forward by Venable, Pries-Heje, and Baskerville [51]. To this end, our evaluation strategy consists of comparing the results obtained by our novel multi-agent-based chatbot approach, a competing artifact and three baselines to the expected output. More precisely, we simulated a real user by randomly demanding requests out of the 6.73 billion combined requests as described in Sect. 4.1. While doing so, we applied 10-fold cross-validation [50] and skipped requests comprised of topics used while training the classifier in the first step. On this basis, we inspected the results of our approach relying on the average values of the well-established evaluation measures precision, recall, accuracy and F1 score [37]. To do so, we expected the conjunction of answers contained in the chatbot datasets (cf. Sect. 4.1) as correct output regarding a combined request. Moreover, we compared the results of our approach to the only competing artifact from chatbot literature (Huang et al. [12]) which can be applied in a multi-agent-based chatbot context without the presence of priority lists or a huge amount of user-provided ratings and feedback. Huang et al. [12] provide an appropriate approach, which is able to choose chatbots over time by ranking them based on the similarity of the new request and each chatbots' history (capabilities) as well as a specific prior probability of each chatbot. Although the prior probability is based on user feedback, Huang et al. state that the prior probability of chatbots without any feedback can be initially assigned to a constant value and updated with the presence of feedback [12]. Hence, we used the approach of Huang et al. [12] without any feedback as first competing artifact. Furthermore, we also compared our results to three multi-agent-based chatbot approaches as baselines, each selecting the answering chatbots based on a specific probability distribution. By doing so, the first sampling approach used an equalized distribution for choosing the answering chatbot (probability of 16.67% for choosing each chatbot); the second applied a distribution depending on each chatbots' dataset size (e.g. probability of 26.78% for choosing MITSUKU); and the third, skewed, baseline preferred the chatbot with the highest amount of requests (preferring the Chatbot dealing with algebraic word problems with a probability of 99.99%). Note that baselines and competing artifact are not able to decompose or compose requests and thus, our decomposing and composing agents are used to decompose and compose combined topics in our comparison. Regarding the comparison with baselines and competing artifact, shown in Table 2, our approach answered 94.78% of the user requests successfully. In contrast, the approach based on Huang et al. [12] reached a value of 73.78% successfully answered requests. With 33.82%, the skewed sampling approach reached the third highest value for successful answers, but the lowest F1 score of all approaches. However, the skewed sampling approach mainly just identified the correct answers regarding the scope of one chatbot, leading to high recall (98.33%) and precision (34.01%) for this chatbot but a critically low precision (0.11%) and recall (0.001%) regarding the scope of the other five chatbots. Finally, the equalized and size-dependent sampling performed worst by only obtaining 16.39% and 23.46% of the successful answers. Summing up, the results of our comparisons reveal that our approach provides higher quality than the competing artifacts and thus, indicate the suitability of our novel chatbot approach to scale beyond the scope of a single chatbot.

Table 2. Evaluation measures of our approach in comparison with baselines and competing artifacts (maximum values are marked bold)

Approach	Recall (%)	Precision (%)	Accuracy (%)	F1 score (%)
Equalized sampling	16.39	16.41	16.39	14.45
Size-dependent sampling	16.39	16.39	23.46	16.39
Skewed sampling	16.39	5.76	33.82	8.43
Huang et al. [12]	76.97	77.33	73.78	72.69
Novel approach	**91.11**	**93.11**	**94.78**	**92.12**

5 Conclusion, Limitations and Future Research

Despite emerging scientific work in the field of conversational agents, we still observe a lack of research on how multiple well-designed chatbots could be reused in a jointly coordinated manner to scale beyond the scope of a single chatbot application. Thus, we, proposed an approach consisting of three steps allowing us to decompose a user request into topics, classifying the answering chatbot for each topic and finally obtaining a composed answer. The evaluation of our approach using six chatbots provides promising results and illustrates the ability of our approach to scale beyond the scope of a single chatbot by orchestrating chatbots with different capabilities.

Thus, our work contributes to literature by taking an integrated perspective by not only ranking chatbot answers based on priority lists or human feedback but rather combining research streams by embedding chatbots as provider-agents into a multi-agent-based architecture. Furthermore, our novel approach paves the way for immensely versatile chatbot applications insofar as our chatbot approach is capable of scaling the scope of answerable questions and topics depending on the amount of integrated chatbots. In practice, for example, using our approach a company can combine freely available chitchat chatbots or certain specific chatbots with their domain-specific chatbot in order to enhance response options and capabilities. In addition, our approach might be used to address constraints such as "information overload" and "redundant information" mentioned in Stoeckli et al. [52]. Against this background, our novel multi-agent-based chatbot approach constitutes a promising first step in order to overcome current challenges in giving reasonable answers to a variety of possible requests.

Even though our research provides a sound integration of chatbots into a multi-agent-based architecture, there are some limitations, which can serve as starting points for future research. First, we only considered the decomposing, planning and composing of single-turn requests and independent topics in our demonstration and evaluation. While in a first step, it seemed appropriate to take such a perspective, future studies can enhance the agents of the second and third step in order to demonstrate and evaluate our novel approach in terms of sequential dependent topics. Moreover, in our demonstration and evaluation, we consider only a simple parametrization and compare our approach with competing artifacts applicable in a multi-agent-based chatbot context. Thus, we encourage further research to evaluate our approach using more complex parametrizations and competing artifacts in non-multi-agent-based contexts. Finally, it would be of interest to analyze the feedback of real chatbot users in order to evaluate the proposed approach in a real-world application as well as integrate dynamic learning strategies that consider context information and personal preferences of users.

Summing up, we believe that our study is a first, but an indispensable step towards multi-agent-based chatbots. We hope our work will stimulate further research in this exciting area and encourage scientists as well as practitioners to reuse existing chatbots by applying the "divide and conquer" paradigm of multi-agent-based systems.

References

1. Ahmad, N.A., Che, M.H., Zainal, A., et al.: Review of chatbots design techniques. IJACSA **181**(8), 7–10 (2018)
2. Klopfenstein, L.C., Delpriori, S., Malatini, S., et al.: The rise of bots: a survey of conversational interfaces, patterns, and paradigms. In: Proceedings of the 12th Conference on Designing Interactive Systems, pp. 555–565 (2017)
3. Chaves, A.P., Gerosa, M.A.: Single or multiple conversational agents? An interactional coherence comparison. In: Proceedings of the 36th CHI (2018)
4. Masche, J., Le, N.-T.: A review of technologies for conversational systems. In: Proceedings of the 5th ICCSAMA, pp. 212–225 (2017)
5. Dhanda, S.: How chatbots will transform the retail industry. Juniper Research (2018)
6. Abdul-Kader, S.A., Woods, J.C.: Survey on chatbot design techniques in speech conversation systems. IJACSA **6**(7), 72–80 (2015)

7. Chen, H., Liu, X., Yin, D., et al.: A survey on dialogue systems: recent advances and new frontiers. ACM SIGKDD Explor. Newslett. **19**(2), 25–35 (2017)
8. Ramesh, K., Ravishankaran, S., Joshi, A., Chandrasekaran, K.: A survey of design techniques for conversational agents. In: Kaushik, S., Gupta, D., Kharb, L., Chahal, D. (eds.) ICICCT 2017. CCIS, vol. 750, pp. 336–350. Springer, Singapore (2017). https://doi.org/10.1007/978-981-10-6544-6_31
9. Wallace, R.S.: The anatomy of ALICE. In: Epstein, R., Roberts, G., Beber, G. (eds.) Parsing the Turing Test, pp. 181–210. Springer, Dordrecht (2009). https://doi.org/10.1007/978-1-4020-6710-5_13
10. Serban, I.V., Sankar, C., Germain, M., et al.: A deep reinforcement learning chatbot (2017)
11. Pichl, J., Marek, P., Konrád, J., et al.: Alquist: the Alexa prize socialbot. In: Proceedings of the 1st Alexa Prize (2017)
12. Huang, T.-H.K., Chang, J.C., Bigham, J.P.: Evorus: a crowd-powered conversational assistant built to automate itself over time. In: Proceedings of the 36th CHI (2018)
13. Papaioannou, I., Curry, A.C., Part, J.L., et al.: Alana: social dialogue using an ensemble model and a ranker trained on user feedback. In: Proceedings of the 1st Alexa Prize (2017)
14. Pinhanez, C.S., Candello, H., Pichiliani, M.C., et al.: Different but equal: comparing user collaboration with digital personal assistants vs. teams of expert agents (2018)
15. Janarthanam, S.: Hands-On Chatbots and Conversational UI Development. Packt Publishing, Birmingham (2017)
16. Chandar, P., et al.: Leveraging conversational systems to assists new hires during onboarding. In: Bernhaupt, R., Dalvi, G., Joshi, A., Balkrishan, D., O'Neill, J., Winckler, M. (eds.) INTERACT 2017. LNCS, vol. 10514, pp. 381–391. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67684-5_23
17. Jennings, N.R.: Commitments and conventions: the foundation of coordination in multi-agent systems. Knowl. Eng. Rev. **8**(3), 223–250 (1993)
18. Jennings, N.R.: An agent-based approach for building complex software systems. Commun. ACM **44**(4), 35–41 (2001)
19. Klusch, M., Sycara, K.: Brokering and matchmaking for coordination of agent societies. a survey. In: Omicini, A., Zambonelli, F., Klusch, M. (eds.) Coordination of Internet Agents, pp. 197–224. Springer, Heidelberg (2001). https://doi.org/10.1007/978-3-662-04401-8_8
20. Peffers, K., Tuunanen, T., Rothenberger, M.A., et al.: A design science research methodology for information systems research. JMIS **24**(3), 45–77 (2007)
21. Maglio, P.P., Matlock, T., Campbell, C.S., Zhai, S., Smith, B.A.: Gaze and speech in attentive user interfaces. In: Tan, T., Shi, Y., Gao, W. (eds.) ICMI 2000. LNCS, vol. 1948, pp. 1–7. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-40063-X_1
22. Cui, L., Huang, S., Wei, F., et al.: Superagent. A customer service chatbot for e-commerce websites. In: Proceedings of the 55th Annual Meeting of the ACL, pp. 97–102 (2017)
23. Arentze, T., Timmermans, H.: Modeling the formation of activity agendas using reactive agents. Environ. Plan. B **29**(5), 719–728 (2002)
24. Ehlert, P., Rothkrantz, L.J.M.: Microscopic traffic simulation with reactive driving agents. In: 4th Proceedings of IEEE Intelligent Transportation Systems, pp. 861–866 (2001)
25. Rao, A.S., Georgeff, M.P.: BDI agents. In: 1st ICMAS, pp. 312–319 (1995)
26. Barua, A., Whinston, A.B., Yin, F.: Value and productivity in the internet economy. Computer **33**(5), 102–105 (2000)
27. Decker, K., Sycara, K., Williamson, M.: Middle-agents for the internet. In: Proceedings of the 15th IJCAI, pp. 578–583 (1997)
28. Hettige, B., Karunananda, A.S.: Octopus: a multi agent chatbot. In: Proceedings of the 8th International Research Conference, pp. 41–47 (2015)
29. Gregor, S., Hevner, A.R.: Positioning and presenting design science research for maximum impact. MIS Q. **37**, 337–355 (2013)

30. Baskerville, R., Baiyere, A., Gregor, S., et al.: Design science research contributions: finding a balance between artifact and theory. JAIS **19**, 358–376 (2018)
31. Hevner, A.R., March, S.T., Park, J., et al.: Design science in information systems research. MIS Q. **28**, 75–105 (2004)
32. Labrou, Y., Finin, T., Peng, Y.: Agent communication languages: the current landscape. Intell. Syst. Appl. **14**(2), 45–52 (1999)
33. Park, S., An, D.U.: Automatic e-mail classification using dynamic category hierarchy and semantic features. IETE Tech. Rev. **27**(6), 478–492 (2010)
34. Li, N., Wu, D.D.: Using text mining and sentiment analysis for online forums hotspot detection and forecast. DSS **48**(2), 354–368 (2010)
35. Storn, R., Price, K.: Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. **11**(4), 341–359 (1997). https://doi.org/10.1023/A:1008202821328
36. Russell, S.J., Norvig, P.: AI. A Modern Approach. Pearson Education, London (2010)
37. Sokolova, M., Japkowicz, N., Szpakowicz, S.: Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation. In: Sattar, A., Kang, B. (eds.) AI 2006. LNCS (LNAI), vol. 4304, pp. 1015–1021. Springer, Heidelberg (2006). https://doi.org/10.1007/11941439_114
38. Skorochod'ko, E.F.: Adaptive method of automatic abstracting and indexing. In: Proceedings of the 5th Information Processing Congress, pp. 1179–1182 (1972)
39. Beeferman, D., Berger, A., Lafferty, J.: Statistical models for text segmentation. Mach. Learn. **34**(1–3), 177–210 (1999). https://doi.org/10.1023/A:1007506220214
40. Wooldridge, M., Jennings, N.R.: Intelligent agents: theory and practice. Knowl. Eng. Rev. **10**(2), 115–152 (1995)
41. Fikes, R.E., Nilsson, N.J.: STRIPS: a new approach to the application of theorem proving to problem solving. Artif. Intell. **2**(3–4), 189–208 (1971)
42. Dang, V., Croft, B.W.: Query reformulation using anchor text. In: Proceedings of the 3rd WSDM, pp. 41–50 (2010)
43. Mitsuku Dataset. https://github.com/pandorabots/Free-AIML. Accessed 06 Dec 2019
44. Rosie Dataset. https://github.com/pandorabots/rosie. Accessed 06 Dec 2019
45. Quora Dataset. https://www.kaggle.com/c/quora-question-pairs. Accessed 06 Dec 2019
46. Wikipedia Dataset. https://www.kaggle.com/rtatman/questionanswer-dataset. Accessed 06 Dec 2019
47. Ling, W., Yogatama, D., Dyer, C., et al.: Program induction by rationale generation: learning to solve and explain algebraic word problems. In: Proceedings of the 55th Annual Meeting of the ACL, pp. 158–167 (2017)
48. Bedué, P., Graef, R., Klier, M., et al.: A novel hybrid knowledge retrieval approach for online customer service platforms. In: Proceedings of the 26th ECIS (2018)
49. Aimpulse Spectrum. https://developer.aimpulse.com. Accessed 23 Aug 2019
50. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proceedings of the 14th IJCAI, vol. 14, no. 2, pp. 1137–1145 (1995)
51. Venable, J., Pries-Heje, J., Baskerville, R.: FEDS: a framework for evaluation in design science research. Eur. J. Inf. Syst. **25**, 77–89 (2016)
52. Stoeckli, E., Uebernickel, F., Brenner, W.: Exploring affordances of slack integrations and their actualization within enterprises-towards an understanding of how chatbots create value. In: Proceedings of the 51st HICSS (2018)