Advising Agent for Service-Providing Live-Chat Operators

Aviram Aviv[1][0000−0003−1451−7281], Yaniv Oshrat(✉)[1][0000−0003−3785−4977], Samuel Assefa[2], Toby Mustapha[3], Daniel Borrajo[4][0000−0001−5282−0463], Manuela Veloso[3][0000−0001−6738−238X], and Sarit Kraus[1][0000−0003−4672−623X]

[1] Bar-Ilan University, Ramat-Gan, Israel
aaviv10@gmail.com, oshblo@zahav.net.il, sarit@cs.biu.ac.il
[2] US Bank AI Innovation, United States
sammy.assefa@gmail.com
[3] JP Morgan AI Research, New-York, United States
tmusta78@gmail.com, manuela.veloso@jpmorgan.com
[4] University Carlos III of Madrid, Spain
dborrajo@ia.uc3m.es

Abstract. Call centers, in which human operators attend clients using textual chat, are very common in modern e-commerce. Training enough skilled operators who are able to provide good service is a challenge. We propose a methodology for the development of an assisting agent that provides online advice to operators while they attend clients. The agent is easy-to-build and can be introduced to new domains without major effort in design, training and organizing structured knowledge of the professional discipline. We demonstrate the applicability of the system in an experiment that realizes its full life-cycle on a specific domain, and analyze its capabilities.

Keywords: Human study · advising agent · human-agent interaction · call center.

1 Introduction

In modern e-commerce, many business services are provided via the Internet. Not only do new web-oriented enterprises use this option, but traditional ones as well have moved relevant services to the digital medium. For example, banks are increasingly closing their physical branches and moving services, formerly provided only face-to-face, to the internet [42]. There are many actions that customers can perform by themselves via the Internet, without human intervention, either by a self-service application or using a conversational chatbot. However, when customers want to perform actions that do not yet have an online solution, or when they fail to do it by themselves, they still need to approach the bank's customer service and seek human help.

There are several communication channels between the customers and the call center employees (operators). The first method is a telephone call. This method gives the customer the full attention of someone capable of helping, but at the same time it forces the operator to wait for the customer's reactions. In many of these calls, the customers need to perform actions with which they are not familiar, making the operator wait idly for the customers to finish. Since the operator can attend only one customer at a time, this approach wastes time that could be better utilized. With the rise of the Internet, another approach for call centers emerged, using a text-based chat service. This method obviates the constraint of giving one customer the full attention of the operator, as it parallels the service. Instead of talking with one customer at a time, the operator interacts textually with 2-4 customers in parallel, switching between customers instead of waiting for the customer's reactions.

While this approach has its advantages, it also raises some challenges for the human operator to deal with. As the number of tasks that the operators have to perform simultaneously grows, so may their stress. Operators also need to prioritize the tasks, keep track of each individual's information while assisting different clients, and provide help without making any client wait too long.

We propose to mitigate these challenges by assisting the human operator in creating an **advising agent**. This kind of agent works alongside the operator during the chat session, and suggests on-line advice to help the operator deal with a given situation. To the best of our knowledge, we are the first to use an advising agent to cope with these challenges. However, building an advising agent and training it to the specific service domain can be a long and expensive process that requires both domain expertise and system engineering knowledge.

In this work we present a design for an automated agent that assists the operator during textual chat interactions with customers in real-time, by providing the operator with advice about possible actions and relevant information. Our design combines standard ML methods with domain-expert annotations, and tries to predict the actions and suggestions of the expert. The novelty of our method is twofold: First, the assistance of the agent is not focused on providing answers to customers' questions (as in former works, e.g [11, 20]), but rather in guiding the operator as to what questions she should ask in order to get the required information to provide service. Second, the process of training the agent to a new domain is short and does not require many resources or domain knowledge from outer sources. Finally, we field-test our design on a specific domain and present our findings.

2 Related Work

2.1 Call Centers

Many research fields look at call centers as a source of interesting problems to study. Some of them analyze the call center as a business to be run, trying to improve the total income and customer satisfaction [24], predict customer abandonment [23], or predict attrition rates [12]. Other fields examine the effect of working in call centers on the human employees [29]. However, when looking at call centers from the computer science point of view, most of the research is focused on solving problems like staffing, customer queuing or scheduling processes [34, 31, 1, 51, 30].

In recent years, many companies have started to develop chatbots for the task of customer service [45]. There is much work regarding the design and deployment of such bots in various domains [46]. It is evident that currently chatbots cannot fully replace human workers, and when a bot detects that it cannot help the customer, it refers the customer to human help. Li *et al.* [36] and Liu *et al.* [37] explore the challenge of detecting this kind of situation in various domains.

Lee *et al.* [33] show that chatbots can reduce the human workload, but the change is almost imperceptible (less than 5% improvement). The vast majority of the problems found refer to parts of the call center that have little to no effect on the human workers. To the best of our knowledge, this is the first research study that tries to help the human worker directly by providing advice.

2.2 Agents that Support Human Actions

Intelligent agents that support humans in their complex activities need to be able to predict the user's behavior and decisions [4, 48, 32, 49]. This is a difficult task because of an extensive set of factors that influence human decision-making and behavior [15], such as experience [26], decision complexity [19] and emotions [7]. These factors also include inherent differences between individuals and groups of individuals [9], which make the prediction of an individual's decisions and behavior even more challenging [44].

There are several previous methods for prediction of human behavior and decisions in agent-human interactions. Azaria *et al.* [6] developed CARE (automobile Climate control Advisor for Reducing Energy consumption) – an agent that uses two models: one for predicting the influence a certain climate has on the human driver, and one for estimating the energy consumption of a particular setting. The agent finds a compromise between them and offers it to the driver, who chooses whether or not to accept it. Rosenfeld *et al.* [47] developed automated agents that can assist a single operator to better manage a group of robots in a search task and a warehouse operation task, showing that an agent can significantly improve the performance of a team comprised of an operator and ten low-cost mobile robots. The agent uses the world state to determine which human actions are urgent and which actions can wait for later. This work also compared two approaches of advice: one looks for the advice that will have the best impact on the current situation, and the other searches for advice that will lead to better results in the near future (2 or 3 actions ahead). In both domains, the agent gave the operator advice about what should be done next, depending on the world state at every moment.

2.3 Method of Advice Provision

When it comes to advising in repeated human interaction environments, several methods have been used in the literature. Rosenfeld *et al.* [47] directly estimated the reward for every possible piece of advice and recommended a piece of advice that maximizes the reward that the user will get if the provided advice is chosen. Elmalech *et al.* [16] suggested that the agent will try to find a compromise between maximizing rewards and user acceptance, and Azaria *et al.* [5] used human models inspired by behavioral economic theories for advice provision. The common ground of these advising agents is that they all advise in order to maximize a certain reward function. The drawback of this kind of advising mechanism is that it tends to recommend non-intuitive advice that the operators often reject, making it ineffective [10].

2.4 Learning How to Provide Advice

In recent years, companies began keeping records of their workers' actions and their interactions with the customers due to low digital storage costs [27]. This accumulation of information was mostly used for basic performance analysis, but with the improvement of machine learning capabilities this abundance called for new uses [17].

Using human actions and decisions as the base of the learning process has many names in the literature such as learning from observation, learning by demonstration, or mimic agent, among others. Argall *et al.* [2] united many of these names under "learning from demonstration" (LfD), and mentioned that LfD does not require expert knowledge of the domain dynamics, an essential notion for our research as we use demonstrations from people with little to no experience in the field. The LfD approach is used in a large variety of fields (e.g, [54, 18]). In our context, Levy and Sarne [35] combined LfD and advice provision as they created an agent that used the way people act in a specific scenario in order to guess what they would do in similar situations.

Even though there are many examples in the literature of ways to generate conversational data [13, 39, 38], we focus on using human-human conversations for the learning process of the agent because they better reflect the real-life scenario [53] and hopefully help in generating more intuitive advice.

3 Modus Operandi and Life-Cycle of the Agent

Our research goal is to develop an easy-to-build, data-driven method for an automated system that will assist in the operators' training process and daily activities, will help new and experienced operators, and will advise the operators about possible actions and relevant information during textual chat interactions with customers in real-time. Implementing the automated system (i.e., the agent) in a call center has the potential to reduce the daily workload and improve interaction with the customers from the human operator's point of view. It will improve the system's service efficiency and reduce the time needed to help each customer.

3.1 Agent's Life-Cycle

The process of building an advising-agent for a new domain is performed in three phases, as follows:

1. **The Apprentice Phase (Phase 1)** – experienced human operators serve human customers regarding the new domain of service. The operators tag the information they find important in the chat conversation: They may do it in real-time, as the chat goes on, or afterwards. The collected data is fed to the learning process (as detailed in Section 3.2). This phase exists only for the sake of collecting information for the next phases, and does not include any agent assistance. Section 4.2 elaborates about the experimentation of this phase.
2. **The Novice-Advisor Phase (Phase 2)** – this phase contains both data collection (for the improvement of the agent's capabilities) and service to clients: the agent works alongside a non-experienced human operator who attends clients, and it simultaneously advises and learns. For advising the human operator, the agent uses the tagging from the chat conversation to predict messages that the operator should send or actions it should perform, and offers them to the operator. The operator may use this advice or not, as suits her.
   In addition, the data collected in this phase may be fed into the agent's machine learning model in order to improve its tagging and advising capabilities. This feeding may be performed daily, weekly, monthly or in any batch form that is suitable to the managers of the service. Additional details regarding this phase are presented in Section 4.3.
3. **The Expert-Advisor Phase (Phase 3)** – The agent works alongside a non-experienced human operator and provides advice based on former tags and a learned model. The agent is not engaged in further learning, since its capabilities have already reached an adequate level. This phase is the final and steady state of the agent in the current domain.

A rollback from Phase 3 to previous phases may be performed if needed (as elaborated on in Section 4.4). The system can be returned to Phase 1 or to Phase 2 (according to the managers' preferences), collect additional data (i.e. tagged chat conversations) and feed them to the machine learning model. Upon reaching the desired level of advice, the agent may be advanced again to Phase 3, and so forth.

The 3-Phase life-cycle was chosen because it enables the adjustment of the phase to the opportunities and the needs of the users: It uses the knowledge of experienced operators in Phase 1, it combines exploration and exploitation in Phase 2, and enables steady production in Phase 3. As pointed out in the previous paragraph, the system may be switched between phases according to the needs of the users and other circumstances.

This model suggests a method to implement an assisting agent in a new domain with a relatively small effort: The needed knowledge is derived from authentic dialogues with clients, that is to say it uses the resources that are already invested to build the domain knowledge. The specific agent that is built is suited to the specific domain, but the method is domain-independent, and it may be applied to a wide variety of domains.

3.2 The Learning Process

In order to provide advice, the agent relies on a predictive model learned from observations of the domain: Operators conduct chat sessions with clients and attend to their needs. During the chat sessions, the operators tag the vital information items they used to reach the satisfactory outcomes. An information item may include a single word or a phrase (a few words), and it depends on the specific domain in which the service is provided. All the tagging is done during the chat conversation or after it; there is no tagging in advance.

A tag contains **a label**, which is the category of the tag, and **information**, which cites the specific knowledge of the tag. For example, if an operator asks clients about their occupation, then the label of the tag will be "occupation", and possible information can be "engineer", "marketing manager", "driver", "none", etc. Figure 1 presents several examples of tags from various domains.

<vehicle><Ford Explorer>	<occupation><none>
<vehicle><Toyota Avensis>	<occupation><cost analyst>
<education><PhD>	<occupation><bus driver>
<education><BSc>	<occupation><accountant>
<education><MA>	<university><ucla>
<military service><yes>	<university><stanford>
<military service<no>	<university><cmu>

Fig. 1. Examples of tags. In this structure, the first item is the label of the tag and the second one is the information.

Each session's tag-list is turned into an **information vector**. Each time a new tag is added, the vector's current version is saved to be used later in the learning process as an information vector.

Building the Information Vector We build the information vector as follows: First, we take the n most common labels that operators marked in the data and sort them in alphabetical order (the label list).

Notation remark: We write X_t as the information vector after t pieces of information (that is, X at time t), and $X[t]$ for the value of X at index t. Whenever we mention tag i, we refer to the value of the tag list at index i.

We define two vectors of size n. The first one is:

$$V[i] := \begin{cases} item, & \text{if a known item was tagged as label i} \\ "unknown", & \text{if an unknown item was tagged as label i} \\ "-", & \text{if no input was tagged as label i} \end{cases}$$

A known item is an item that was already tagged (in previous chat conversations or previously in the current conversation). An unknown item is an item that has not yet been tagged.

The second vector is:

$$W[i] := \begin{cases} 1, & \text{if there is an input tagged as label i} \\ 0, & \text{otherwise} \end{cases}$$

We define $X[i] = (V[i], W[i])$ (that is, the vector X is made of (V, W) tuples). Figure 2 demonstrates the building of the information vector.

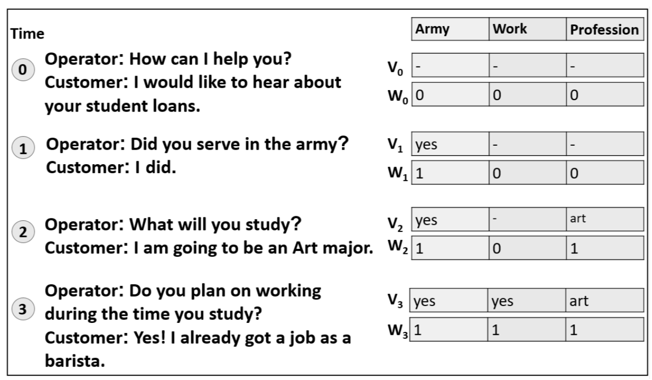

Fig. 2. The process of building an information vector as the chat between an operator and a client proceeds.

Advice Types There are three types of advice that we wish to provide: (1) Topic acquisition – questions the operator should ask the client in order to acquire information she needs in order to help him; (2) Resolution – data the operator should provide to the client as a response to his query; and (3) Useful information – data the operator may need in order to provide answers, such as relevant websites, calculations, etc.

Advice Providing Process During the chat conversation with the customer, whenever the operator uses a website operation or finds out new information about the customer, useful information is marked under a suitable label, and the customer information vector X is updated accordingly.

For each advice type i of the three mentioned above, advice is provided by taking the label vector X_t and inserting it into a machine learning module F_i that tries to find the best set of advice A for the current situation (A_t). The module uses k pairs $D_1 \ldots D_k$ of past experiences D_j=(X_j,A_j), where X_j is an information vector at time j and A_j is the respective set of advice, in order to find a set that maximizes the chance to be the most used set of advice in the past similar situations: $P(A_t = A|X_t, D_1, ..., D_k)$.

For the learning algorithm, we wanted to find an algorithm with the ability to work efficiently on several domains and handle messy and conflicting data. The first model that came to mind was Random Forest [8], a model that works well but cannot fully utilize the vast amount of data usually available in such domains. To deal with this problem, we thought of using neural networks. That idea was relatively successful, but an architecture that works on one domain might fail to learn on another. With all that in mind, we decided to combine them as an ensemble method of neural networks [21] where each network takes the information known about a customer at a certain time and outputs the recommended set of advice for the situation. Each network in the ensemble was trained on a subset of the data and had a random number of layers of an arbitrary length, as shown in Algorithm 1.

Algorithm 1: Training the ensemble:

Result: a list of trained neural networks
nets=$\emptyset$
while *length(nets)* $<$*ensembleSize* **do**
    num=GetRandomNumber()
    if *num* $>$*0.5* **then**
        trainSet=getRandomSubSet(trainData)
    else
        trainSet=getBalancedSubSet(trainData)
    end
    net=GenerateRandomNeuralNetwork()
    train(net,trainSet)
    P_{net}=accuracy(net,testData)
    if $P_{net} > P_{threshold}$ **then**
        nets $\xleftarrow{\text{add}}$ net
    end
end

The final set of advice was chosen using a majority voting variation (as shown in Algorithm 2). We also tested the ensemble method of neural networks against other variations of Random Forest (LGBM [28] and regular Random Forest) and other crowd related algorithms (SVM and KNN). This method outperformed the others in an 80:20 cross-validation where the target label needed to be in the top 2 recommendations (the ensemble reached 87% accuracy, regular and gradient boosted Random Forests with 84%, KNN with 83%, neural network with 77% and SVM with 70%). We chose this metric because there can be a large variation based on the operator's preferences, even with a small amount of data.

Algorithm 2: Using the ensemble:

```
Result: Final recommendations
results=∅
X=getData()
for net in ensemble do
    results ←add prediction(X)
end
bestOptions=twoMostCommonOptions(results)
finalRecommendations=∅
if rankOf(bestOptions[0]) >firstOptionThreshold then
    finalRecommendations ←add bestOptions[0]
end
if rankOf(bestOptions[1]) >secondaryOptionThreshold then
    finalRecommendations ←add bestOptions[1]
end
return finalRecommendations
```

As can be seen in Algorithm 2, the agent can recommend one set, recommend a combination of two sets, or remain silent (when $\emptyset$ is chosen or when finalRecommendations is empty).

4 Experiment

Our experiment was designed to test whether working with the suggested assisting agent improves operators' performance. For this purpose we chose a domain, set up a working environment and recruited participants to play the roles of operators and clients in various configurations. At the end of each session, the operators filled out questionnaires to quantify their opinions regarding different aspects of the performance. We analyzed the results, learned some lessons and made amendments to the model. We will now describe the setup and course of the experiments. The results will be presented in Section 5.

4.1 Experiment domain - student loans

The domain on which we chose to perform our experiment is student loans in the US. Customers who are interested in understanding their options in getting such loans, either for themselves or for their relatives (usually a son or a daughter), call the information center and chat with the operator. In some cases, the customers know what the relevant data is, and they can provide it to the operator right away. Nevertheless, in many cases the customers are not familiar with the parameters that define their entitlement to a loan, and they should be guided. For example, in the US men are required to register in the Selective Service System in order to be entitled to a federal loan. Many applicants are not aware of this requirement, and informing them of it, or of other parameters that might affect their ability to get a loan of the sum they need, is very beneficial. Good service by the operator should clarify these issues in order to enable the customer to exhaust his rights. Hence, there is much room for accurate advice to the operator in this process. Since Phase 3 in our model is the steady state working mode, we performed our field experience on phases 1 and 2 which implement the building of the model.

4.2 Phase 1 – the Apprentice Phase

As mentioned above, the goal of this phase is to provide the agent with labeled data regarding our domain by listening to sessions in which experienced human operators chat with clients. This phase was implemented in our experiment by recruits that played the operators and the customers. The operators were thoroughly briefed and trained about the domain and the service they should provide to customers. At the end of this preparation stage, it was assumed that the recruits were at the level of a practised operator in the domain of student loans. The customer received storyboards, each with character information (profession, university, savings, financial status etc.) and objectives to achieve (loan options, pre-specified information about the loans etc.).

The chat between the customers and the operators was performed using a textual chat application. We used “WhatsApp" as a basis for our interface, as this application is commonly used by businesses for communications with customers (e.g. [41, 43, 25]). The operators used a computer where half of the screen is a “WhatsApp web" interface with a special browser extension that knows when the operator switches between two conversations (as a single operator attended 2-3 customers simultaneously), and allows the operator to mark words. The other half of the screen shows a website which presents information and enables the operator to perform common calculations by clicking on pre-defined buttons.

The subjects played multiple client-operator sessions. In these sessions there was no participation of an assisting agent, and only the human operators and the human clients took part. During the sessions, in addition to collecting relevant information from the clients and answering clients' questions, the operators also tagged phrases in the chat. They were asked to tag (by marking words on the screen) any information that they considered relevant to the loan issue. Each tag contained a label (e.g. university name) and information for that label (e.g. UCLA, MIT, Columbia University). Operators were neither told nor limited regarding what labels of tags they could mark. They saw what labels were tagged and used before, but did not see their information. They could add additional tags as needed.

In this phase of experiment, 4 subjects took part as operators and one subject played the clients. Note that this subject played 2-3 clients simultaneously, but since the work of the operator is more complicated than the work of the clients, and since the experimenter who played the clients was practiced and followed pre-prepared scenarios, he was able to play more than one client simultaneously without causing a delay to the work of the operator. In total there were 76 sessions, and in each of them a single operator attended 2-3 simulated clients.

4.3 Phase 2 – the Novice-Advisor Phase

The goal of the Novice-Advisor Phase is twofold: To assist operators in their work, as well as to collect additional data for the improvement of the agent. In the experiment, our main goal was to evaluate the helpfulness of the agent we built in Phase 1.

In our experiment this phase was implemented using recruits from the AI course for undergraduate students in Bar-Ilan University as clients, and paid recruits from the general population as operators or clients. Each operator played two sessions: one with an agent's assistance and the other without it. Half of the operators played the assisted session first and the unassisted session second, and the other half vice versa. Each client played a single session, in which they received two different storyboards and played them with the operator.

At the end of each session, we asked the participants who played the operators to fill out a NASA-TLX questionnaire [22], which is an assessment tool for comparing the workload of different tasks (a summary of the NASA-TLX questionnaire can be found in Appendix A in [3]). These opinions were analyzed in order to evaluate the performance of the agent and its contribution to the performance of the operators. The findings of the analysis are presented in Section 5.

At this point we had 23 operators who played 46 sessions: 15 of the 23 operators attended 2 clients simultaneously and the remaining 8 operators attended 3 clients simultaneously. The tagging of the text was done manually by the operators during the sessions.

The Tagging Problem The tagging of the chat conversations is essential for the agent in order to follow the line of conversation and provide proper advice. Our preliminary design was to tag the chat by the human operator, in real-time, during the session. Unfortunately, we found out that the operators of Phase 2 managed to perform the tagging well while attending 2 clients simultaneously, but when they needed to attend 3 clients simultaneously the workload was too heavy, and they could not tag the conversation properly; as the session proceeded, there was much less tagging or none at all. As a result, the ability of the agent to provide advice weakened. This situation called for a change.

In order to perform good tagging even in stressed sessions, we introduced an automated tagging mechanism. We took the raw data in real-time and made the agent use it directly, a common notion in goal-oriented dialogue systems, and chatbots in general [50]. We used a machine learning module that follows the messages in real-time and outputs annotations for the advising agent. The module that we chose is a combination of two sub-models, as follows: We denote a network consisting of a BERT [14] embedding layer with a linear layer on top as a **BERTLL**. At first, a BERTLL predicts what labels the message may contain. For each label that the first model predicted, another BERTLL predicted what information the message may contain (again, see Figure 1 for the tag structure). We chose to use this combination after it reached a maximum F_1 score of 0.72 and was seen to generalize well in practice. It also outperformed a gradient boosted

Random Forest (that reached an F_1 score of 0.7), a single BERTLL for all the labels (that reached a maximum F_1 score of 0.5) and a large variety of neural network-based models that were far from reaching a 0.5 F_1 score. Implementing the automated tagging mechanism relieves the operator from the tagging task, and enables her to concentrate on the sole task of attending the clients.

Another improvement in the experiment method (relative to the original design) was the introducing of bots as clients in this phase: Instead of human subjects playing the role of clients, we deployed bots that were built using a combination of two strategies: a rule-based approach, and a learning approach. The first approach followed the spirit of early chatbots, such as Eliza [52]. Based on the previous interaction with the operator, the bot would randomly generate answers to operator's questions, or questions to ask of the operator. The second approach used BERT to learn how to perform the interaction. We found out that these two models (knowledge-based and learning-based) complemented each other quite well in overcoming their respective disadvantages. This change was made because the use of bots instead of human subjects made the experiment much easier to perform, since we needed to recruit and to brief only the operators, and the influence of the agent on the clients was not examined in this study.

A third change was to introduce a level test to the recruits who were to play the role of operators, after their briefing and training. In order to verify that the recruits are indeed trained and to an appropriate professional level, each of them took a short test with questions regarding the domain and the service. Only after successfully completing the test with high grades were the recruits allowed to move on to the experiment.

After implementing the aforementioned changes, we performed the experiment of Phase 3, this time with each operator attending 3 clients simultaneously. In this improved design we did not encounter an excessive load on the operators, since the tagging was done automatically by the agent. We had 14 operators playing two sessions each (again, half of the operators played the assisted session first and the unassisted session second, and the other half vice versa). Together with the 15 operators who attended 2 clients simultaneously, we had 29 operators, and each of them played 2 sessions.

The experiments were performed according to the institution's guidelines regarding experimenting with humans, and permission to perform the experiments was accepted from the institution IRB prior to the experiments. The demographic data regarding the subjects in the experiments is presented in Appendix B in [3].

4.4 Phase 3 – The Expert-Advisor Phase

As explained above, the goal of Phase 2 is to help operators while they serve clients, and at the same time to improve the capabilities of the agent to provide good advise in the domain. The system works in Phase 2 (i.e., new data is fed to the machine learning model) as long as managers feel the agent needs improvement and the performance of it indeed improves with the additional data. At a certain point there is no further need for improvement, and the system can be turned to Phase 3 - the Expert-Advisor Phase. The machine learning model is stabilized, and the collection of data is stopped. The agent works alongside the operators and provides advice according to the data that was collected in the previous phases. Therefore, there was no need to perform experiments on Phase 3.

Nevertheless, the experiment domain may illustrate the possibility of phase rollback described above (Section 3.1). In our experiment domain of student loans, if, for example, new terms of loans are available in the market, the steady-state agent will not know how to advise operators regarding them. In order to teach the agent about the new terms, the system should be returned to Phase 1 or to Phase 2, collect data (i.e. tagged chat conversations) and feed them to the machine learning model. When reaching the desired level of advice again, the agent may be advanced again to Phase 3, and so forth.

5 Results

5.1 Operators' Opinions

The participants who played the role of operators filled out NASA-TLX questionnaires. The goal of this process is to compare the grades regarding sessions that were played with the agent's assistance to the grades regarding sessions that were played without the agent's assistance, in order to learn about the operator's experience with the advising agent. The results of these questionnaires are presented in Figures 3-4: Fig. 3 presents the data of the experiment in which a human subject played the role of clients (two clients simultaneously). Fig. 4 presents the data of the experiment in which bots were deployed as clients (three clients simultaneously). In both cases, human subjects played the role of the operators. We present the total TLX grade, which sums the six categories of the questionnaire (as elaborated on in the Appendix A). In addition, we present the grade of the Temporal Demand category, since this category is of special concern in our model.

As we described in Section 4.3, each operator played two sessions. Therefore, the data in Figures 3-4 is presented in two views:

1. First Session - counts only the first session of every operator (whether with the assisting agent or without it).
2. Total Sessions - counts all of the sessions (both the first and the second) of all of the operators.

It can be seen that both total workload (Total TLX) and temporal demand **decreased in all cases** when having the agent working alongside the operator as compared to not having the agent. All data presented in Figures 3-4 are statistically significant ($p < 0.05$).

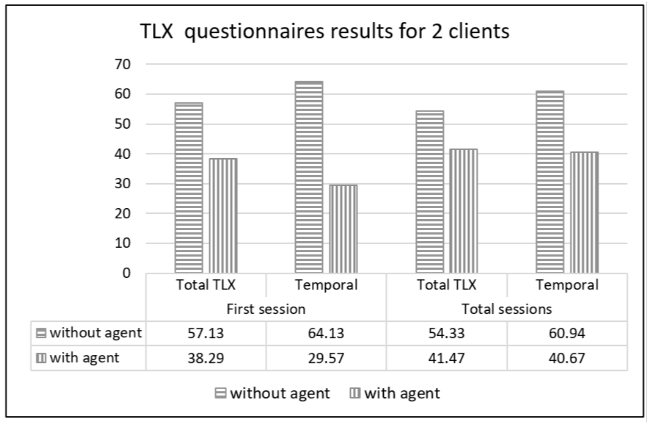

Fig. 3. NASA-TLX questionnaires' data of Phase 2 (lower is better). The experiment with 2 simultaneous clients was conducted using human subjects as clients.

5.2 Time Performance

We presumed that a good performance of the agent would be manifested in providing the service in less time, and with less idle time during the session. Figure 5 presents the time performance data of the operators in three categories:

1. Total session time - the average length of a full session, including clients' time, operator's time and idle time.
2. Maximal waiting time - the maximal time a client had to wait for an operator's response.
3. Total waiting time - the average total time a client spent waiting for an operator's responses during a session.

In all categories the times are shorter when the agent assisted the operator than when it did not, and in most of the categories the reduction is greater than 10%. It implies that the use of an agent alongside the operator reduces the time needed for the session in general as well as the time spent by the client idly waiting for the operator to respond. Nevertheless, the data was not found to be statistically significant in most of the categories.

5.3 Learning Effectiveness of Phase 2

The Apprentice Phase (Phase 1) is, naturally, crucial to the building of the preliminary knowledge base of the agent. Nevertheless, we wondered whether Phase 2 actually improves the capabilities of the agent, or if it is superfluous and we may skip it and go straight to the final stage (Phase 3). In order to answer this question we compared the performance of the tagging model in two configurations: The first one was based on data collected in Phase 1 only, while the second one was based on data collected in both Phase 1 and Phase 2. We found that

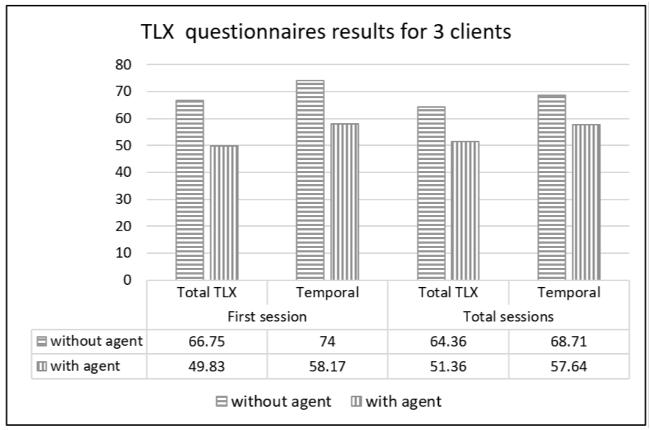

Fig. 4. NASA-TLX questionnaires' data of Phase 2 (lower is better). The experiment with 3 simultaneous clients deployed bots as clients.

the performance of the second model (precision – 78%, recall – 75%, F_1-score – 72%) was better than the performance of the first model (precision – 65%, recall – 60%, F_1-score – 58%). This result indicates that although the data of Phase 1 alone suffices to provide basic assistance to human operators, expanding it with the data of Phase 2 significantly improves the tagging capability and, as a result, the quality of the agent's performance.

6 Conclusions, Discussion and Future Work

In this paper we introduced an algorithm and a method to implement an advising agent that assists operators who attend clients in a call center using chat conversations. The main advantage of this method is its adaptability – the agent can be fitted to a new domain with relatively little effort and little time. Training the agent does not require prolonged design, domain analysis or rule-definition. In an existing human call center, the agent only needs tagged conversations of experienced human operators with clients in order to build all the required knowledge. Having said that, we still think that additional experimenting is needed in order to conclude that this method is domain-independent, and specifically it should be tested in other domains.

Integrating the results of the role-playing experiment, we see that operators who were assisted by the agent enjoyed a lower cognitive load in attending their clients, with less effort and less time-pressure. Time is used more efficiently, as sessions are shorter and less time is spent on idle waiting. This trend is evident both in the objective measure of time to perform a mission (Section 5.2) and in the subjective views of the participants who played the operators (Section 5.1).

There are several issues that still need to be examined. One such issue is optimization of the process of adjusting the agent to a new domain. We found

Time Performance (in minutes)				
Category	parameter	without agent	with agent	reduction
First session	total session time	40.86	34.29	16.1%
	maximal waiting time	4.69	4.57	2.6%
	total waiting time	22.63	19.38	14.4%
Second session	total session time	37.43	34.63	7.5%
	maximal waiting time	5.57	3.81	31.6%
	total waiting time	20	15.56	22.2%
Total sessions	total session time	39.27	34.47	12.2%
	maximal waiting time	5.1	4.17	18.2%
	total waiting time	21.4	17.33	19.0%
First with agent	total session time	37.43	34.29	8.4%
	maximal waiting time	5.57	4.57	18.0%
	total waiting time	20	19.36	3.2%
First without agent	total session time	40.86	34.63	15.2%
	maximal waiting time	4.69	3.81	18.8%
	total waiting time	22.63	15.56	31.2%

Fig. 5. Time performance data (in minutes, decimal notation).

that the Novice-Advisor Phase (Phase 2) indeed improves the performance of the agent, and therefore the 3-stage process that was suggested is justified. However, the optimal conditions for switching from Phase 2 to Phase 3 still need to be determined. Another issue is the possibility that an operator attend to a larger number of clients simultaneously when having the agent's assistance. We performed experiments when attending 2 and 3 clients because this was seen to be a reasonable number (several views on this issue are presented in [40]). However, an operator might be able to attend more than 3 clients simultaneously by having an agent working alongside her. The feasibility of this option should be tested as well.

Note that this research study was designed to examine the effects of the assisting agent on the assisted operators. A differently designed experiment may explore the influence of the agent on the service from the clients' perspective.

References

1. Aktekin, T., Ekin, T.: Stochastic call center staffing with uncertain arrival, service and abandonment rates: A bayesian perspective. Naval Research Logistics (NRL) **63**(6), 460–478 (2016)
2. Argall, B.D., Chernova, S., Veloso, M., Browning, B.: A survey of robot learning from demonstration. Robotics and autonomous systems **57**(5), 469–483 (2009)
3. Aviv, A., Oshrat, Y., Assefa, S.A., Mustapha, T., Borrajo, D., Veloso, M., Kraus, S.: Advising agent for service-providing live-chat operators. arXiv preprint arXiv:2105.03986 (2021)
4. Azaria, A., Gal, Y., Kraus, S., Goldman, C.V.: Strategic advice provision in repeated human-agent interactions. Autonomous Agents and Multi-Agent Systems **30**(1), 4–29 (2016)
5. Azaria, A., Rabinovich, Z., Kraus, S., Goldman, C.V., Gal, Y.: Strategic advice provision in repeated human-agent interactions. In: Twenty-Sixth AAAI Conference on Artificial Intelligence (2012)
6. Azaria, A., Rosenfeld, A., Kraus, S., Goldman, C.V., Tsimhoni, O.: Advice provision for energy saving in automobile climate-control system. AI Magazine **36**(3), 61–72 (2015)
7. Bechara, A., Damasio, H., Damasio, A.R.: Emotion, decision making and the orbitofrontal cortex. Cerebral cortex **10**(3), 295–307 (2000)
8. Breiman, L.: Random forests. Machine learning **45**(1), 5–32 (2001)
9. Bruine de Bruin, W., Parker, A.M., Fischhoff, B.: Individual differences in adult decision-making competence. Journal of personality and social psychology **92**(5), 938 (2007)
10. Carroll, J.S., Bazerman, M.H., Maury, R.: Negotiator cognitions: A descriptive approach to negotiators' understanding of their opponents. Organizational Behavior and Human Decision Processes **41**(3), 352–370 (1988)
11. Cheetham, W.: Lessons learned using cbr for customer support. In: FLAIRS Conference. pp. 114–118 (2003)
12. Coussement, K., Van den Poel, D.: Improving customer attrition prediction by integrating emotions from client/company interaction emails and evaluating multiple classifiers. Expert Systems with Applications **36**(3), 6127–6134 (2009)
13. Crook, P.A., Marin, A.: Sequence to sequence modeling for user simulation in dialog systems. In: INTERSPEECH. pp. 1706–1710 (2017)
14. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding (2019)
15. Dietrich, C.: Decision making: Factors that influence decision making, heuristics used, and decision outcomes. Inquiries Journal **2**(02) (2010)
16. Elmalech, A., Sarne, D., Grosz, B.J.: Problem restructuring for better decision making in recurring decision situations. Autonomous Agents and Multi-Agent Systems **29**(1), 1–39 (2015)
17. Esposito, C., Ficco, M., Palmieri, F., Castiglione, A.: A knowledge-based platform for big data analytics based on publish/subscribe services and stream processing. Knowledge-Based Systems **79**, 3–17 (2015)
18. Floyd, M.W., Turner, J., Aha, D.W.: Using deep learning to automate feature modeling in learning by observation: a preliminary study. In: 2017 AAAI Spring Symposium Series (2017)
19. Ford, J.K., Schmitt, N., Schechtman, S.L., Hults, B.M., Doherty, M.L.: Process tracing methods: Contributions, problems, and neglected research questions. Organizational behavior and human decision processes **43**(1), 75–117 (1989)

20. Graef, R., Klier, M., Kluge, K., Zolitschka, J.F.: Human-machine collaboration in online customer service–a long-term feedback-based approach. Electronic Markets pp. 1–23 (2020)
21. Hansen, L.K., Salamon, P.: Neural network ensembles. IEEE transactions on pattern analysis and machine intelligence **12**(10), 993–1001 (1990)
22. Hart, S.G., Staveland, L.E., et al.: Development of nasa-tlx (task load index): Results of empirical and theoretical research **52**, 139–183 (1988)
23. Hathaway, B.: Data-driven studies of caller behavior under call center innovations (2019), https://doi.org/10.17615/7ckx-xq92s
24. Holman, D., Batt, R., Holtgrewe, U.: The global call center report: International perspectives on management and employment (2007), https://ecommons.cornell.edu/handle/1813/74325
25. Jitesh: Whatsapp customer support. https://www.wati.io/whatsapp-as-a-customer-support-channel/ (2020), online, Accessed: 04/07/2021
26. Juliusson, E.Á., Karlsson, N., Gärling, T.: Weighing the past and the future in decision making. European Journal of Cognitive Psychology **17**(4), 561–575 (2005)
27. Katal, A., Wazid, M., Goudar, R.H.: Big data: issues, challenges, tools and good practices. In: 2013 Sixth international conference on contemporary computing (IC3). pp. 404–409. IEEE (2013)
28. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.Y.: Lightgbm: A highly efficient gradient boosting decision tree. In: Advances in neural information processing systems. pp. 3146–3154 (2017)
29. Kenda, I.: Assessment of cognitive impairment generated by job strain. cas of call center teleoperators in kinshasales. (09 2017)
30. Koçağa, Y.L., Armony, M., Ward, A.R.: Staffing call centers with uncertain arrival rates and co-sourcing. Production and Operations Management **24**(7), 1101–1117 (2015)
31. Koole, G., Mandelbaum, A.: Queueing models of call centers: An introduction. Annals of Operations Research **113**(1-4), 41–59 (2002)
32. Kraus, S.: Human-agent decision-making: Combining theory and practice. arXiv preprint arXiv:1606.07514 (2016)
33. Lee, K., Jo, J., Kim, J., Kang, Y.: Can chatbots help reduce the workload of administrative officers?-implementing and deploying faq chatbot service in a university. In: International Conference on Human-Computer Interaction. pp. 348–354. Springer (2019)
34. Legros, B., Jouini, O.: On the scheduling of operations in a chat contact center. European Journal of Operational Research **274**(1), 303–316 (2019)
35. Levy, P., Sarne, D.: Intelligent advice provisioning for repeated interaction. In: Thirtieth AAAI Conference on Artificial Intelligence (2016)
36. Li, C.H., Yeh, S.F., Chang, T.J., Tsai, M.H., Chen, K., Chang, Y.J.: A conversation analysis of non-progress and coping strategies with a banking task-oriented chatbot. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems. pp. 1–12 (2020)
37. Liu, J., Gao, Z., Kang, Y., Jiang, Z., He, G., Sun, C., Liu, X., Lu, W.: Time to transfer: Predicting and evaluating machine-human chatting handoff. arXiv preprint arXiv:2012.07610 (2020)
38. Madotto, A., Cahyawijaya, S., Winata, G.I., Xu, Y., Liu, Z., Lin, Z., Fung, P.: Learning knowledge bases with parameters for task-oriented dialogue systems. arXiv preprint arXiv:2009.13656 (2020)

39. Majumdar, S., Tekiroglu, S.S., Guerini, M.: Generating challenge datasets for task-oriented conversational agents through self-play. arXiv preprint arXiv:1910.07357 (2019)
40. Mashburn, J., Rogers, A., Rogers, I., Cheung, a., Wallace, J.: how many customers can an agent handle? https://www.quora.com/During-a-customer-live-chat-about-how-many-customers-on-average-can-a-single-agent-handle-concurrently (2021), online, Accessed: 03/10/2021
41. Modak, A., Mupepi, M.G.: Dancing with whatsapp: Small businesses pirouetting with social media. In: Conference Proceedings by Track. vol. 51 (2017)
42. Nam, K., Lee, Z., Lee, B.G.: How internet has reshaped the user experience of banking service? KSII Transactions on Internet & Information Systems **10**(2) (2016)
43. Naneetha, R.: A new paradigm shift on how whatsapp empower small business to develop customer relationship and it becomes an integral part of business. Research Journal of Humanities and Social Sciences **9**(1), 119–124 (2018)
44. Nguyen, T.H., Yang, R., Azaria, A., Kraus, S., Tambe, M.: Analyzing the effectiveness of adversary modeling in security games. Twenty-Seventh AAAI Conference on Artificial Intelligence (2013)
45. Nuruzzaman, M., Hussain, O.K.: A survey on chatbot implementation in customer service industry through deep neural networks. In: 2018 IEEE 15th International Conference on e-Business Engineering (ICEBE). pp. 54–61. IEEE (2018)
46. Okuda, T., Shoda, S.: Ai-based chatbot service for financial industry. Fujitsu Scientific and Technical Journal **54**(2), 4–8 (2018)
47. Rosenfeld, A., Agmon, N., Maksimov, O., Kraus, S.: Intelligent agent supporting human–multi-robot team collaboration. Artificial Intelligence **252**, 211–231 (2017)
48. Rosenfeld, A., Keshet, J., Goldman, C.V., Kraus, S.: Online prediction of exponential decay time series with human-agent application. In: Proceedings of the Twenty-second European Conference on Artificial Intelligence. pp. 595–603. IOS Press (2016)
49. Rosenfeld, A., Kraus, S.: Predicting human decision-making: From prediction to action. Synthesis Lectures on Artificial Intelligence and Machine Learning **12**(1), 1–150 (2018)
50. Serban, I.V., Lowe, R., Henderson, P., Charlin, L., Pineau, J.: A survey of available corpora for building data-driven dialogue systems. arXiv preprint arXiv:1512.05742 (2015)
51. Ta, T., l'Ecuyer, P., Bastin, F.: Staffing optimization with chance constraints for emergency call centers. In: MOSIM 2016 - 11th International Conference on Modeling, Optimization and Simulation, Aug 2016, Montréal, Canada. hal-01399507f (2016)
52. Weizenbaum, J.: Eliza—a computer program for the study of natural language communication between man and machine. Commun. ACM **9**(1), 36–45 (Jan 1966). https://doi.org/10.1145/365153.365168, https://doi.org/10.1145/365153.365168
53. Williams, J.D., Young, S.: Partially observable markov decision processes for spoken dialog systems. Computer Speech & Language **21**(2), 393–422 (2007)
54. Wong, J., Hastings, L., Negy, K., Gonzalez, A.J., Ontañón, S., Lee, Y.C.: Machine learning from observation to detect abnormal driving behavior in humans. In: The Thirty-First International Flairs Conference (2018)