Improving the Scalability of MOOC Platforms with Automated, Dialogue-based Systems

Abstract— **In recent years e-learning environments got more and more popular, which resulted in a diverse, fast-growing community that among other anticipate individualisation and flexibility in means of customer support and recommendations. Scalability is required when the number of learners and the number of learning resources increase. This work targets dialogue-based systems used for Massive Open Online Course (MOOC) platforms to reduce human resources for answering technical support requests. The dialogue-based system must be available 24/7 and response immediately mainly to platform-specific frequently asked questions which were identified by an analysis of needs on existing support requests. Based on this comprehensive concepts, prototypes were implemented to prove the feasibility and explore the possibilities. The achievements of the deployment of a dialogue-based system for a MOOC platform were evaluated with well-defined metrics. These metrics showed, that even when deploying the most basic version of a dialogue-based system, the number of support requests created is reduced and thus the costly manual effort.**

Keywords—MOOCs, Chatbots, Dialogue-based system, Customer-Support-Chatbots, Language Processing, Machine Learning, Artificial Intelligence

I. INTRODUCTION

In recent years and with the change of technology to a web-based, smarter and more productive environment, learning needs and behaviour has evolved too. As demand for acquiring new skills in a rapid, personalised and flexible manner is increasing, so is the need for novel and inventive solutions [1]. MOOCs have emerged as a promising model for bringing together people interested in learning and making higher education freely available via the internet [2]. Although the introduction of MOOCs has, particularly for users, been overwhelmingly beneficial, challenges remain.

Some of these challenges can be found in the area of scalability. Scalability is required when the number of learners and the number of learning resources increase. Several separate aspects can be derived from that. First, increased user numbers demand higher capacities in assignment grading and the provision of individual evaluation. To manage this, the approach of peer assessment for MOOCs was introduced [3] [4].

A second aspect deals with the need for technical scalability due to the higher workload of a course, as it is available to a massive audience at once. These challenges can be tackled by using modern architecture which allows adding new machines on demand. Additionally, research has been conducted on using scalable cloud architecture to auto-scale computational power for the MOOC as the number of users increases [5].

This paper addresses another equally challenging aspect for scalability: support requests. With rising user numbers, many MOOC platforms face an immense increase in support requests which usually are answered manually. The limitations of that have become particularly apparent following a spike in user registrations as a result of COVID-19. It is essential to distinguish technical support requests from questions concerning learning content, which are discussed in forums and are not part of this research. We analysed the individual support requests, here on further referred to as tickets, of the MOOC platform [*platform name*], initially developed by [*institutional name*]. Our analysis concluded that many of the subjects of these tickets had already been answered in the frequently-asked-questions (FAQ) section of the platform.

To minimise manual effort and still provide user-friendly customer support, we implemented a dialogue-based system (chatbot), trained on a dataset containing the frequently-asked-questions of [*platform name*]. A chatbot is a conversational system that is able to interact with users in natural language. Existing research surrounding chatbots for educational purposes includes MOOCBuddy [6], the first educational chatbot related to MOOCs, which provides learners with personalised course recommendations and a chatbot to promote a learner's meaningful interaction in online courses [7]. With the help of our chatbot, we want to offer users technical support for our MOOC platform [*platform name*] and thus reduce the amount of customer support tickets that have to be answered manually.

We first describe the needs analysis of an automated dialogue-based system for [*platform name*]. In Section III, we provide an overview of the applications of artificial intelligent (AI) used to increase the efficiency of the conversational agents. The concept of the chatbot and its initial implementation is described in Section IV. Section V describes the deployment of the chatbot on [*platform name*] which is evaluated in Section VI. We conclude by giving a summary of this paper and discussing approaches to develop our research further.

II. ANALYSIS OF NEEDS

Following the outbreak of COVID-19 around the world, on one particular day, more than 9,000 new learners had registered on [*platform name*], resulting in a total of over 1.4 million users. In 2019 on average 50 tickets were created each month. Since February 2020 the number of monthly tickets increased to approximately 140 times its original number. We conducted a keyword-based search on frequently asked subjects of all tickets until May 2020 (about 21,000).

80% of these tickets were created with the language flag "en". It is important to note that even though English was selected as language of choice, some tickets were written in a different language, e.g. Spanish or Arabic. As visualized in Following the outbreak of COVID-19 around the world, on one particular day, more than 9,000 new learners had registered on [*platform name*], resulting in a total of over 1.4 million users. In 2019 on average 50 tickets were created each month. Since February 2020 the number of monthly tickets increased to approximately 140 times its original number. We conducted a keyword-based search on frequently asked subjects of all tickets until May 2020 (about 21,000)., most tickets by far were created containing questions on the topic *Certificate*, followed by tickets with questions concerning the topic *Exam* and *Email*. Further analysis of the Certificate-related tickets showed that more than 70% of the tickets' questions have already been answered in the FAQ-section but were answered by cost- and time-consuming manual work. These requests dealt with very similar questions such as *where to download the certificate*, *which certificates are available* or *why a certificate cannot be downloaded* – mainly because the user did not reach enough points in the course yet.

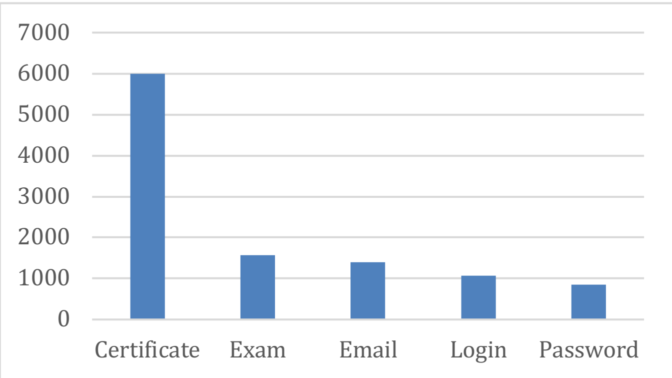

Figure 1 Most Common Subjects of Support Requests by Keyword-Based Search; Data Period: 02.2017 - 05.2020

III. Application of AI for Dialogue-based Systems

Simpler dialogue-based systems make use of decision trees or keyword recognition to create a conversation and mostly accept only a limited, predefined set of questions. Advanced and more intelligent dialogue-based systems use methods of artificial intelligence (AI), e.g. natural language processing (NLP) and machine learning (ML), to understand the meaning and keeping the context of the conversation allowing high-level communication.

A. Natural Language Processing

The concept of dialogue-based systems originates from traditional question-answering systems (QAS) and presents a good solution for queries by accessing unstructured and structured data in natural language [8]. With the help of NLP, an advanced dialogue-based system classifies a human written or spoken sentence into an intent. An intent defines the overall topic of a user's statement, the utterance. The sentence can also include entities, which are used to refine an intent.

B. Machine Learning

Advanced dialogue-based systems use a machine learning model to predict the subsequent action of the chatbot. The models are regularly trained on predefined training data. Actions can consist of requests to APIs and other servers and utterances that are returned to the user.

IV. Concept and Implementation

The main challenge for [*platform name*] is clearly the number of tickets created. A novel and efficient solution to this problem is the usage of a dialogue-based system. The system can be used as an alternative source of customer service, which is available 24/7 for answering questions in a dialogue-based conversation. Moreover, in case some questions cannot be answered with the help of the chatbot, it is still possible to create a ticket through the support form, which will then be answered manually. If the chatbot fulfils the necessary quality requirements for the interpretation of requests, i.e. a human-like experience as well as a professional appearance, users may trust the bot and may use it for their questions [9]. This provides a meaningful experience and in particular a high satisfaction for the user. In this initial concept, the chatbot does not have access to personal information and user data. Therefore, concerns about security and privacy may be low or even non-existent.

The dialogue-based system was implemented with the help of the open-source framework Rasa[1] and a custom user interface. Rasa consists of two main components: Rasa NLU and Rasa Core. Rasa NLU is an open-source language processing tool for intent classification of a conversation. As shown in Figure 2, a user might say to the chatbot: "Show me yesterday's financial news!". NLU extracts the entities yesterday and financial. Then an intent is matched, e.g. *showNews*. The intent, any entities that were found and the original message are converted into a dictionary and handed over to Rasa Core.

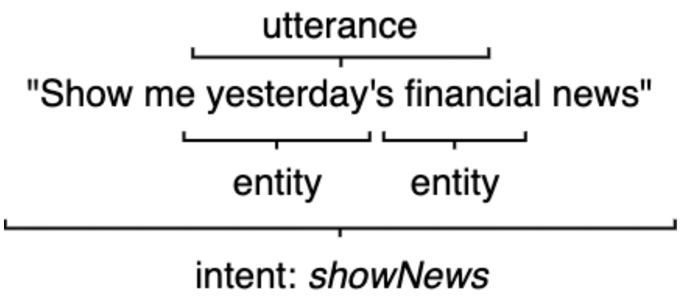

Figure 2 Structure of an Utterance in Combination with an Intent and Entities

Rasa Core is an open-source dialogue engine that predicts the next action with the help of policies. Many policies are already predefined within the Rasa framework and just have to be configured by, e.g. epochs, batch size or hidden layers sizes. As this paper concentrates on the application and integration of chatbots and not on the engine optimisation of chatbot frameworks, we will not go further into details here.

The chatbot for [*platform name*] is available in English. Answers to both general and technical queries concerning the courses can be obtained from the chatbot. As shown in the Analysis of Needs, most requests regard the topics certificate, exam and log in. Concerning the topic certificate, the chatbot

[1] https://rasa.com/

can answer questions about the difference between the certificate types, how to achieve the certificates and where to download them. Moreover, users can receive information on how to get started, where to log in and what to do when they forget their password. The domain for the chatbot of [*platform name*] also includes general requests about course exams and grading, language support and compatibility of course videos with different smartphones and operating systems. To enhance the experience of the conversation, the chatbot holds conversational intents. Conversational intents are phrases for greetings, saying thank you and for asking again. So far, 23 contextual dialogues and 20 domain-specific intents with approximately 15 variations for each intent have been defined.

The dialogue-based system is built to self-learn. Whenever a subject, which is not yet part of the domain, repeatedly arises as a question through the support form a new intent with this particular subject is added to the domain. Additionally, with the help of user feedback, new variations for an intent can be added. That means, if a user asks the chatbot a question and the NLU is not able to match one intent to that question, the chatbot will give the user the most probable intents to choose from. The user's question is then added to the set of variations of the selected intent. The variations are regularly cleared by a human agent to avoid bad data and incorrect answers.

V. DEPLOYMENT

For the integration of the chatbot in the user interface, two different variations were designed. Based on the results of an A/B/n testing the best version should be detected and used in the future. Version 1 was the first option to be implemented for [*platform name*]. In the following, Version 1 and Version 2 are described and compared to each other. Additionally, the first analysis of Version 1 was performed and evaluated. For the evaluation and a better comparison of the versions, two metrics were defined and are used in Section VI.

A. Version 1

The first version for the integration of the dialogue-based system to [*platform name*] is not a chatbot as such. The user interface shows the original support request form. As soon as a user has typed in the title of the issue suggestions, based on the title, are loaded. Once finished, an additional box appears in the form, which shows a possible answer to the issue. If the answer is sufficient, the user can close the form. Otherwise, a ticket can be created as before. A screenshot of the integration is shown in Figure 3.

Even though the user interface might not show a chatbot dialogue window, the backend uses the chatbot's logic. The title of the issue is used as the beginning of a conversation with the chatbot and therefore forwarded to Rasa NLU. As described before in Section III.A, the question is then analysed and matched to a possible intent. Next, Rasa Core predicts the possible response. The model of Version 1 only contains utterances in natural language. That means, no additional requests to APIs are performed. In this case, the utterance, which actually should be displayed as the chatbot's response, is displayed as a suggestion in the form. The context of the conversation is limited to one input (the title of the issue) and the corresponding response (suggestion). A change in the title of the issue will thus create a new conversational context. If an issue title cannot be matched to an intent, no suggestion is shown.

The suggestions of Version 1 are based on the standardised answers given by human agents when answering the tickets manually. Therefore, it is expected that users are satisfied with the response and close the form without creating a support request. Additionally, the users might feel that their question is valued more as they instantly get an almost individual suggestion to their problem. As a result, they can continue browsing and attending courses on [*platform name*] without waiting for an answer from the customer support team. As the domain is so far limited to general issues, the chatbot cannot answer specific questions concerning, for example, a particular course section, exam or requests with personal data. The chatbot, however, will provide a generic answer that might help the user to solve the problem on their own. For Version 1, conversational intents were removed from the chatbot's domain as they are not needed and might create confusion among the users.

This version of the dialogue-based system integration supports self-learning in a limited way. As the chatbot does not receive feedback about the accuracy of the suggestions given, manual effort is needed. That means that tickets are selected on a sample basis, and the conversation is evaluated. Based on the results, new intents and variations can then be added to the chatbot's training data.

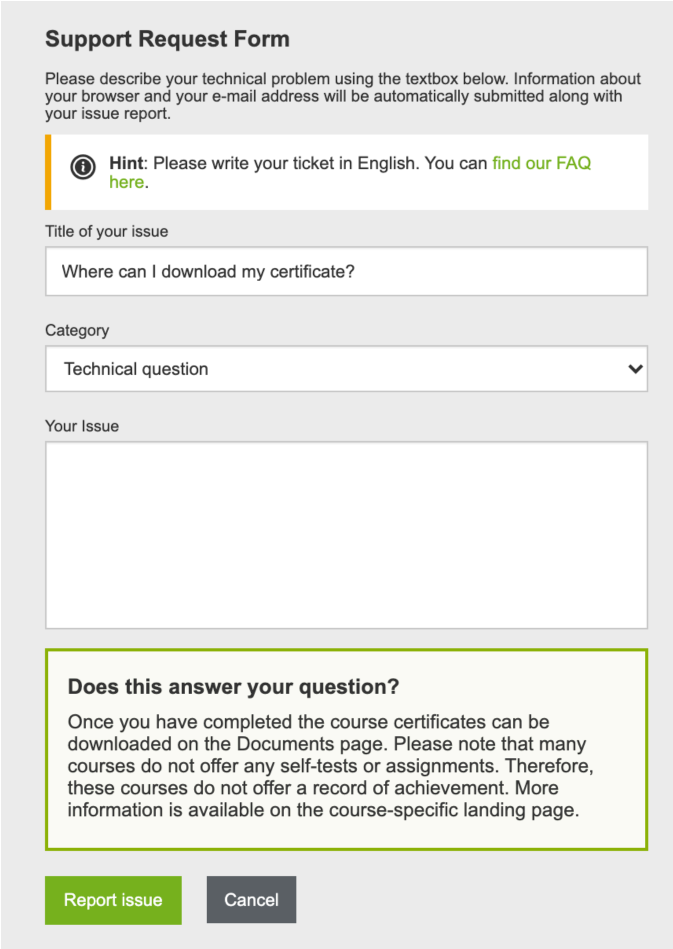

Figure 3 Integration of Version 1 into [platform name]'s UI

B. Version 2

Version 2 represents a dialogue-based system in the traditional conversational way. The user interface of *[platform name*] is modified as described in the following. At the head of the support request form, an additional switch was added. The user is now able to select between the original form view to create a ticket or switch to the chatbot view and have a conversation with it. The default view shows the chatbot to encourage the user to explore and test the dialogue-based system. As with Version 1, if the users are satisfied with the outcome of the conversation, they can close the form without creating a ticket. However, if the answers provided by the chatbot are not precise or helpful enough, the user has the option to switch to the support request form to request manual support.

Additionally, users can more easily evaluate the answer of the chatbot and give feedback based on their experience and the chatbot's helpfulness. Therefore, a simple 5-star rating is shown to users at the end of a conversation. As part of the chatbot's self-learning ability, the feedback is evaluated, and, on that basis, the domain is updated. Compared to Version 1, the dialogue-based system of Version 2 suggests several topics if a question is not understood. The user can then select one of the topics or try to rephrase the question. The initial question of the user is then manually added to the selected topic's set of variations to further enhance the chatbot's knowledge.

In comparison, Version 1 is a very slim implementation of the initial concept. Some important chatbot functionalities are left out. These include the possibility for the chatbot as well as the user to ask again if something was unclear. Additionally, having a conversation with a chatbot results in a human-like customer support experience. In Version 1, this experience is lost as well. On the other hand, with Version 2, the user can easily avoid the chatbot by switching to the support request form. Version 1 always gives the user a suggestion without any additional effort on the user's end. Nevertheless, to select the most efficient version, it is best to evaluate the outcome of both versions when deployed. In the following, an analysis of the first findings of Version 1 is presented.

VI. EVALUATION

With the deployment of the first version for the dialogue-based system for [*platform name*], four events have also been implemented for evaluating the usage: *Form open, Chatbot replied, Ticket created* and *Form closed*. If users are satisfied with the suggestion given by the chatbot and they close the form, no *Ticket created* event is saved. The events do not contain any personal user data but can be associated with a session identification number. It is possible that one session contains several *Chatbot replied* events as the title of the issue can be changed; every other event can only exist once. Contextual user data such as language, platform and date of the event are saved with every event. Additionally, the chatbot replied event contains the question, the response and the success status of the request. When a ticket is created, the title, body and category of the issue are stored as well.

The events have been tracked between 09th of April 2020 and 24th of May 2020. Data between 24th and 27th of April was cleared from the evaluation because the support request system was unavailable during that time. As the chatbot is so far only available in the browser, data was filtered on the platform type Windows, Mac and Linux. All other events from mobile platforms like Android, have been excluded.

On an average, approximately 570 *Form open*, 70 *Chatbot replied* and 35 *Ticket created* events have been tracked daily. About three weeks after the deployment, on 27th of April, an update of the chatbot's default answers was conducted. Before that, on average, 75% of all users that received suggestions from the dialogue-based system still created a ticket afterwards. However, after providing clearer and simpler answers, more users were satisfied with the suggestions given and closed the form without creating a ticket. The time curve of this development is illustrated in Figure 4.

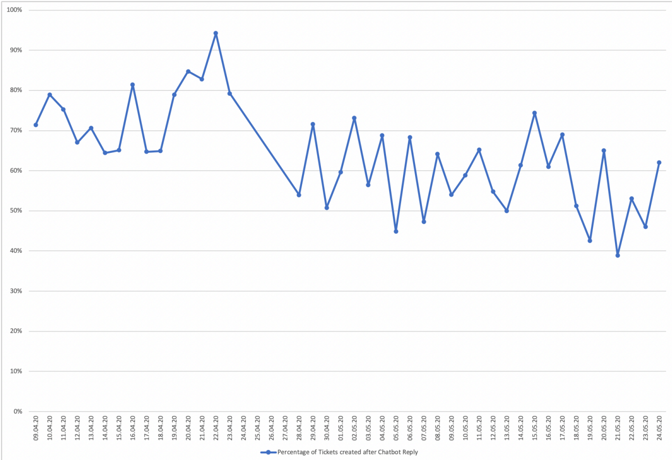

Figure 4 Percentage of Users that Create a Ticket after a Successful Chatbot Response; Data Period: 09.04. - 24.05.2020

The grey area of Figure 5 represents the potential of this integration. It is calculated by the differences between the number of chatbot replies and the number of support tickets created. The bigger this area is, the greater are the savings of human resources. Since the update of the utterances on the 27th of April, the average potential (delta) increased from 25% to 42%.

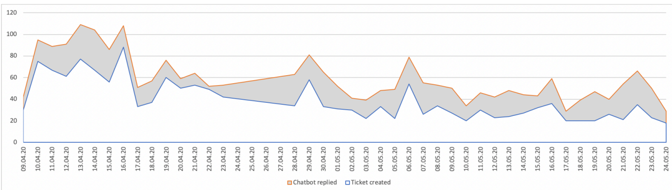

Figure 5 Delta - the Potential of Version 1; Data Period: 09.04. - 24.05.2020

In summary, Version 1 seems to be a promising first solution for reducing the number of tickets. Nonetheless, these tickets are continuously evaluated and, based on the outcome, the domain of the dialogue-based system will be enhanced. It is anticipated that with subsequent improvements in suggestions, the rate of users that still create tickets after receiving suggestions will strongly decrease. Additionally, newly emerging, frequently asked questions will be added to the chatbot's knowledge.

VII. Conclusion and Future Work

Reducing the human resources of scaling systems required for replying to technical support requests was the goal of this paper. This work provided well-defined concepts and metrics for automated, dialogue-based systems deployed on a specific scaling system. The basic concept describes a chatbot which provides a 24/7 automated customer support and mainly answers platform-specific FAQs. The first metric describes the number of users who were not satisfied with the suggestion of the chatbot and still created a support request. The second metric calculates the current potential of the deployment by deducting the numbers of created tickets from the number of chatbot replies. The evaluation showed that with the deployment of chatbots, the number of support requests, and thus the manual effort needed, was reduced.

Leading on from this, we plan to make the chatbot more personalized, allowing the system to access user data to give individual recommendations and to fulfil user requests for profile changes. We are in the process of improving the current proof of concept by further analysing the domain and adding new intents as well as enhancing the UI/UX experience. Additional chatbots are rolled out to other platforms in different IT-related domains to find out if there are differences in their efficiency based on the platform's nature and audience.

References

[1] A. McAuley, B. Stewart, G. Siemens and D. Cormier, The MOOC model for digital practice, Massive Open Online Courses: digital ways of knowing and learning~…, 2010.

[2] A. Adams, T. Liyanagunawardena and S. Williams, MOOCs: a Systematic Study of the Published Literature 2008-2012, International Review of Research in Open and Distance Learning, 2013.

[3] T. Staubitz, D. Petrick, M. Bauer, J. Renz and C. Meinel, Improving the Peer Assessment Experience on MOOC Platforms, Association for Computing Machinery, 2016.

[4] N. Sonwalkar, "The first adaptive MOOC: A case study on pedagogy framework and scalable cloud Architecture—Part I," in *MOOCs Forum*, 2013, pp. 22-29.

[5] A. Bouziane, D. Bouchiha, N. Doumi and M. Malki, "Question answering systems: survey and trends," *Procedia Computer Science,* no. 73, pp. 366-375, 2015.

[6] A. Iftene and J. Vanderdonckt, "Moocbuddy: a chatbot for personalized learning with moocs," in *RoCHI–International Conference on Human-Computer Interaction*, 2016, p. 91.

[7] D. Song, E. Y. Oh and M. Rice, "Interacting with a conversational agent system for educational purposes in online courses," in *2017 10th International Conference on Human System Interactions (HSI)*, 2017, pp. 78-82.

[8] A. FØLSTAD, C. B. NORDHEIM and C. A. BJØRKLI, "What makes users trust a chatbot for customer service? An exploratory interview study," in *International Conference on Internet Science*, Springer, 2018, pp. 194-208.

[9] H. K. Suen, "Peer assessment for massive open online courses (MOOCs)," *International Review of Research in Open and Distributed Learning,* vol. 3, no. 15, pp. 312-327, 2014.