Empowering University-level Help Desk for International Applicants with AI Chatbots

Empowering University-level Help Desk for International Applicants with AI Chatbots

Zahra Karimi, Giuseppe Gallipoli, Luca Cagliero, Francesca Chicco, Carola Rosa, Alessandra Sechi
Politecnico di Torino
Turin, Italy
{name}.{surname}@polito.it

***Abstract*—In this paper, we present the design, development, and testing of AI chatbots to support the help desk service provided by the Recruitment and Admissions Unit at Politecnico di Torino, a technical university located in the north-west of Italy. We explore the use of data-driven, AI-based conversational agents providing targeted responses to applicants' queries based on both the past student-office interactions through a ticketing system and a collection of Frequently Asked Questions (FAQs). With an ever-increasing number of requests from international applicants (20k+ requests from 100+ countries since January 2024), the adoption of AI-based solutions allows a significant reduction of the average waiting time per request, fostering the application, enrollment, and integration of foreign students coming from a variety of different countries. We develop separate chatbot systems to handle FAQs and manage inquiries submitted through the university's ticketing system. We explore the use of intent-based and generative approaches as well as a combination of the two. Our findings indicate that the intent-based approach excels in handling FAQs and well-defined tickets, whereas the generative-only strategy is more suitable for open-ended requests.**

***Index Terms*—Education, Conversational AI, Chatbots, Large Language Models**

I. Introduction

Recruiting and admission are crucial steps of the university-level student career. The help desk services of the university are often the very first point of contact between prospective students from all over the world and the hosting universities. Ensuring a high quality of service not only increases the admission and enrollment rates but also guarantees the maximal inclusivity of the offered higher level education towards different countries, races, and ethnic groups [1], [2].

The Recruitment and Admissions Unit at Politecnico di Torino, a technical university located in the north-west of Italy, manages prospective students from both Italy and around the world. These students are eager to learn about Politecnico di Torino's academic offer and admission procedures. Despite the comprehensive information available on the university website[1], students often seek more interaction and turn to the ticketing system for even general inquiries.

The office has been experiencing an increasing volume of requests, making it challenging to respond promptly. For context, from January 1st, 2024, to October 18th, 2024 the office received 18,102 tickets[2] and provided 16,794 initial replies and 26,400 total replies (including multiple replies to the same ticket from both operator and student). The average waiting time for the first reply is 4.7 days. Additionally, the office handles various topics, including (1) Bachelor prospective students with Italian degrees, (2) Bachelor prospective students with non-Italian degrees, (3) Master prospective students with non-Italian degrees, (4) Credential evaluation, (5) Credit transfer, (6) Recognition of foreign degrees, and (7) Single courses.

Regarding international applicants, for the 2024/25 academic year, the office received 7677 applications for the Master of Science level from 107 countries and 15425 applications for the Bachelor of Science level from 130 countries. The office also received 29960 Bachelor of Science applications from students with Italian high school diploma. These numbers refer to the applications received (multiple courses per applicant), not the number of applicants who finalized their enrollment.

Due to the high volume of applicants, the office required a tool to provide responses to general questions. These responses are typically available in the FAQ system, which leads to the opening of tickets. However, it was found that students prefer to ask their questions directly, regardless of the answers provided in the FAQ system.

Artifical Intelligence (AI) chatbots are autonomous agents that are capable of maintaining a conversation with a user in natural language, mimicking human-to-human interactions. With the recent advances of Generative AI and Large Language Models, chatbot systems have further empowered their capabilities to handle complex questions on broader topics. As discussed in Section II, the role of AI chatbots in education is already established [3]. For these reasons, Politecnico di Torino is seeking automated, AI-based solutions to effectively and efficiently process students' inquiries.

In this paper, we describe the design, development, and testing of chatbots based on AI to support the help desk service at Politecnico di Torino. We believe that the usage of AI-based conversational agents, in conjunction to traditional front-end and back-office services, could be a solution for both students and Recruitment and Admissions Unit employees. On the one hand, exposing a chatbot service allows students to ask questions directly to the chatbot, which would provide

[1]https://www.polito.it/en/education/international-students/exchange-students/admission-procedure/applypolito – Latest access: November 2024

[2]This number refers to the first ticket opened thus disregards the subsequent replies.

the correct answers or, for more detailed inquiries, redirect the student to the ticketing system. The chatbot would act as a filter that reduces the volume of requests and allows the university offices to better manage the workflow. On the other hand, providing office employees with AI-empowered services for semi-automatic generation of ticket replies allows the office to handle inquiries more efficiently.

We design, implement, and compare different types of chatbot systems relying on intent detection only, generative AI, and a mix of the above. The experimental results, achieved on historical data, show that the use of intent-based approach excels in handling structured content like FAQs or more standard ticket requests, whereas generative AI turns out to be more suited to less standardized, open-ended inquiries. The high complexity and variety of ticket messages also highlight new challenges and leave room for relevant future extensions.

II. Related Works

Previous research on chatbot sytems in education encompasses the use of deep learning techniques. For example, in [4] the authors employ the Rasa framework to develop a chatbot system providing daily updates of curriculum, admission for new students, tuition fees, and IELTS tests' outcomes. In [5] a chatbot integrating the classical Named Entity Recognition, intent detection, and Question Answering about finance has been proposed. The authors in [6] present a preliminary attempt to integrate FAQs, where bot-students' interactions are mainly rule-based. A similar approach has been adopted in [7], [8], where the goal is to reduce the workload of university employees. To the best of our knowledge, the latest work addressing a performance comparison between chatbot platforms used by online students has been presented in [9]. However, the authors explored the use of FAQs only for intent detection and AI model training. Conversely, we consider both FAQs and ticket requests and try to combine classical intent-based and generative approaches together. Furthermore, we leverage T5 prefix-based training to potentially improve performance by providing the model with additional context instead of only the user's question.

III. Dataset overview

The Recruitment and Admissions Unit extracted data from its IT system, resulting in two distinct datasets, one containing FAQs and the other comprising ticket logs. In the following, we provide a description of each dataset, presenting their structure and content.

A. FAQ dataset

It contains Frequently Asked Questions and their corresponding answers, totaling 260 question-answer (QA) pairs. The content is predominantly in English, with approximately 10% in Italian. In addition to a unique identifier, each QA is characterized by the following attributes: *subject*, which specifies whether the question refers to admission to BSc or MSc degree programs, and whether the candidate has a foreign or Italian previous qualification; *sub-subject*, which indicates the topic of the question (e.g., "How to apply", "Deadlines"); *question*, representing a commonly asked question by users; and *answer*, which provides a concise and informative response, potentially including hyperlinks.

Example of FAQ: "*If I don't have a language certificate as required, may I apply for a Master of Science program?*"
Answer: "*No, it is not possible: you have to upload the required language certification, otherwise your application will not be considered eligible. See [URL] for further details.*"

B. Ticket dataset

It spans a one-year time frame of one year and contains tickets submitted by prospective students, along with corresponding responses from operators. It contains a total of 43,198 tickets, of which 34,459 (approximately 80%) are in English, while the remaining 8,738 (around 20%) are in Italian. The dataset includes both single-turn conversations, consisting of a single QA pair, and multi-turn conversations, which include multiple exchanges (e.g., Q-A-Q-A). Each ticket is assigned a unique identifier and, similar to the FAQ data, is characterized by a *subject* and a *sub-subject* for more precise categorization. Additionally, tickets present the following attributes: *title*, which represents the subject of the request provided by the user; *date*, encoding the timestamp when the ticket was created, enabling chronological tracking; *question* and *answer*, which contain the content of the user's query and the operator's answer; and *operator_response*, which is a binary indicator (1 if the operator responded, 0 if the user replied) that tracks the interaction between the user and the operator.

After the pre-processing steps described in IV-A, we analyzed the distribution of the number of turns per ticket. We found that the most frequent case involves a single turn, with 8,084 tickets. However, conversations with two to five turns are also common, ranging from 6,240 tickets (for two turns) to 575 tickets (for five turns). In total, the dataset contains 8,084 single-turn conversations and 11,068 multi-turn conversations, highlighting the varying levels of interaction across different inquiries. The average number of tokens per question and answer is 86.2±71.1 and 45.6±42.9, respectively. We also analyzed the distribution of tickets across subjects and sub-subjects to identify the most common topics of inquiry among users. Specifically, the most frequent subjects are "Access to Bachelor's degree" and "Access to Master's degree", while the most common sub-subjects are "Registration", "External student access", and "Apply". This information is crucial for the Recruitment and Admissions Unit as it provides insights into users' primary concerns and highlights areas that may require additional attention or resources to improve service delivery.

Example of ticket: "*Dear office, my name is [NAME] a student from [LOCATION], my id is [STUDENT_ID] and i have been eligible to study master in civil engineering. as mentioned on your website eligibility letter will be available in main section but still its not available. kindly update it at earliest so i can proceed with pre-enrolment. regards*"

Answer: "*Dear applicant, congrats! Confirmation of eligibility letters will start to be sent soon (from April on) so you have to wait a little while still. Please check this page [URL] as there are all the steps to follow for your pre-enrolment. Kind regards.*"

IV. Method

In this section, we provide a detailed description of our approach. Specifically, following the pre-processing of both datasets, we implemented two ad hoc chatbot systems: one for handling FAQs and another dedicated to ticket messages.

A. Pre-processing

We perform pre-processing operations on both datasets to clean the data, enhance its quality and enable more effective use for subsequent analysis and model training. As the pre-processing steps differ between the two datasets, we discuss them separately.

a) FAQ pre-processing: After removing duplicate entries, we identify FAQs written in Italian using a language detection tool and translate these QA pairs into English using a pre-trained machine translation model. This step ensures that all content is in English, and we manually verify the correctness of the translation outputs. We also remove extra spaces and special characters, standardizing URLs and HTML tags within answers to improve text readability and consistency. To increase the variety of the dataset, and thus enhance the chatbot generalizability, we augment the questions by generating alternative reformulations using a pre-trained text paraphrasing model. For each QA pair, we generate alternative reformulations of the question, associating them with the original answer. The augmented dataset includes approximately 1,500 data samples.

b) Ticket pre-processing: We first clean the dataset by removing empty questions, duplicate messages, automatic responses (e.g., messages that inform users of potential delays during busy periods, such as the enrollment period) and status messages (e.g., notifications indicating that a ticket has been marked as closed by operators or students). For each ticket, we detect the language of each field (e.g., title, question, answer) and translate it into English if necessary. We clean both the original and translated texts by removing irrelevant or special characters and manually reviewed a subset of the translated outputs to verify the quality of the translations. Subsequently, we merge and concatenate messages associated with the same ticket, i.e., multi-turn conversations, with the goal of having a single QA pair per ticket. Lastly, since ticket messages often contains personal data, we apply anonymization techniques using the Presidio [10] tool. Specifically, it provides modules for identifying and anonymizing text entities including proper nouns, email addresses, and passport and phone numbers. Additionally, we defined custom patterns to handle Italian SSNs and prospective student usernames and passwords. After the data cleaning and merging steps, the final ticket dataset contains 19,152 records.

B. Chatbot for FAQs

At the beginning of the conversation, the system collects user information by asking three questions using input buttons: admission status (i.e., already admitted or not), degree program of interest (i.e., Bachelor or Master), and previous qualification (i.e., Italian or non-Italian). This information is stored in *slots*, which serve to retain user details in the long-term memory of the conversation. The use of slots is especially crucial for disambiguating questions that do not explicitly include user information but whose answers depend on it. As an example, if a user inquiries about enrollment deadlines, the system lacks information whether the user is interested in deadlines for a Bachelor's or Master's program. By leveraging slots, the chatbot can effectively disambiguate the correct user intent and provide the appropriate response.

To analyze the differences and effectiveness of various solutions, we implement three approaches: intent-based, generative-only, and a combination of the two. In all cases, we rely on the Rasa open-source framework [11], [12] as the interface with the user. However, each method has unique characteristics: the intent-based solution leverages the complete Rasa pipeline, the generative-only method employs a language model to answer questions, and the hybrid approach combines both strategies to provide a more adaptable and robust solution. In the following we describe the three approaches in more detail.

Intent-based: After the user submit a request, it is processed by the Rasa Natural Language Understanding (NLU) module. After creating lexical and syntactic feature, by leveraging both regular expressions and the Transformer-based DIET classifier [13], it simultaneously performs entity recognition and intent classification. The result is then sent to the Tracker component which stores the conversation history in memory. The next action to perform is determined by the Policy component which utilizes both a rule-based strategy and the Transformer-based TED model [14]. Finally, the Action component executes the action predicted in the previous step, by also accessing the conversation state stored by the Tracker, to generate an output which is then returned to the user.

To correctly associate intents (i.e., user questions) with the corresponding actions (i.e., answers), we manually organize the training data into Rasa NLU and domain files, categorizing them (if needed) based on user information. To handle both unclear and out-of-scope user queries, we define two fallback responses: one prompts the user to reformulate the question, while the other informs the user that the current request cannot be handled.

Generative-only: Unlike the intent-based method which utilizes the complete Rasa pipeline to process and respond to user queries, the generative-only approach uses Rasa solely as an interface to interact with the user. Both the understanding of the user's request and the generation of responses are handled by the Transformer-based T5 model [15]. The goal is to leverage the contextual understanding capabilities of language models to enhance both question comprehension and response

generation, thus improving generalizability. We specialize the T5 model's knowledge by fine-tuning it on the FAQ dataset, using the question as input and the corresponding answer as the expected output. We implement two variants of this approach: in the first, only the question is provided as input; in the second, by leveraging the T5's task-specific prefix strategy, we use the ticket *subject* as a prefix followed by the user's actual question.

Since some of the QA pairs contain hyperlinks in their answers, and generative models may struggle or hallucinate when generating hyperlinks, we mask them in the training set using `[URL]` placeholders. To correctly replace these placeholders in the generated answers, we post-process the output text before returning it to the user. Specifically, we first use the Rasa NLU module to retrieve the most relevant hyperlinks for the answer. If any placeholders remain, we fill them with hyperlinks from the most similar answer in the dataset, retrieved using a tf-idf-based similarity search.

Hybrid: In this approach we combine both the intent-based and generative-only methods. Specifically, because generative models may produce inaccurate or malformed results, we primarily rely on the Rasa pipeline, resorting to the generative approach only when necessary. The goal is to leverage the superior understanding capabilities of language models while ensuring at the same time the accuracy of the chatbot's responses. A sketch of the hybrid approach is displayed in Figure 1. The input query is initially processed by the Rasa NLU module, which identifies the user's intent. However, if the predicted intent's confidence score falls below a configurable threshold, we do not execute the associated action. Instead, the input question is processed using the T5 model, and the generated answer is returned after post-processing, as described previously. By adjusting the threshold, it is possible to balance the use of the Rasa pipeline and the generative model, prioritizing higher reliability of answers (with lower threshold values) or better generalization (with higher threshold values).

C. Chatbot for Tickets

Given the much higher variety of requests submitted through tickets, we utilize the T5 model to develop a chatbot system capable of handling ticket messages. This is done because manually categorizing each request to associate it with a corresponding FAQ (if any) or custom answer would be infeasible. Furthermore, by using a generative model, we aim to leverage the superior generalization capabilities of language models, potentially allowing the system to handle more complex requests that may require a tailored response, therefore not necessarily present in the set of FAQs.

To this purpose, we fine-tune a T5 model on the ticket dataset to specialize the model's knowledge on the ticket domain. Similar to the generative-only approach adopted for the FAQ chatbot, we explore multiple prefixes during the training of the T5 model. The goal is to provide contextual information to the model, identifying the most effective prefix that could enhance its performance in handling ticket data. We consider the following information or metadata as possible prefixes: *title*, to allow the model to focus on the user's specific query as expressed in their own words; *subject*, to provide the model with the main category of the ticket, helping it to contextualize the question more effectively within a specific area of concern; *sub-subject*, to further refine the context of the request, providing the model with a more detailed understanding of the specific topic of the ticket; and *date interval*, to inform the model about the specific academic period or seasonal context in which the ticket was submitted.

Using time intervals can help the model better understand the temporal relevance of queries, especially those that are time-sensitive, such as questions about deadlines or semester start dates. We manually define time intervals with the help of Recruitment and Admissions Unit operators, considering intervals such as the beginning of the first/second semester, the first/second semester class periods, winter/summer/autumn examination sessions, and summer holidays. In this second chatbot, the Rasa framework serves only as an interface with users, providing access to the fine-tuned T5 model.

V. Experiments

In this section, we describe the evaluation metrics and experimental settings used in this work. We also present and discuss the main results obtained on both the FAQ and ticket datasets, including qualitative feedback provided by the operators of the Recruitment and Admissions Unit.

A. Evaluation metrics

To evaluate the performance of the implemented chatbot systems, we use both syntax-oriented and semantic-oriented metrics. Specifically, we employ the **ROUGE-1/2/L** (R1/2/L) F1-score [16], which measures the syntactic overlap in terms of unigrams, bigrams, and longest common subsequences between the chatbot's response and the corresponding gold reference. For semantic evaluation, we utilize the **BERTScore** (BS) F1-score [17], which leverages a BERT-based model to assess the semantic similarity between the chatbot's answer and the reference response.

B. Experimental settings

To implement the chatbot systems, we utilize the Rasa open-source framework and Hugging Face to access pre-trained model checkpoints.

For the Rasa NLU module, we use the default configuration to extract lexical and syntactic feature. We train the DIET classifier for named entity recognition and intent classification for 100 epochs. To handle unclear and out-of-scope questions, we set the fallback classifier thresholds to 0.1 and 0.7, respectively. For the Policy component, we set the maximum history length of the conversation to 7, which specifies how many dialogue turns are considered when determining the next action. As policies, we employ both the rule-based strategy and the TED model trained for 100 epochs. We fill slots (i.e., admission status, degree program of interest, and previous qualification) with user information collected at the beginning

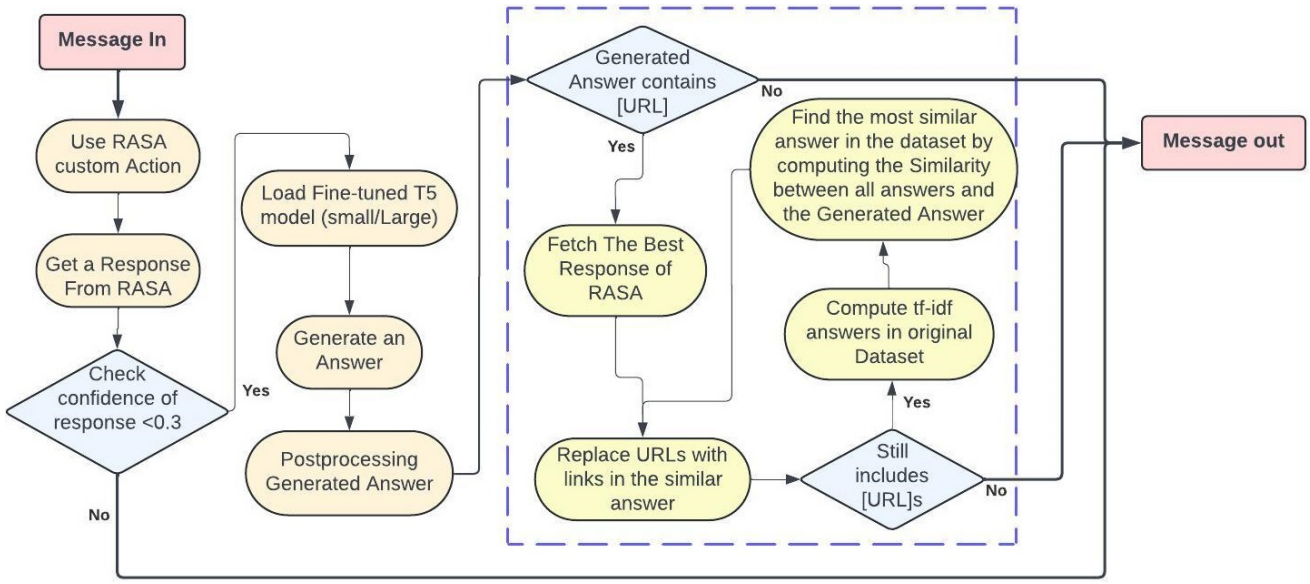

Fig. 1. Chatbot for FAQs – Hybrid approach.

of the conversation using questions and predefined answer options provided through input buttons.

To translate Italian FAQs or ticket messages into English, we use the `Helsinki-NLP/opus-mt-it-en` machine translation model. We do not analyze how much this translation step influences the results. However, as we employ state-of-the-art Transformer-based models and the proportion of Italian samples is limited (i.e., nearly 15%), we believe it does not significantly affect the results. We augment the initial FAQ dataset with alternative question formulations using the paraphrasing model `tuner007/pegasus_paraphrase`. For each source question, we generate up to five additional formulations and to improve their diversity, we set the model's temperature to 5 and the number of beams for generation to 10.

For both the FAQ and ticket dataset, we consider the T5 language model in both small and large versions. We first identify the optimal hyperparameter configuration using the small version and then test the same configuration with the large version. We set the learning rate to $\left[10^{-4}, 5 \cdot 10^{-4}\right]$ and train the model for 10 epochs. When training the model on the FAQ data, we split the dataset into training (90%) and validation (10%) sets, selecting the best model checkpoint based on the performance in ROUGE-1 on the validation set. To evaluate the performance of the three approaches implemented for handling FAQs, we prepare a test set consisting of 225 samples. The test set includes noisy and manually paraphrased questions that do not overlap with the training set, as well as out-of-scope questions to assess the robustness of each approach. Based on preliminary experiments, we set the confidence threshold for the hybrid approach of the FAQ chatbot to 0.3.

When training the model on the ticket data, we adopt a hold-out approach, splitting the dataset into training (80%) and test (20%) sets. To further enrich the test set, we paraphrase a portion of existing tickets and manually creating additional tickets to simulate a broader range of queries. To ensure the accuracy and relevance of the model's performance, we manually analyze the results by simulating real-world interactions. This allows for a comprehensive evaluation of the chatbot's ability to respond to varied queries, thus simulating a real-world scenario.

TABLE I
GENERATIVE-ONLY RESULTS ON THE FAQ VALIDATION SET USING T5.

dataset	model	input	LR	R1	R2	RL	BS
original	T5 small	question	1e-4	28.5	27.4	28.1	75.6
		question	5e-4	38.9	38.0	38.6	79.1
		subject: question	5e-4	38.6	37.5	38.3	79.1
expanded	T5 small	question	1e-4	46.1	45.1	45.7	80.2
		question	5e-4	91.0	89.9	90.7	97.4
		subject: question	5e-4	95.6	94.5	95.2	99.2
	T5 large	subject: question	5e-4	**98.5**	**98.1**	**98.4**	**99.8**

TABLE II
RESULTS OF THE FAQ CHATBOT USING THE THREE APPROACHES.

model	approach	R1	R2	RL	BS
–	intent-based	**90.6**	**88.6**	**89.9**	**97.9**
T5 small	generative-only	73.7	69.8	71.8	92.8
	hybrid	79.2	74.4	77.4	93.8
T5 large	generative-only	81.8	78.7	80.5	94.6
	hybrid	87.9	85.3	86.9	95.7

C. Results on FAQ dataset

Table I presents the results achieved by the T5 model in the generative-only approach for the FAQ chatbot. We evaluate the performance on the validation set for varying model sizes and inputs, comparing results using both the original dataset and the expanded dataset with paraphrased questions.

Expanding the training set significantly improves performance, highlighting the need of increasing the dataset size given the limited number of samples of the original dataset. Specifically, T5 small's performance when only the question is provided as input improves from 38.9 to 91.0 in ROUGE-1 and from 79.1 to 97.4 in BERTScore. When additional context (i.e., the subject of the ticket) is included as a prefix, no performance improvement is observed on the original dataset. This is likely due to the limited number of samples, which prevents the model from effectively leveraging the additional information. However, considering the expanded dataset, adding the subject subject as a prefix leads to further improvements of +4.6 and +1.8 points in ROUGE-1 and BERTScore, respectively. After determining the best hyperparameter configuration and model input, we use these settings to train and evaluate the performance of T5 large. Using the larger model further enhances performance, achieving scores of 98.5 in ROUGE-1 and 99.8 in BERTScore.

Table II presents the results on the test set for the three approaches of the FAQ chatbot we implemented. The intent-based approach, utilizing the full Rasa pipeline, achieves the highest performance across all metrics. While there is an improvement in the generative-only approach when moving from T5 small to the large version, it still underperforms compared to the intent-based method (i.e., 90.6 vs. 81.8 in ROUGE-1). This difference in performance is likely because, within the context of FAQ data, an intent-based approach that leverages multiple components (e.g., feature extractors, entity and intent classifiers, policy manager) can better understand user queries, leading to more accurate and relevant predefined responses. In contrast, although a generative-only approach may offer better generalization capabilities, relying on a single model for both intent recognition and response generation leads to lower results. Furthermore, generative models are prone to common issues in text generation such as generating inaccurate content. This risk is not present with the intent-based approach, where the responses are always selected from a predefined set, ensuring the returned content is free of hallucinations or inaccuracies even if the intent is misclassified.

Integrating the intent-based and generative-only strategies into the hybrid approach demonstrates promising results, improving the performance of the generative-only method (i.e., 81.8 vs. 87.9 ROUGE-1 for T5 large). This indicates that employing the generative model selectively (i.e., only when the confidence in intent prediction is low) enables better and more robust control over the chatbot's outputs, reducing the likelihood of inaccurate or incomplete responses. However, the best-performing approach is the intent-based strategy, which fully relies on the Rasa pipeline for both understanding and responding to user queries, achieving better results than the hybrid method (i.e., 90.6 vs. 87.9 in ROUGE-1). This suggests that, especially for more structured and self-contained inquiries, as in the case of FAQs, leveraging the intent-based approach managed by Rasa yields better results compared to the generative-only method, which struggles to achieve the same level of accuracy in the generated answers.

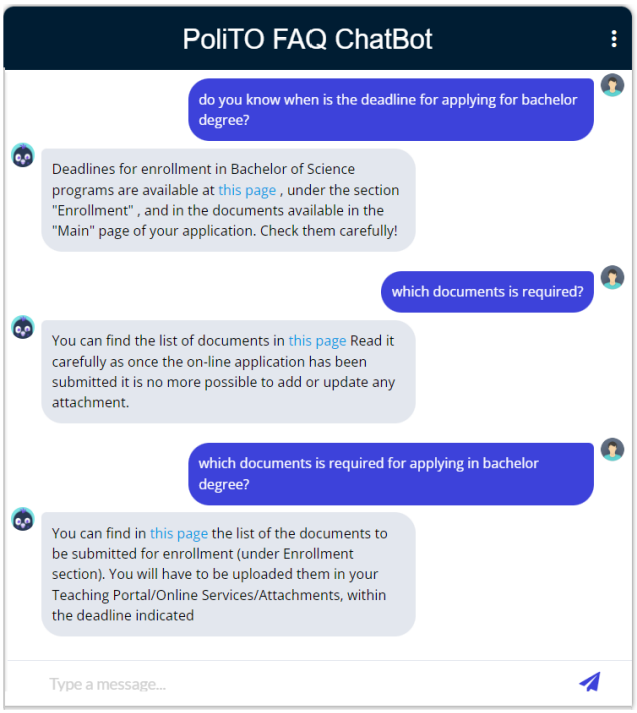

Fig. 2. Chatbot for FAQs – Example of interaction.

TABLE III
RESULTS OF THE TICKET CHATBOT USING T5.

model	input	R1	R2	RL	BS
T5 small	question	23.3	7.8	19.9	85.6
	subject: question	27.1	11.3	23.6	86.1
	sub-subject: question	**27.6**	**11.7**	**24.1**	**86.3**
	title: question	27.2	11.4	23.6	86.1
	date interval: question	27.2	11.5	23.7	86.1
T5 large	sub-subject: question	22.1	7.0	18.9	85.3

The chatbot interface and an example of interaction between the user and the FAQ chatbot are shown in Figure 2. As illustrated, the chatbot correctly identifies the user intents and retrieves correct answers. This supports the claim that it is possible to achieve high accuracy using the intent-based approach when user intents are clearly defined and the corresponding answers are structured and self-contained, as is the FAQ scenario.

D. Results on Ticket dataset

Table III presents the results of the Ticket chatbot using the T5 model, comparing different model sizes and input formats. We conducted experiments with multiple input types, providing the model with either the user's question alone or with also additional information or metadata as a prefix. Similar to the results on the FAQ dataset, the additional context provided to the model proves to be beneficial, resulting in improved performance (e.g., 23.3 vs. 27.1 in ROUGE-1 by adding the ticket subject as a prefix). Among the different prefixes tested, including the ticket subject, title or date interval yields similar results, without significant differences among

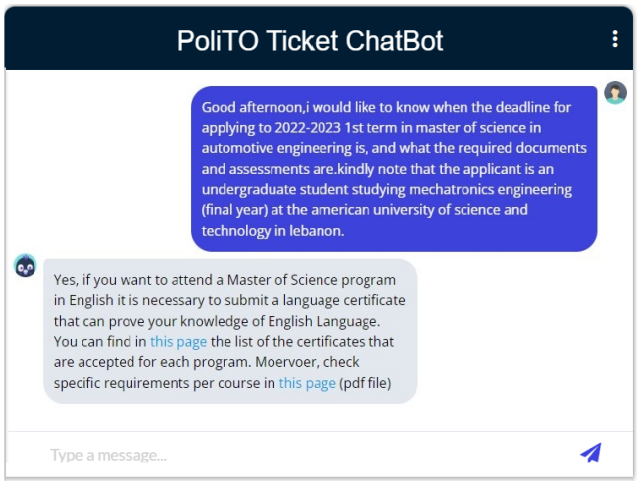

Fig. 3. Chatbot for Tickets – Example of interaction.

the various alternatives. The best performance across all metrics is achieved by using the ticket's sub-subject as a prefix, with an increase of +4.3 and +0.7 points in ROUGE-1 and BERTScore, respectively. We hypothesize that using the sub-subject performs best because it provides more fine-grained information on the ticket's topic than the general subject field. While the title could also be informative, it may introduce noise and not necessarily represent the ticket's content, as it is written by the user. Lastly, the date interval prefix appears less effective, possibly because certain inquiries recur across different or all periods, making it challenging for the model to consistently associate queries with specific time periods.

In summary, incorporating prefix-like context generally enhances model performance, underscoring the benefit of supplying additional information to support the model's understanding. In general, the use of prefix turns out to be beneficial for both FAQ and ticket data. This is likely true especially for more ambiguous questions (e.g., "When are the application deadlines?", without specifying whether the user is referring to Bachelor's or Master's degree), where providing additional context helps in identifying the correct answer more effectively. We also evaluate the best-performing hyperparameter configuration identified with the T5 small model using its large version. Surprisingly, we observe a decrease in the performance, likely due to the larger model's difficulty in generalizing effectively on the ticket dataset, which contains more complex and noisy data compared to the FAQ dataset.

Figure 3 shows an example of interaction between the user and the Ticket chatbot. The output demonstrates that, to a certain extent, the chatbot can understand the user's query and provide a reasonable response. However, through further manual simulations of interactions with the chatbot, we observe that its performance is highly variable, depending on the diversity and specificity of tickets. As a consequence, the model sometimes struggles to provide precise answers, as tickets often include more complex requests containing highly personal and varied content. Additionally, the presence of noise may further hinder the model's understanding, leading to poorer generalization capabilities within the ticket domain. Furthermore, in some cases we also observe common issues associated with text generation, such as the generation of inaccurate content, especially when dealing with highly personalized or uncommon queries that fall outside the model's learned scope. One potential approach to mitigate hallucination issues could involve leveraging more powerful models (e.g., Large Language Models), either in place of the T5 model employed in the current implementation or as validators of the generated outputs, equipped with access to the domain knowledge base. Altogether, this highlights the considerably higher complexity of ticket data compared to FAQs, underscoring the need for more advanced techniques or models to handle them effectively.

VI. Conclusions, Limitations, and Future Work

In this paper, we presented an empirical comparison between different chatbot systems developed at our technical university in Italy. Specifically, we explored the use of FAQs and ticket requests to drive the conversational agents. We also investigated the use of both intent-based and generative approaches. The results show the effectiveness of semantic search for intent detection while coping with questions tailored to either FAQ-related questions and answers or tickets. Conversely, ticket conversations covering a broader or alternative scope are not easily manageable by intent-based approaches, thus requiring the adoption of generative AI.

The current solution has known limitations:

- Due to privacy reasons, the system disregards personal information about prospective students interacting with the system. Thus, conversational agents cannot leverage in-depth user profiling.
- Ticket requests have variable style. The language is often misaligned with the official set of FAQs.
- The implemented chatbot systems currently support only the English and Italian languages.

As future work we aim to:

- Fine-tune language-specific open-source language models (e.g., BART-IT [18]) or adopt lightweight fine-tuning on multilingual Large Language Models (e.g., Mixtral [19]) to enhance or broaden the language coverage of the implemented system;
- Apply ad hoc text paraphrasing techniques (e.g., [20]) to enhance student-generated text quality;
- Summarize verbose ticket conversations [21];
- Generate responses from FAQs to answer unstructured ticket requests.

Acknowledgment

This study was carried out within the MICS (Made in Italy – Circular and Sustainable) Extended Partnership and received funding from Next-GenerationEU (Italian PNRR – M4 C2, Invest 1.3 – D.D. 1551.11-10-2022, PE00000004). This study was also partially carried out within the FAIR (Future Artificial Intelligence Research) and received funding

from Next-GenerationEU (Italian PNRR – M4 C2, Invest 1.3 – D.D. 1555.11-10-2022, PE00000013). The research has also been partly carried out within the SmartData@Polito interdepartmental center at Politecnico di Torino. This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

REFERENCES

[1] V. P. Mahlangu, "Rethinking student admission and access in higher education through the lens of capabilities approach," *International Journal of Educational Management*, vol. 34, no. 2, pp. 450–460, 2020.

[2] I. R. Claire L. Adida, Adeline Lo and S. Williams, "Do equity, diversity and inclusion (edi) requirements change student political attitudes?" *Politics, Groups, and Identities*, vol. 12, no. 4, pp. 958–967, 2024.

[3] C. W. Okonkwo and A. Ade-Ibijola, "Chatbots applications in education: A systematic review," *Computers and Education: Artificial Intelligence*, vol. 2, p. 100033, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666920X21000278

[4] T. T. Nguyen, A. D. Le, H. T. Hoang, and T. Nguyen, "Neu-chatbot: Chatbot for admission of national economics university," *Computers and Education: Artificial Intelligence*, vol. 2, p. 100036, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666920X21000308

[5] A. Jiao, "An intelligent chatbot system based on entity extraction using rasa nlu and neural network," *Journal of Physics: Conference Series*, vol. 1487, no. 1, p. 012014, mar 2020. [Online]. Available: https://dx.doi.org/10.1088/1742-6596/1487/1/012014

[6] B. R. Ranoliya, N. Raghuwanshi, and S. Singh, "Chatbot for university related faqs," in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2017, pp. 1525–1530.

[7] K. Lee, J. Jo, J. Kim, and Y. Kang, "Can chatbots help reduce the workload of administrative officers? - implementing and deploying faq chatbot service in a university," in *HCI International 2019 - Posters*, C. Stephanidis, Ed. Cham: Springer International Publishing, 2019, pp. 348–354.

[8] H. Mangotra, V. Dabas, B. Khetharpal, A. Verma, S. Singhal, and A. K. Mohapatra, "University auto reply faq chatbot using nlp and neural networks," *Artificial Intelligence and Applications*, vol. 2, no. 2, p. 140–148, Jun. 2023. [Online]. Available: https://ojs.bonviewpress.com/index.php/AIA/article/view/631

[9] K. Peyton and S. Unnikrishnan, "A comparison of chatbot platforms with the state-of-the-art sentence bert for answering online student faqs," *Results in Engineering*, vol. 17, p. 100856, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2590123022005266

[10] Microsoft, "Presidio: Data Protection and De-identification SDK," 2024. [Online]. Available: https://microsoft.github.io/presidio/

[11] Rasa Technologies, "Rasa Open Source," 2024. [Online]. Available: https://rasa.com/docs/rasa/

[12] T. Bocklisch, J. Faulkner, N. Pawlowski, and A. Nichol, "Rasa: Open Source Language Understanding and Dialogue Management," 2017. [Online]. Available: https://arxiv.org/abs/1712.05181

[13] T. Bunk, D. Varshneya, V. Vlasov, and A. Nichol, "Diet: Lightweight language understanding for dialogue systems," 2020. [Online]. Available: https://arxiv.org/abs/2004.09936

[14] V. Vlasov, J. E. M. Mosig, and A. Nichol, "Dialogue transformers," 2020. [Online]. Available: https://arxiv.org/abs/1910.00486

[15] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 1, jan 2020.

[16] C.-Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries," in *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 74–81. [Online]. Available: https://aclanthology.org/W04-1013

[17] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=SkeHuCVFDr

[18] M. La Quatra and L. Cagliero, "Bart-it: An efficient sequence-to-sequence model for italian text summarization," *Future Internet*, vol. 15, no. 1, 2023. [Online]. Available: https://www.mdpi.com/1999-5903/15/1/15

[19] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. de las Casas, E. B. Hanna, F. Bressand, G. Lengyel, G. Bour, G. Lample, L. R. Lavaud, L. Saulnier, M.-A. Lachaux, P. Stock, S. Subramanian, S. Yang, S. Antoniak, T. L. Scao, T. Gervet, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed, "Mixtral of experts," 2024. [Online]. Available: https://arxiv.org/abs/2401.04088

[20] M. La Quatra, G. Gallipoli, and L. Cagliero, "Self-supervised text style transfer using cycle-consistent adversarial networks," *ACM Trans. Intell. Syst. Technol.*, vol. 15, no. 5, Nov. 2024. [Online]. Available: https://doi.org/10.1145/3678179

[21] G. Gallipoli, L. Cagliero, and P. Garza, "Extractive conversation summarization driven by textual entailment prediction," in *2023 IEEE 17th International Conference on Application of Information and Communication Technologies (AICT)*, 2023, pp. 1–6.