Prototyping a chatbot for practising conversational English with corrective feedback

Author: Alfiia Gaiazova

Advisors: Nora Aranberri and Arantxa Otegi

hap/lap

Hizkuntzaren Azterketa eta Prozesamendua
Language Analysis and Processing

Final Thesis

June 2022

Departments: Computer Systems and Languages, Computational Architectures and Technologies, Computational Science and Artificial Intelligence, Basque Language and Communication, Communications Engineer.

Abstract

This work focuses on chatbots' application in the educational domain, and precisely, in the area of foreign language learning. The goal of this project is to create a prototype of a chatbot that aims at improving learners' English speaking skills by participating in a conversation on a certain topic. The additional feature of this prototype is the chatbot's ability to generate Corrective Feedback (CF) that allows students to reflect on the mistakes made. The prototype, that was built with the help of the RASA Framework, combines several Natural Language Processing tasks such as automatic Grammar Error Correction and Question Generation. The presented version of the chatbot is capable of identifying errors made by learners and present immediate implicit CF in the form of incorporated interrogative recast.

Contents

List of Figures

List of Tables

1 Introduction

Technology has not always been an essential part of teaching and learning. However, nowadays, amidst the fast development of Artificial Intelligence (AI), it is becoming a frequent aspect within the educational domain. The technology this work is build around is a conversational agent, or a chatbot. The most known and well functioning examples of intelligent assistants are Apple's Siri and Amazon's Alexa. Continuing advances in this field of chatbots' application have created countless opportunities for language learners.

Chatbots' ubiquitous implementation has shifted the traditional perception of how a foreign language is taught and practised. Providing location-independent and 24/7 access, conversational agents made this technology accessible to language learners from all over the globe, who now are able to practice communication skills in a second or foreign language of their choice.

Various applications to practice conversational English or any other foreign language have already been created, yet not all of them can provide a learner with Corrective Feedback (CF), which is crucial to the learning process. CF not only provides students with additional information on whether or not their utterance was correct, but also with explanation in the form of hints or prompts. Therefore, implementing CF should potentially improve the learner's performance and help them see what aspects of the language should be reviewed and practised more.

The goal of this project is to design a prototype of a chatbot that would help language learners whose native language is Spanish practice their English speaking skills by engaging in a conversation with the bot on a particular topic that was previously introduced and learned in a classroom (in this case "Travelling"). The chatbot is going to mimic the teacher's presence by providing students with CF upon conversing to encourage self-reflection.

However, despite the increasing popularity and implementation, creating a successful chatbot capable of sustaining a natural conversation with its user appears to be quite challenging. Both of these aspects, chatbot's advantages and limitations, will be discussed later in this work.

This work is organised as follows. In the beginning (Section 2) a theoretical framework on both technical and educational elements of this work is provided. Next, the objectives are explained (Subsection 3.1), together with the educational setup (Subsection 3.2) for this project and the architecture of the chatbot (Subsection 3.3). This is followed by the description of a dataset that was created for the purposes of this work (Subsection 3.4) and two rounds of experiments that were conducted to select models responsible for the

CF generation (Subsection 3.5). After that, the models will be integrated and the final prototype of a chatbot will be built (Subsection 3.6). And lastly, some conclusions are presented alongside ideas for future work.

2 Theoretical framework

In this section all the theoretical and background information needed to understand the scope of this work is introduced. The section contains 4 subsections that include material on chatbots, their applications, advantages and challenges of this technology, together with a glance into the RASA framework. It is followed by an introduction of CF that is going to be integrated in the chatbot. Lastly, to finish up this theoretical framework section, two of the Natural Language Processing (NLP) tasks that are used to build the chatbot, automatic Grammar Error Correction (GEC) and Question Generation (QG) are presented.

2.1 Chatbots

Chatbot is an AI technology that makes it possible to understand human language through NLP, and to communicate with humans by generating responses to their utterances. The first chatbot was developed back in 1966 and was called ELIZA (Weizenbaum, 1966). It was the first attempt to build a system that would allow users to have conversations with a computer.

In recent years chatbots applications have become ubiquitous and have been integrated by various industries to assist with particular services and provide support to customers. This has been successfully accomplished throughout such sectors as retailing, banking, healthcare, travel, entertainment etc. Five industries that benefit the most from the usage of chatbots, according to the Chatbot Survey (2017)[1], are depicted in Figure 1. Examples of some of the most popular and established conversational agents are Siri, Alexa, IBM Watson, and Dialog Flow. Most of these agents already have integrated cutting-edge AI and NLP technologies that only require some additional fine-tuning in order to customize the agents depending on the project's goals and requirements.

2.1.1 Chatbots and their application in the educational domain

Since for this project the educational domain was chosen as a field of research, a survey on the various chatbot applications for students and teachers have been conducted in order

[1] https://www.ubisend.com/insights/2017-chatbot-report

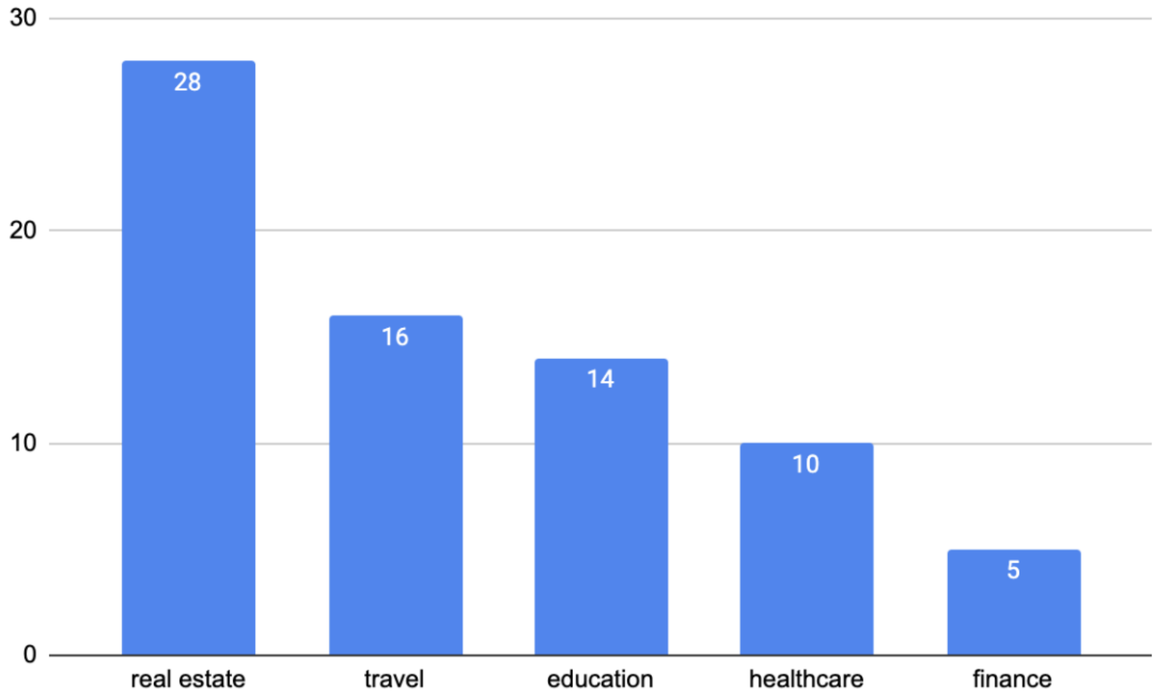


Figure 1: Five industries that benefit the most from chatbots' implementations (in %) according to the Chatbot Survey (2017).

to present the variety of existing approaches and use cases.

This research covers in detail chatbots that are being applied in the educational domain. Recently the pandemic has caused popularisation of e-learning platforms and online education, that has correspondingly resulted in increasing demand to support and automate student learning. It would be interesting to see how these potential language partners represented by chatbots might be exploited in a way that would be beneficial for educational purposes.

The scope of chatbots already incorporated or being researched within the education field is quite vast on its own. They are used as a student support agent, answering frequently asked users' questions about admissions, enrollment, exams, and other organisational processes (Fleming et al., 2018). Also, there are some chatbots that act as teacher's assistants and monitor students' attendance, send them tests or assignments, and conduct assessments (Smutny and Schreiberova, 2020). For example, quite recently a chatbot that has integrated face recognition modules to track students' assistance (Margreat et al., 2021) was introduced. Another example of a chatbot being used in the educational domain is a work of Chen et al. (2016) who have investigated how the usage of a virtual environment supports students' interest and motivation in scientific learning.

Chatbots' ability to communicate in any target language and their 24/7 availability make them also appealing for language teaching researchers. Therefore, there have been numerous studies conducted on integrating chatbots into the language acquisition process to help learners work on their listening, reading, speaking and writing skills. One of them focuses on students being able to answer comprehension questions after reading a story: Xu et al. (2021) have developed an automated conversational agent that narrates stories through a smart speaker and later asks gist or multiple-choice questions and provides feedback. Others have designed a system consisting of various vocabulary tests and quizzes that could be incorporated in Moodle as a part of vocabulary acquisition (Jia et al., 2012). This system also allows teachers to decide on how to assess conducted tests: manually or automatically.

A slightly different approach to chatbot's implementation within foreign language learning domain was made by Lin and Chang (2020). Instead of focusing on reading comprehension, they developed a chatbot that assisted students with academic writing: to come up with their thesis statements, conclusions, essays etc. Apart from the writing skills, there are other researchers who decided to focus on training different students' skill sets such as critical thinking (Goda et al., 2014). They proved that having a conversation with a bot prior to a discussion activity leads to a boost of interactions between the students, and it makes them aware of the critical thinking process.

Some researchers are interested in virtual learning environments and their integration with a chatbot. One of the examples is VILLAGE - a Virtual Immersive Language Learning and Gaming Environment (Wang et al., 2017). It is a 3D virtual simulator with a built-in chatbot that is a part of every role-playing learning scenario (going to a store or a restaurant, at a hotel or an airport etc.) and allows users to have real time conversations according to the situation. The intent behind the VILLAGE is to use it for conversation practice. The biggest online e-learning platforms such as Duolingo[2] and Mondly[3] have already implemented conversational agents in their applications so that their users could develop their conversational skills online based on a topic they have been studying.

A role-playing activity, that has already been mentioned in the previous paragraph, as a means of L2 conversational practice has been a subject of various implementation cases when being integrated in a chatbot. For instance, Ayedoun et al. (2015) designed a chatbot that would simulate a conversation between a user and a waiter that according to the researchers encourages willingness to communicate. Another team of researchers have built a pedagogical agent named Dr. Brown to simulate academic interactions (Yang and

[2]https://www.duolingo.com/
[3]https://app.mondly.com/home

Zapata-Rivera, 2010). This bot is capable of performing certain physical movements such as head and eyes movements, raising a hand, etc.

Considering all the multiple applications of chatbots introduced in this subsection and variations of their execution (robots, simulations, speaking and writing applications), it was decided to build a prototype of a chatbot, that would serve as a practical exercise where students are able to converse on a particular topic akin to the chatbot-simulators presented earlier.

2.1.2 Advantages of using chatbots for language learning

So why has the usage of chatbots in Second Language Acquisition (SLA) attracted the interest of so many researchers? The most obvious reason is their round-the-clock availability. Students can get access to a chatbot at any time of the day and receive an immediate reply, whereas an actual teacher figure who has a limited working schedule is unable to follow that. On top of that, there are also no geographical limitations and attachment to a physical classroom. For instance, in an experiment from Kim (2018), Korean students were using a chatbot as an extracurricular activity to practice English vocabulary outside of a class. All of the above are the examples of conversational agents that are accessible and easy-to-use.

Another advantage of chatbots being used for language learning is their convenience. A lot of chatbots are integrated into ubiquitous messengers or located directly on a web-page, eliminating a need to download extra applications. The proof of that can be found in a research conducted by Smutny and Schreiberova (2020), where they evaluated 47 chatbots within the educational domain that use the Facebook Messenger.

There have been numerous researches in SLA on chatbots being able to engage students in the learning process and lessen their stress and anxiety. According to Fryer and Carpenter (2006) students for the most part enjoyed the process of communicating with a bot, and some of them preferred a chatbot over a teacher. It was explained by decreased levels of shyness that the users demonstrated. Ayedoun et al. (2019) in their research concluded that incorporating chatbots in a foreign language learning process results in students developing more willingness to communicate and having less anxiety. A similar conclusion was reached by Bao (2019), who stated that the use of AI chatbots encourages students while diminishing speech-related anxieties and L2 inhibitions.

An additional feature that characterises some conversational agents is their ability to offer its users an experience of having a conversation with a native-speaker, especially if it is a voiced chatbot. This opportunity, considering that not all the students have access to this kind of teacher, is rather beneficial and alluring to them.

All of the mentioned studies agree in one aspect, activities involving AI-powered conversational agents can improve students' results in SLA. Furthermore, their ability to provide 24/7 service and their accessibility make this technology a valuable instrument for foreign language learners from all over the world who have access to the internet.

2.1.3 Challenges of using chatbots for language learning

Despite having so many benefits to chatbots' application, there are certain challenges and limitations that surround the use of chatbots, their design and implementation. As an example, in their work presented in Fryer et al. (2017) were assessing conversational agents' inability to sustain users' interest in foreign language learning and found out that the levels of engagement in a control group communicating with chatbots decreased compared to a group that was speaking to a human partner. Later, Fryer et al. (2019) mentioned a "novelty effect" to be one of the reasons why the students' interest and the positive effects in performance improvement seem to decrease after a period of time (16 weeks). This effect applies to a technology that is brand-new to the students, making the learners' engagement short-term and causing the effectiveness to disappear after students become more familiar with the technology (Chen et al., 2016).

Despite the expanding implementation of chatbots across the educational domain that was proven in Subsection 2.1.1, not all the existing chatbots are able to improve learners' performance. Various cases of chatbots producing low accuracy utterances have been mentioned in the literature. There are separate reasons for this. The most obvious one is students' misspellings and typos that prevent the chatbot from correctly identifying their intents. As a result the generated outputs might be completely unrelated and incompatible to the input (Haristiani, 2019). According to Yin and Satar (2020) failed communication also occurs when students introduce incomplete sentences as input. Gallacher et al. (2018) argued that the lack of emotions and compassion when communicating with a chatbot reduces the positive effects of language acquisition. Another technical limitation that influences the chatbot's performance is its inability to recognise and process multiple sentences at once (Kim et al., 2019).

On top of that, chatbots' intelligence primarily depends on the dataset chosen to train it since that also influences its ability to decode user messages. For instance, Yang and Zapata-Rivera (2010) have mentioned a problem that chatbots have when learners are changing or introducing a new topic to the conversation, making it difficult for the bot to comprehend their utterances.

It can be concluded that despite the recent advances in AI and chatbot implementations in foreign language learning contexts, there are still challenges to overcome, such as their low intelligence, inability to follow the topic of the conversation, and the necessity of entering short and correctly spelled sentences. All of those raised concerns should be taken into consideration when designing a prototype of a chatbot that would be able to maintain intelligent and natural dialogue with a student.

2.1.4 The RASA framework

There are quite a few development frameworks nowadays that allow developers to build and create a powerful conversational agent, such as Microsoft Bot Framework, RASA, Dialogflow, IBM Watson, etc. The prototype of a chatbot for this project was designed using a tool called RASA[4] since it is a pretty straightforward and an open-source framework to use. Moreover, it has accessible documentation and a possibility of integrating other modules through APIs.

RASA is an open source Machine Learning (ML) framework for building AI assistants and chatbots that was developed so that ML based dialogue management and language understanding could be available to people who are not familiar with software development (Bocklisch et al., 2017). Basically, it consists of two libraries - RASA NLU and RASA Core. The former attempts to interpret a user's message and extract its intent, while the latter decides on which action to take next based on complex algorithms explained below. RASA NLU and RASA Core are independent of each other and can be used separately.

In order to build and train a Natural Language Understanding (NLU) model that is able to identify the user's intents, the training data needs to be defined in *nlu.yml* file which format is quite simple and allows to include as many examples as you want for each intent. The more examples are present, the more accurately the intent will be classified. In fact, it is a good idea to include slang and contracted forms of words that people tend to use while texting. This can make the chatbot better at understanding the way humans interact and express their thoughts.

The second component of this framework, RASA Core, controls the flow of the conversation between a user and the chatbot. To ensure that the dialog flow function performs well, the chatbot is trained using stories that are essentially the samples of a conversation between a user and the bot in a form of various paths the conversation can take. Those stories must be listed in a separate *stories.yml* file. Rules are another type of training data used to train the dialogue model. Rules describe short pieces of conversations that should always follow the same path. They are great at handling small and particular conversation patterns, however, unlike stories, they are not capable of generalising and predicting unseen conversation paths. Therefore, by combining rules and stories when building a chatbot,

[4]https://rasa.com/

its ability to handle real user behaviour is enhanced. After supplying RASA Core with a number of story path examples and rules that a user is expected to follow, a probabilistic model that is used to predict the next action RASA should take is built.

The crucial features of RASA NLU are entities and slots. Entities are keywords that can be extracted from a user's message that are also introduced in the *nlu.yml* file and have specialised syntax. For example, in a possible sentence from a user "*My friends and I went to [Portugal](country) [a few years ago](time)*" the entities are *country*, whose value is *"Portugal"* and *time*, whose value is *"a few years ago"*. Slots represent the bot's memory. They store information that the user has provided (like country "Portugal" from the previous example) or they can also store information extracted via API or database calls. Slots are defined in the slots section of the domain file with the information about their name, type and whether or not they should influence the bot's behaviour.

One of the most important elements of RASA Core are actions. They can be separated into two categories: responses and custom actions. A response is a message that the bot will send back to the user. It is easier to program and therefore, is a more frequent type of action compared to custom actions. Responses are usually executed when the chatbot's job is to send text, images, and buttons back to the user. Responses are introduced under the responses section in the domain file. A custom action is an action that runs a code previously saved in *actions.py* file. There is also the option of creating multiple Python Scripts for RASA custom action or calling an external API using additional Python packages.

The decision on which action to execute next is taken by a policy. There are several training policies already designed by RASA that could be used as a part of the pipeline, such as MemoizationPolicy, MappingPolicy, TEDPolicy, FallbackPolicy etc, that can be combined and applied together. The policies are defined in the *config.yml* file. When several policies are introduced into the pipeline, their execution will be prioritised. The priority is calculated taking into account the confidence score of each policy presented.

In order to store information about the conversation in RASA, an object called Tracker is used. Every conversation session operates with one tracker object that stores the chat's data, such as intents, entities, slots, and a log of all events that took place inside the conversation, as well as a log of all the events that led to that state and have occurred within a conversation. In other words, all of the information that is stored in tracker object is taken into account when policy is being activated.

The process of a chatbot-user interaction is depicted in Figure 2. The first step is executed by Rasa NLU, while the remaining ones are administered by Rasa Core.

We can see that as soon as a message is received at the input it is delivered to an Interpreter (Rasa NLU) to extract information, like entities, from a user's utterance. Next,

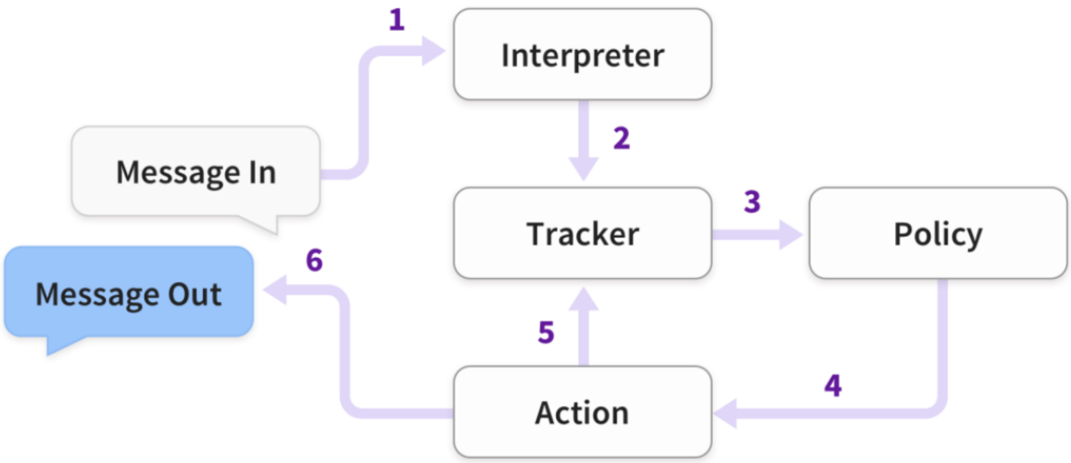


Figure 2: RASA architecture. Source: Bocklisch et al. (2017).

the tracker notes the conversation state and communicates it to the policy. After receiving the updated state of the tracker by the policy it proceeds to select an action to execute. This chosen action is logged by the tracker and delivered to a user in a form of reply. After that, the bot is ready for another round of processing of the user's input and is in "*action_listen*" mode.

Based on the information presented in this section, it can be concluded that the RASA framework possesses all the necessary features, such as ML algorithms, policies, NLU module, custom actions learning, and can successfully serve the purpose of designing a complex and technologically advanced chatbot.

Now that it is clear what conversational agents are, what they are built for, why their usage might be beneficial or challenging, and how they are designed, we can proceed to an educational component of this work, corrective feedback, and how it is used to correct students' errors.

2.2 Corrective Feedback in language learning

Errors embody an important part of the language learning process. According to Ellis and Ellis (1994) errors committed by students help teachers stay informed on their progress and understand what to focus on. Throughout the years the vision of CF in second language teaching has changed from viewing it as redundant and counterproductive to it being a necessary part of SLA process (Arroyo and Yilmaz, 2018). Carroll (1977) has observed that good second language (L2) learners always vigorously ask for information on how correct and appropriate their efforts are. Moreover, according to Han (2002) CF is capable of accelerating the learning process, and, therefore, the absence of it may have a negative impact on the SLA process. Also, the lack of correction can result in confusion among students since they normally require confirmation on whether or not they have been right when they sense that an error has occurred.

2.2.1 Types of Corrective Feedback

In order to explore all the types of CF and identify those that suit the implementation purposes of this project the best, it was decided to use CF classification presented by Lyster and Ranta (1997), as it is being the fullest and most cited up to date. They have distinguished 6 types of CF, that are explained further down the pages together with their examples[5]:

- explicit correction,
- recast,
- clarification request,
- metalinguistic feedback,
- elicitation,
- and repetition.

Explicit correction

When providing students with explicit correction a teacher in a clear manner indicates that their utterance contains a mistake, followed by the correct form and sometimes an explanation or a grammar rule reminder. For example,

– *S: *I am a little scared, there is so many traffic in my city.*
– *T: We don't say "many traffic" in English. "Traffic" is an uncountable noun. It should be "much traffic".*

Where S stands for a student, and T for a teacher.

Recast

There are several definitions of recast that have been formulated throughout the years. For example, Lyster and Ranta (1997, page 10) say it is "the teacher's reformulation of all or part of a student's utterance, minus the error". In other words, a recast happens

[5] All the examples of CF in this subsection are taken from the manually designed dataset described in subsection 2.2.1

when a teacher corrects a student by rephrasing their utterance but does not necessarily say that the utterance was incorrect. Lyster and Ranta (1997) have also distinguished 4 types of recast: isolated declarative, incorporated declarative, isolated interrogative, and incorporated interrogative. Unlike isolated recasts that simply transform a student's utterance without adding supplementary information, incorporated ones complement the original utterances by adding extra words, or in case of an incorporated interrogative recast, a question. The examples are shown in Table 1.

Recast Type	Example
isolated declarative	– T: Is the location important for you? – S: Of course, I like to live more close to the city centre. – T: **Closer.**
incorporated declarative	– T: Is the location important for you? – S: Of course, I like to live more close to the city centre. – T: **I see, you like to live closer to the city centre.**
isolated interrogative	– T: Is the location important for you? – S: Of course, I like to live more close to the city centre. –T: **Why do you prefer living closer to the city centre?**
incorporated interrogative	– T: Is the location important for you? – S: Of course, I like to live more close to the city centre. – T: **I see, you like to live closer to the city centre. And why is it important to you?**

Table 1: Examples of the four types of recast (where recast is marked in bold).

Clarification request

In clarification requests a teacher expresses the misunderstanding caused by a student's utterance by saying special clarifying phrases like: *"Excuse me...?"*, *"What do you mean by. . . ?"* that may be interpreted by a student as a sign of possible presence of an error. For instance,

– *T: What did you like the most about this country?*
– *S: Everything! *I liked the food and the people is very nice.*
– *T: Excuse me?*
– *S: the people are very nice.*

Metalinguistic feedback

Metalinguistic feedback is given when a teacher is not explicitly providing the corrected utterance nor the location of an error, but gives a learner some sort of a comment or a question (or, in other words a metalinguistic clue) regarding their input formulation that makes them aware that their sentence might have been wrongly formed. Without providing the correct form, the teacher asks questions or provides comments or information related to the formation of the student's utterance. For example,

– *S: *My family and I go to Italy last summer for a couple of weeks.*
– *T: It happened in the past, didn't it?*

or

– *T: Do we say "I go" when talking about the past?*
– *S: I went to Italy.*

Elicitation

Openly eliciting the corrected utterances from a student is a type of feedback that is named accordingly - elicitation. The difference between this type of CF and a metalinguistic one lies in the fact that elicitation formulations entail an answer to be a particular form of a corrected word or a phrase and not just a "yes" or "no" response. Lyster and Ranta (1997) mention several approaches that teachers can apply to elicit the corrected reply from a student, such as pausing (for example *"It's a . . . "*), asking questions (for example, *"How do we say X in English?"*), and asking learners to paraphrase their previous utterances:

– *S: *I don't like to go with public transport*
– *T: (Excuse me), you don't like to go __________ ?*
– *S: by public transport?*

Repetition

The last type of CF, repetition, takes place when a teacher repeats the learner's incorrect utterance, bringing their awareness by adjusting the intonation. For instance,

– *S: *I goed to Croatia.*
– *T: You goed? (rising intonation)*

Apart from the above mentioned six types of CF researchers have also been mentioning other approaches when it comes to distinguishing CF depending on the manner in which the feedback is given (implicit and explicit feedback) or the timing of it (delayed and immediate). Both of these approaches complement the already presented classification made by Lyster and Ranta (1997) and can be viewed below.

2.2.2 Implicit vs explicit feedback

CF can be implicit or explicit. Implicit feedback normally suggests that there is no indication made by a teacher that the error has been committed, whereas explicit feedback usually indicates where the error has been made together with occasional detailed explanations (Han, 2002). The most common example of implicit feedback is recast, whereas explicit feedback could be given in a form of explicit correction.

Ellis and Loewen (2007) in their studies have proved that explicit feedback in the form of metalinguistic information is in most cases more effective than implicit feedback that is given in the form of recasts. On the other hand, according to the results obtained by Kim and Mathes (2001) on two different groups of students, there were no significant differences found in the effectiveness of implicit/explicit feedback. In 2006 a research comparing 11 studies dedicated to the influence of implicit and explicit feedback on L2 acquisition was published (Ellis et al., 2006). According to it, it was difficult to reach clear conclusions due to the differences in environments, interactions, tasks provided to the students, measurement techniques and so on. However, it said that explicit feedback had an advantage over implicit one in the studies where the production tests were performed.

Therefore, there is a certain level of uncertainty in claiming that implicit feedback surpasses the explicit one, since there are no standardised evaluation methods that could be used to analyse and draw conclusions from previous research.

2.2.3 Delayed vs immediate feedback

Apart from looking at the extent of the explicitness of CF, researches have also been debating on the feedback timing. Some of them have argued that it should be given immediately (Harmer, 2007), while others stated that CF should be delivered with a delay (Smith and Kimball, 2010). The former position is based on the desire to maintain uninterrupted flow of a performing activity, focusing on fluency. Moreover, there are studies proving that delayed CF is attributed to lower levels of anxiety among students (Rahimi and Dastjerdi, 2012). In 2018 Arroyo and Yilmaz conducted an experiment on three groups of L2 learners that were given different kinds of feedback: immediate, delayed, and no feedback. The results showed that the immediate feedback group was more effective than the other two in terms of oral production. However, there was no significant difference between immediate and delayed feedback groups in terms of students' grammar awareness (that was measured with a Grammaticality Judgement Test). Another interesting discovery was stated by Li (2017), he mentioned that students were the ones who were favouring immediate feedback, unlike teachers, who were reluctant to give it.

Therefore, once again, there is no clear superiority of delayed feedback over the immediate one and vice versa. It seems like it is a matter of a teacher's preference and the class objective, that is whether the main goal of a lesson is fluency or accuracy improvement.

Introducing all the different approaches in CF classification is important since we should take into consideration that some feedbacks could be multidimensional, for instance, multilinguistic feedback is implicit, but it can also be either delayed or immediate. All the types of CF introduced in this subsection are going to be analysed according to their implementation strategy along with discussing the possibility of their integration to the chatbot in question in subsection 3.3.3.

What all six types of CF have in common is the teacher's awareness of mistakes being made. However, a bot designed using RASA framework has no such awareness, moreover, one of chatbot's drawbacks mentioned in subsection 2.1.3 points out how incorrectly entered input negatively impacts a chatbot's ability to recognise the intent behind. To solve the issue of artificially raising chatbot's awareness of an error being made by a student, and potentially empowering it with the ability to correct it, an NLP task of GEC was investigated.

2.3 Automatic Grammar Error Correction in language learning

Since being introduced in 2014 the task of automatic GEC has attracted researchers to investigate new methods and techniques that could be implemented to solve it (Ng et al., 2014). In that study GEC is defined as a task of automatically correcting grammatical errors in written text. In other words, it is a system that analyses the input data in order to detect and subsequently correct identified grammatical errors if they are present in the text. In this project, GEC is an essential part of CF that is implemented in the chatbot that takes as input sentences produced by a student. For the purposes of this project it is important to develop a robust error correction mechanism that would be able to automatically identify the grammar mistakes made by a user and correct them.

Nowadays it is hard to imagine any application involving typing that does not have an incorporated grammar checker. Early grammar checkers were designed manually by creating a set of grammar rules. For instance, Richardson and Braden-Harder (1993) designed a system called Critique that processed text in 6 steps including such processes as determining the sentence/paragraph boundaries, together with lexical, syntactical, and stylistic analysis. Despite being effective for certain kinds of errors, rule-based approaches cannot be used in cases when errors are more complex, as it is infeasible to define rules for every probable grammar error there could be.

In line with the technological advances in NLP, researchers switched over to data-driven methods and started exploiting ML techniques to design classifiers for certain error types. For example, Rozovskaya et al. (2014) developed an integrated model that combines a statistical ML approach with a rule-based one. Although being more effective than the previous approaches, ML classification methods solely focus on distinguishing one error type from another when performing classification, neglecting the cases when L2 learners commit several mistakes within a single sentence. Within the chatbot, implementing a GEC system that is capable of identifying only one type of error per student's utterance and disregarding others would not be sufficient for the purpose of this research, as the chatbot should be ready to deal with the cases when there are several errors of different type present in the input.

To solve the problem mentioned in the previous paragraph, researchers have proposed other grammar correction techniques. Machine translation (MT) algorithms that allow to automatically translate text from a source language into a target language can also be used in the GEC tasks (Rozovskaya and Roth, 2016). Error correction in a way can be interpreted as a translation problem, only in this case the source and the target texts are both written in the same language, the difference lies in the source text being written with mistakes, while the target one does not contain any. Neural MT (NMT) that has developed rapidly in recent years, has proved to be suitable for GEC tasks. For example, Yuan and Briscoe (2016) have developed a NMT model for error correction that has solved the problem of out-of-vocabulary words that often occur in grammatically wrong sentences (for instance, proper nouns and misspeled words). However, this approach also has other limitations. Providing a dataset that represents all the possible errors and their combinations could be problematic and time-consuming. Luckily, this issue was addressed when transformers were designed, since they are able to process larger amounts of data.

The first transformers were introduced by Vaswani et al. (2017) and ever since then NLP has been steadily providing stable performance results. Transformers are neural networks with sequence-to-sequence (Seq2Seq) architecture and the attention mechanism that decides which parts of the input sequence are important. One of the examples was a work of Alikaniotis and Raheja (2019) who used language models based on the transformer architecture and have proved that transformers can outperform n-gram language models when dealing with GEC related tasks. This new technology has also offered a way to deal with errors that are not necessarily present in the dataset, since transformers are capable of generalising the data they have seen. That way, the need to create every possible grammar rule to be able to correct a sentence is no longer relevant.

Overall it is clear that transformers to this day represent the cutting edge solution to solving the GEC problem. However, there are still some complex linguistic phenomena and language use that have not been covered by the existing technology, such as, for instance, ambiguity on lexical, syntactic, and semantic levels.

2.4 Question Generation

In order to generate cogent and cohesive questions that would allow the chatbot to advance with the conversation, it must be able to produce certain questions. To do that another NLP task called QG is used. RASA's output generation framework is rather limited. The response could either be chosen from previously defined responses in the domain.yml file or programmed in custom actions.

In general, QG consists of target answer selection and question construction (Sun et al., 2018). The former allows a model to elicit the part of a passage or a sentence the question should be about. The latter, respectively, is responsible for defining the structure and the form of the question.

The usage of QG is quite vast, whether it is a reading comprehension task, or a conversational system, or a chatbot (Zhang et al., 2021). Moreover, QG is known for its implementation in intelligent tutor systems that are praised for their traits of encouraging self-learning (Nakanishi et al., 2019), which is one of the objectives of this work. Integrating a QG system in the chatbot would strengthen the educational effects of the correct feedback by making students use their analytical skills and personalise the conversation.

The aspiration to teach the machines what questions to ask has existed for over 50 years. Alsubait et al. (2016) believed that the first attempts to implement automatic QG were made in the 1970s. It all started with traditional rule-based algorithms which were transformed into state-of-the-art neural networks. For example, Heilman and Smith (2010) developed manually written rules to execute certain syntactic transformations that changed sentences into questions. The main disadvantages of these conventional methods are their lack of diversity and the fact that it is time-consuming to design the QG models from scratch.

Lately there has been an increase in the interest in Neural Question Generation (NQG) models among researchers. Instead of relying on customised rules, neural models are operating based on large amounts of data-driven approaches and are trained in an end-to-end manner. In other words, those models transform the input at one end to a desired output at another end.

The first neural models used to take a passage and a target answer as an input. Nowadays the presence of the target answer is optional and the input could be embodied in a form of a knowledge base (Elsahar et al., 2018), or deep semantic representations (Rus et al., 2010), or even images (Fan et al., 2018). Compared to the rule-based models NQG has improved the fluency and the flexibility of the generated questions (Yuan et al., 2021).

2.4.1 Types of Neural Question Generation

Considering that the NQG models are the current state-of-the-art it is worth taking a look at the different types of approaches used to build those models in order to decide which ones would be more suitable to implement in the chatbot.

As shown in Figure 3 almost all the NQG architectures could be divided into Pre-trained Seq2Seq, Traditional Seq2Seq, Graph based, and Generative models (Zhang et al., 2021). The first two types of models represent the majority of existing NQG models.

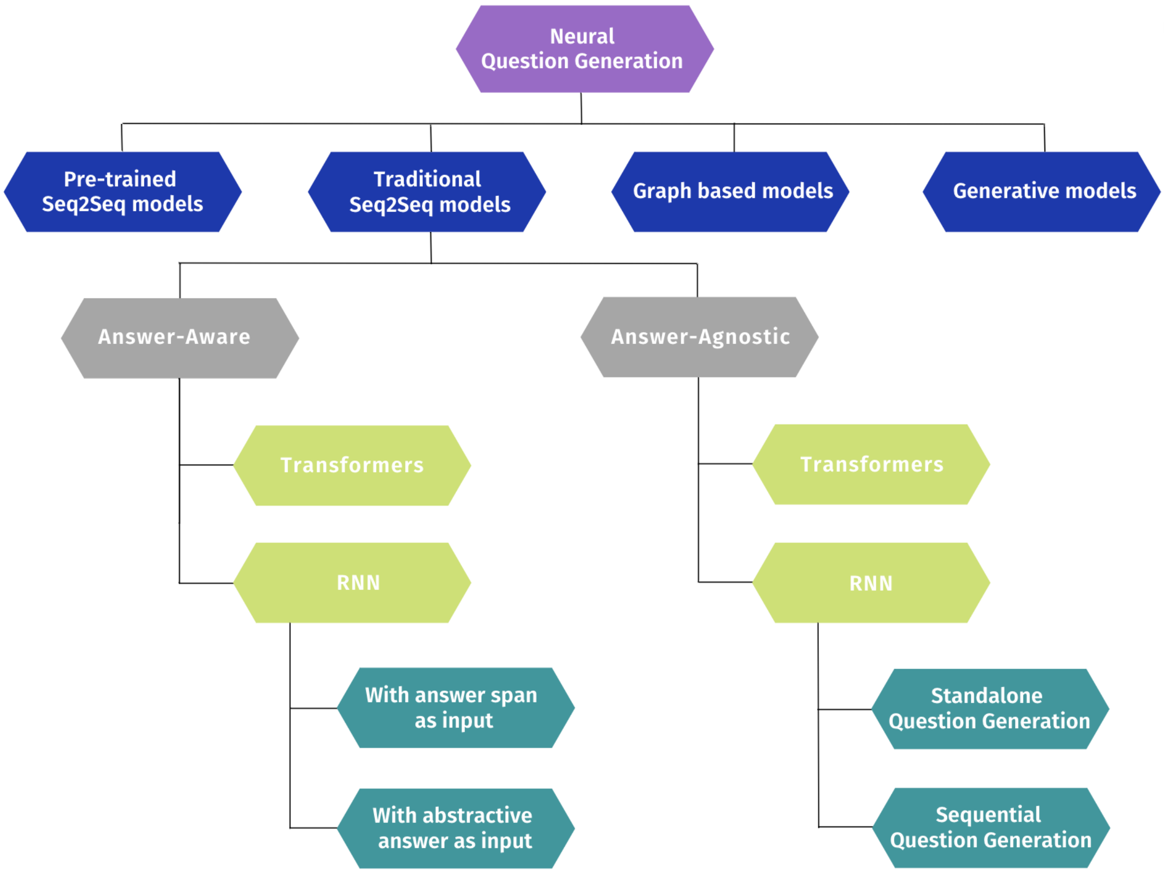


Figure 3: Neural Question Generation Structure. Based on A Review on Question Generation from Natural Language Text by Zhang et al., 2021.

Seq2Seq models are based on encoder-decoder architecture that transforms a set of words at the input to a certain output. The advantage of using a pre-trained Seq2Seq model is that it facilitates the process since it can be fine-tuned to accustom it to any subsequent task.

Traditional Seq2Seq models have different ways to handle the target answer. It could be by using answer span, or abstractive answers (answer-aware models). Alternatively, there could be no answer present at all (answer-agnostic models). Currently, the researchers are more and more interested in designing and developing answer-agnostic models since they allow more freedom when generating the questions (Zhang et al., 2021). However, there are some drawbacks. One of them is the fact that the questions created by those models tend to be unanswerable, meaning that there is no clear answer to be found in the input message.

None of the Seq2Seq models consider the long-distance relationships between sentences. To fix it, graph-based neural models for QG were developed. The graphs are used to design deeply structured objects. Every graph in this kind of model consists of nodes and edges that represent relations between the nodes. This information is loaded into the model in a form of matrix and allows it to find patterns invisible to other models (Yin et al., 2019).

Compared to the previous models, generative ones are able to learn to generate the data similar to the initial distribution. One of the most widespread generative frameworks used in QG is Generative Adversarial Networks (GAN). These models consist of two parts: a generator and a discriminator. The generator's aim is to create new instances, while the discriminator is responsible for determining whether the new generated example belongs to the initial domain distribution or to the model one. The cycle continues by making some adjustments until the model is unable to see the difference between the two (Creswell et al., 2018).

Knowing the existing and state-of-the-art approaches to QG models helps to navigate the research and focus on those models that would suit the task of CF generation better. It would be interesting to compare answer-aware and answer-agnostic models, as well as explore the use of pre-trained Seq2Seq models in QG to determine how their performance differ and which one of them will be more suitable for this project.

3 Setup and integration

This section reveals the motivation behind this project, as well as its objectives and describes the type of learner this chatbot is designed for. On top of that, a detailed look at the chatbot and its structure is given, together with the dataset created, and the experiments conducted in order to determine the models to to be integrated in the prototype. Lastly, the final version of the deployed chatbot is presented, as well as the examples of the conversations between the bot and the user.

3.1 Objectives

The goal of this project is to build and implement a prototype of a chatbot so that Spanish speakers could practise conversational English on a topic of travelling. In particular, this chatbot is required to provide learners with CF to increase the effectiveness and accelerate the learning process.

The motivation behind the idea of creating a chatbot with CF for language learning purposes lies behind years of English teaching experience and realisation that no matter how good the students are at understanding the grammar and learning the required vocabulary, their fluency and accuracy when performing spontaneous conversational practice leaves room for improvement. Unfortunately, not every single student has the opportunity to carry out more individual practice and receive feedback from their teacher. And precisely for this reason, having a chatbot that would be able to perform exactly that task, could contribute to solving this issue.

To achieve the goal a set of specific objectives, each of which are presented in detail in the following subsections, have been established:

1. Build an initial version of a prototype of a chatbot using the RASA framework that would converse with its users on the topic of "Travelling".

2. Analyse six types of CF introduced by Lyster and Ranta (Lyster and Ranta, 1997) in subsection 2.2.1 regarding the possibilities of their implementation in the chatbot.

3. Find or create a dataset of conversations in English relevant to the requirements of the specific educational setup of this project.

4. Conduct experiments to determine the most suitable models of GEC and QG that will be incorporated into the prototype.

5. Integrate the selected models to the already created prototype of a chatbot.

Those objectives will contribute to the purpose of this project, which is exploring the possibilities of creating a tool for those language learners who do not have an opportunity to practise English one-on-one with a teacher. A chatbot will enable them to consolidate the knowledge obtained throughout the classes by practising the material learned.

3.2 Educational setup

In order to build a chatbot that would comply with the goals and objectives established before, it is necessary to determine the type of language learner this chatbot would be beneficial for.

Taking into account multiplying opportunities of chatbots' application in SLA and their variety, teachers should carefully choose which chatbots to introduce in their curriculum. What should be considered beforehand is the students' knowledge of a language and their level. For example, vocabulary training chatbots would better suit beginner groups of learners, while conversational practice ones would be more appropriate for intermediate and advanced students. The reason for that could be found in the study of Fryer and Carpenter (2006), who stated that beginners' ubiquitous mistakes would impede chatbot's ability to analyse the input and, as a result, may generate disappointing responses. Therefore, creating a chatbot for conversational practice would only be suitable for learners of B1 and higher according to Common European Framework of Reference for Languages (CEFR[6]). As stated in a table of CEFR that is called Global Scale, starting from level B1 learners are able to produce simple connected text on topics which they are familiar with, meaning that it would be possible for them to answer the questions or share their experience on a topic previously introduced in a classroom. Starting from level B2 learners, according to the same table, learners are able to express their opinion of a topical issue, together with communicating its advantages and disadvantages. Therefore, it would be appropriate for the purposes of this project to design the prototype of a bot for the learners of levels B1 and B2.

Another important characteristic of learners, that matters when it comes to giving CF, is their native language, since it influences the way learners pronounce words and form the sentences. This phenomenon has been studied by various researchers and is referenced as first language interference (L1 interference). According to Hashim (1999) L1 interference happens when structures of the native language affect learners' performance and development in the target language. This chatbot will be designed for Spanish speakers, therefore the common errors committed by learners that could be explained by L1 interference will be taken into consideration when implementing CF.

It has already been mentioned how students should be able to practice their skills on a certain topic. Thus, the chatbot should also be developed and trained on examples that represent that topic. It was decided to select the topic of "Travelling", since it is introduced at level B1 and could be carried throughout level B2 by adding additional discussions on advantages and disadvantages of a particular type of travel, for instance.

Now that the objectives and educational background for this project have been set up, the next section will describe the initial chatbot's architecture before the CF incorporation followed by the analysis of six types of feedback introduced earlier in order to determine which one of them will be implemented into the final version of the chatbot.

[6]https://www.coe.int/en/web/common-european-framework-reference-languages/level-descriptions

3.3 Chatbot

3.3.1 Architecture of the chatbot

As it was mentioned in section 2.1.4, in order to create a functioning chatbot both components of RASA Framework, RASA NLU and RASA Core, should be trained. Since the theme of "Travelling" has been chosen as context for the English conversational practice, all the intents, stories, and responses were created to satisfy this criteria.

The initial step when creating a new chatbot from scratch is to figure out the possible intent of the user. In other words, what he or she wants to accomplish or to ask. In order to do that an NLU model was built and trained.

Intents created for this task could be divided into two categories:

- common for all of the chatbots that are used in any conversational agents no matter what their functional goals are (for example, *"thank a bot"*, *"affirm"*, *"deny"*, etc),
- specific for the purposes of this chatbot, or, to be precise, intents on various travelling topics (for example, *"places to go"*, *"types of travelling"*, *"accommodation"*, etc.)

The detailed representation of all the 24 intents designed to build the NLU model in terms of this project, together with examples of possible user's utterances, can be viewed in Table 2.

Along with intents, rules, that also take part in RASA NLU training, have been designed. Altogether, there are six rules listed in a *rules.yml* file:

1. Greet the user anytime they say *"hi"*;
2. Say goodbye anytime they say *"goodbye"*;
3. Say *"I am a bot"* anytime the user challenges it;
4. Stop the conversation when the user doesn't want to continue;
5. Say *"You are welcome"* as a response to *"Thank you"*;
6. Ask the user to rephrase whenever they send a message with low NLU confidence (impossible for the chatbot to classify the intent).

In order to train the NLU model the pipeline that is responsible for processing the user's utterance should be defined. It might consist of various elements, each one of which possesses certain functionality, like preprocessing, intent classification, etc. There is are

Type of intent	Name	Examples
common	greet	hey; hello there; good morning
common	goodbye	bye bye, see you later; gotta go
specific	smalltalk_positive	I'm good, thanks; all good in here; I'm fine
specific	smalltalk_negative	really tired; I am a bit sick; Terrible!
common	thanks	alright, thanks; thanks a mil; thx
specific	places_to_go	I want to go to [Japan](country);We went to [Mexico](country) with my friends; We spent time at the [beach](place)
specific	types_of_travelling	I prefer [solo travelling](trave_type); I love going on a [road trip](travel_type) with my friends; lately I only go on [business trips](travel_type)
specific	sightseeing	We went to the [museum](sightseeing_place); We go to the [National Gallery]; I saw many [fountains](sightseeing_place) and [churches](sightseeing_place)
specific	time	I think it was [in May](time); [a few years ago](time); [last summer](time)
specific	good_experience	I really liked the food; I liked everything!;I really enjoyed the weather
specific	reason	because I like; because I think; because in my opinion
specific	reason_covid	I didn't go because of COVID; it's been a while thanks to covid-19; I can't because of coronavirus
specific	transportation	we normally travel by [plane](transp); I prefer [airplanes](transp);we went there by [bus](transp)
specific	accommodation	I usually stay at [hotels](acc); mostly [camping](acc); [hostels](acc) because they are cheaper
specific	advantages	it is cheaper; it's less expensive; it's more convenient
specific	bad_experience	I just don't like it; I hate it; I had the worst trip ever
specific	hope	I hope so; yeah, I hope so too; I hope you are right
common	stop	Can we stop?; Can we finish already?; stop
specific	dont_remember	I have no idea; dont remeber it; it was a long time ago
common	affirm	of course; correct; yes please
common	deny	nope; never; I don't think so
common	bot_challenge	am i speaking to ai?; are you a human or a bot?; are you real?
common	help	help me; how do I use this; I dont understand
specific	help_type	[Accommodation_vocabulary](help_type); [Sightseeing_vocabulary] (help_type); [Types_of_travelling vocabulary](help_type)

Table 2: Examples of the intents and their possible utterances used for the training of an NLU model.

options of selecting elements for your pipeline or creating a custom one. The pipeline defined for this project that consists of predefined elements is depicted in Figure 4.

The WhitespaceTokenizer is used to split the user's utterance into tokens. The RegexFeaturizer creates features for entity extraction and intent classification. During training the RegexFeaturizer generates a list of regular expressions defined in data designed for the NLU training and creates features that are later implemented in entity extraction and intent classification. Similar to that, LexicalSyntacticalFeaturizer also creates features for entity extraction, only in this case they are syntactic and lexical. In order to form bag-of-words vector representations based on the learner's sentences a CountVectorsFeaturiser is used. A bag of words is a text representation that describes the occurrence of words

```
pipeline:
  - name: WhitespaceTokenizer
  - name: RegexFeaturizer
  - name: LexicalSyntacticFeaturizer
  - name: CountVectorsFeaturizer
    analyzer: "char_wb"
    min_ngram: 1
    max_ngram: 4
  - name: DIETClassifier
    epochs: 40
  - name: FallbackClassifier
    threshold: 0.7
  - name: SpacyNLP
    model: "en_core_web_md"
    case_sensitive: false
  - name: "SpacyEntityExtractor"
    dimensions: ["PERSON", "LOC", "ORG", "PRODUCT"]
```

Figure 4: Chatbot's pipeline for training a RASA NLU model.

in a document. A *char_wb* analyzer allows the bot to process them by applying character n-gram models (a way of representing the text in a document as a sequence of characters). DIET stands for Dual Intent Entity Transformer, hence the DIETClassifier is a transformer based algorithm that is able to predict entities and intents. The FallbackClassifier returns a *nlu_fallback* intent when the DIETClassifier could not classify an intent, meaning that the confidence of this intent was lower than the threshold set for the FallbackClassifier. In that case, the chatbot will reply by uttering a default response *"I'm sorry I didn't quite get that. Could you rephrase?"*. The final part of the pipeline is a Spacy's "*en_core_web_md*" language model has also been used to train a chatbot's NLU model. It is a medium-sized English model that was trained on web texts. It is also used to predict the user's entities by using Spacy's entity extraction algorithms within the dimensions set.

The pipeline mentioned above that defines the training process for RASA NLU is a part of the RASA's configuration file. There, alongside the pipeline, the policies for the RASA Core training are also specified. They can be viewed in Figure 5.

As it was previously mentioned, policies establish an action to take at all the steps of a conversation. They could be based either on ML algorithms or on sets of rules. What the MemoizationPolicy does is it remembers the stories set in the stories.yml file. Next, it verifies whether or not the conversation being held matches those stories. If it is a perfect match, then the confidence will be set to 1.0. In the opposite case, no actions are predicted (confidence 0.0). The Transformer Embedding Dialogue (TED) Policy is responsible for

```
policies:
- name: AugmentedMemoizationPolicy
- name: TEDPolicy
  epochs: 50
  constrain_similarities: true
- name: RulePolicy
  core_fallback_threshold: 0.4
  core_fallback_action_name: "action_default_fallback"
  enable_fallback_prediction: True
```

Figure 5: Chatbot's policies for training a RASA Core model.

predicting which action to execute next and recognising entities. As it can be implied from the name, this policy was designed using transformers. In this case, the model was trained for 50 epochs. The RulePolicy is used to control the parts of the conversation that are defined by the rules from the training data.

The last component of a chatbot that will be presented in this subsection is Actions. Altogether there were 33 responses created, some of them are depicted in Figure 6.

```
utter_like_tr:
- text: Do you like traveling?
utter_me_too:
- text: Me too! I love it!
utter_interesting_tell_more:
- text: Oh wow! Tell me more about it
utter_havent_been:
- text: Cool! I've never been there myself.
utter_like_most:
- text: What did you like the most about it?
```

Figure 6: Examples of responses that might be generated by the chatbot.

On top of that there were 5 custom actions designed:

1. action_accom generates a follow-up question using a type of accommodation that was not mentioned by a student. For example, if the accommodation slot is "*hotel*" the chatbot's response would be "*Have you ever tried staying at a hostel?*"

2. action_transp generates a follow-up question using a type of transportation that was not mentioned by a student. For example, if the transportation slot is "*plane*" the chatbot's response would be "*I agree, travelling by plane is the best! And what do you think are perks of travelling by train?*"

3. action_accom_reason_no checks if there is a reason given about the learner's negative answer to the question generated by action_accom. If not, the output would be "*Why not?*", if the reason is given (there is a word "*because*" present in the utterance) the output would be "*I understand*".

4. action_accom_reason_yes checks if there is a reason given about the learner's affirmative answer to the question generated by action_accom. If not, the output would be "*And how was it?*", if the reason is given (there is a word "*because*" present in the utterance) the output would be "*I understand*".

5. action_help provides a student with a review of vocabulary studied in a lesson upon a request. The information on what type of vocabulary to present is taken from the *help_type* slot.

That concludes the chatbot's basic architecture that was developed prior to implementing CF. So far, the chatbot is able to have a conversation about travelling with the learners, but it cannot provide any feedback. The final representation of RASA's architecture for this project, including CF that will be integrated in subsection 3.6 is depicted in Figure 7.

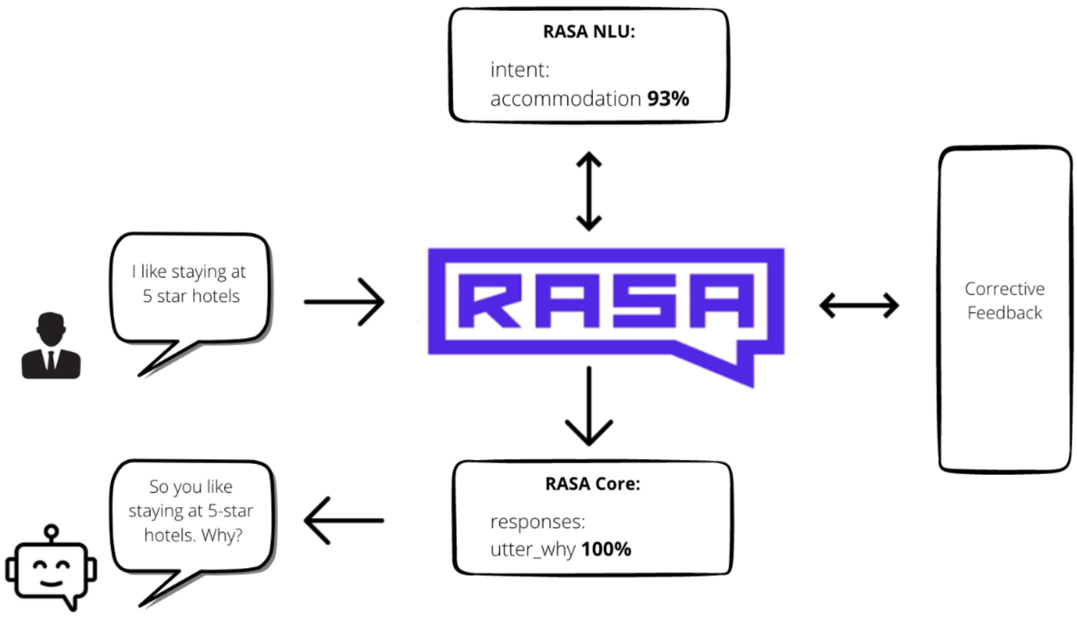
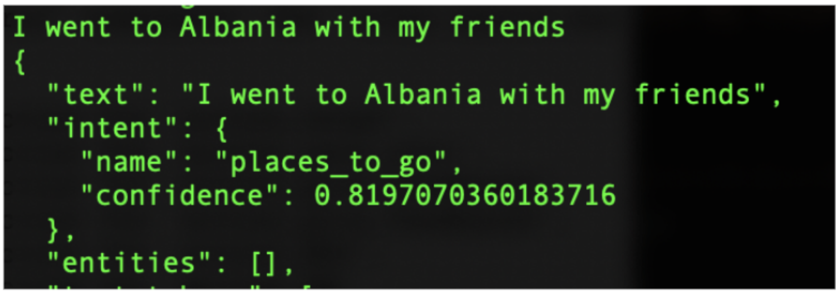


Figure 7: Chatbot's architecture using RASA framework.

3.3.2 NLU model evaluation

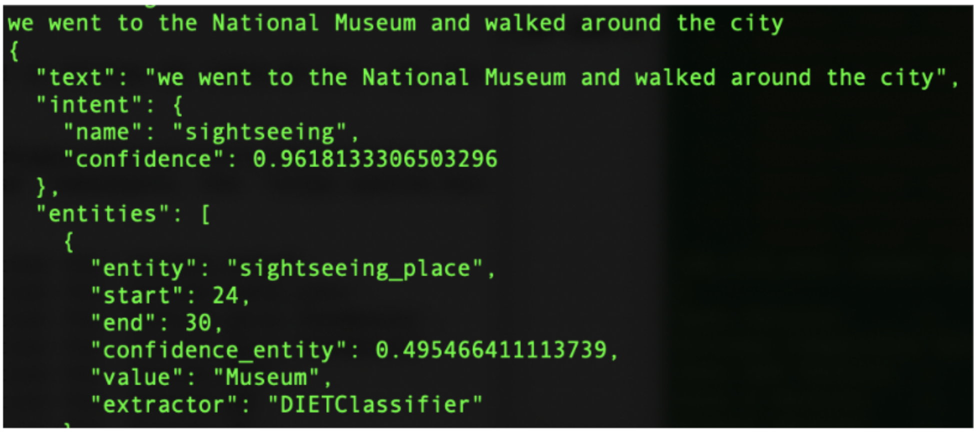
In order to evaluate the built NLU model, RASA has default evaluation metrics. For example, by running a command *"rasa shell nlu"* it is possible to have a conversation with the chatbot and see the confidence that RASA assigns to every intent and entity in that conversation. By feeding the model with utterances that have not been introduced in the training examples, we can see how the model reacts to unseen data. In Figure 8 an example of a successfully identified intent (*places_to_go*) with confidence of 0.82 is presented.



```
I went to Albania with my friends
{
  "text": "I went to Albania with my friends",
  "intent": {
    "name": "places_to_go",
    "confidence": 0.8197070360183716
  },
  "entities": [],
```

Figure 8: Example of intent classification.

Moreover, this evaluation method also provides details about entity classification. In the examples introduced in Figure 8 the entity *"Albania"* that should have been classified as *"country"* was not detected. An example of entities that to a certain extent are present in the training data that have been partially recognised by the NLU model is depicted in Figure 9.



```
we went to the National Museum and walked around the city
{
  "text": "we went to the National Museum and walked around the city",
  "intent": {
    "name": "sightseeing",
    "confidence": 0.9618133306503296
  },
  "entities": [
    {
      "entity": "sightseeing_place",
      "start": 24,
      "end": 30,
      "confidence_entity": 0.495466411113739,
      "value": "Museum",
      "extractor": "DIETClassifier"
```

Figure 9: Example of entity classification.

As you can see, *"Museum"* has been classified as a *"sightseeing place"*, which is not 100% correct, as the full entity value should be *"National Museum"*. There is also information about the type of extraction provided, in this case, the classification was done by the DIETClassifier.

It is inevitable that once the chatbot has been developed and trained, it will be processing messages that have not been seen before when interacting with real users. To see if the model is prepared for it, I have tested the NLU model using an internal RASA's evaluation method that incorporates cross-validation automatically creating multiple train and test splits. Cross-validation is a statistical method to estimate the accuracy of a built model by separating data into several chunks and then testing and training the model on different iterations. In this case, there have been 5 iterations completed. The test results are depicted in Table 3.

	Accuracy	F1-score	Precision
intent	0.702	0.680	0.693
entity	0.937	0.628	0.716

Table 3: Results of NLU model evaluation.

As it can be seen in Table 3, RASA calculates 3 different metrics for both intents and entities: accuracy, F1-score and precision. Based on the obtained results, intent's F1-score (0.680) is slightly higher than that of the entity (0.628). Both models show stable results, however, there is a lot of room for improvement. The most efficient way to do it would be expanding the training data by adding more possible intent examples and paying attention to the examples that could belong to several intents.

3.3.3 Corrective feedback analysis

In this part of the project we consider the analysis of all the different types of feedback, that were previously brought up in subsection 2.2, is conducted. It includes the assessments of the possible implementation of all the types of CF (explicit correction, recast, clarification request, metalinguistic feedback, elicitation, and repetition), along with explanations of the reason why recast has been selected to be a form of feedback that will be integrated in the chatbot.

Explicit correction can be both immediate and delayed. Its implementation in a chatbot could be done by incorporating a rule-based grammar checker with rule description that could be used to generate the definition of an error or a rule explanation. The correction part could be executed by generating either a localised corrected phrase (*"It is more easy"* → '*easier*") or a whole corrected sentence (*"It is more easy"* → *"It is easier"*).

Delayed explicit correction can be revealed to a student at the end of the conversation, by presenting a list of sentences containing errors together with corresponding corrections.

Recasts by definition are implicit and immediate, their implementation depends on their types introduced earlier. For example, the easiest one to implement would be an isolated declarative recast, since the system only needs to know the location of an error and its corrected version. Incorporated declarative one would require additional algorithms, first, for pronoun substitution (*"I goed there with my friends"* → *"you went there with your friends"*) and , secondly, for generation of a string preceding the correction in the output, such as *"So, ..."* or *"I see, ..."*, or *"Oh,..."* (resulting in "*So, you went there with your friends?*"), which again, seems to be quite straightforward. The challenge occurs when investigating both interrogating recasts, since its implementation would require generating of a question that would not only be grammatically correct, but would also not disrupt the natural flow of the conversation and stay on the topic.

As well as recast, **clarification requests** can only reasonably exist in the form of immediate feedback, therefore it would be untroublesome to implement it, since it does not precisely require the generation of a corrected utterance, only its detection.

While being implicit, **metalinguistic feedback** can also be both delayed and immediate. The delayed one could be implemented similarly to the delayed explicit correction way, by providing a student with a list of wrongfully-formed utterances without revealing errors themselves, nor their location. Students might also be asked to attempt correcting them by introducing new sentences. Immediate metalinguistic feedback would also require a use of a rule-based grammar checker in order to generate a cue.

Being implicit and immediate, implementing **elicitation** would be similar to implementing declarative recasts, considering it is also required to get the location of an error (just like isolated declarative recast) and its correction, that after the procedure of pronoun replacement (just like incorporated declarative recast) would substitute the localised error or it's correction for a blank when providing the output.

Implementing **repetition** in a chatbot would only be appropriate if it was a voicebot, which is at the moment out of scope for this project. Of course, it could also be done in a regular chatbot that does not have an integrated voice recognition system. In that case, it would be rather simple and similar to generating an isolated declarative recast.

Undoubtedly, in the real classroom scenarios there are also situations when the teacher operates with combinations of several of the six CF types presented above at once. This should be also taken into consideration when developing a chatbot equipped with a CF layer.

The most suitable types of CF to implement in a chatbot seem to be explicit correction, recast, and metalinguistic feedback. From an NLP perspective, explicit correction and metalinguistic feedback could be implemented using a rule-based grammar checker. However, recast is more challenging since it involves combinations of different NLP techniques. Moreover, among them is Question Generation, which only occurs in interrogative recasts. Therefore, we set to explore the feasibility of this last type in our prototype of a chatbot.

3.4 Dataset

Before trying to automatically implement corrections or generate recasts, a dataset containing examples of dialogues with errors, corrections, and recasts is needed in order to get an idea of the task ahead and structurize the data. To the best of our knowledge, there are no datasets of this kind, and therefore, it was decided to build a new set of instances. Moreover, the dataset should reflect the level of students' English selected before (B1-B2), errors that might be caused by L1 interference, as well as the chosen topic and set required teaching objectives.

The dataset designed for the purposes of this work basically consists of short fictional dialogs (parts of the user-chatbot conversation) based on stories created for RASA, where the users play the role of a Student (S) and the chatbot is their Teacher (T). The mistakes made by the user were artificially created in an attempt to diversify the possible feedback correction methods and identify, if there are, the patterns to generate them. Below an example of a single dataset item is shown.

- T: Did you like the city?

*- S: I don't know. To be honest, I expected more. *The pictures in internet looked so beautiful..but the place was crowded and noisy and a little dirty.*

- T: ***Yeah, the pictures on the internet often mislead us. So you are not recommending this place?***

In this example there was a preposition mistake made (highlighted red), that was later corrected by the bot (highlighted green) using incorporated interrogative recast (in bold). It consists of a correction itself and a follow-up question to guarantee the conversation flow. It was important to create a dataset that illustrates the most common mistakes made by B1-B2 level speakers, so that the results obtained would be robust and representative. For the convenience of future processing every user's utterance contains only one error.

According to Mac Donald (2016), who has investigated the Spanish speaking students learning English, the biggest number of errors were detected within the grammar error category (47% out of all the identified errors). The second largest group consists of the lexical errors (21%). Within the grammar group, the distribution was the following:

1. noun-phrase errors (45% of all the grammar errors), where 27% of them are determiner errors,
2. prepositional-phrase errors (17%),
3. verb-phrase errors (16%).

Taking this distribution into account, and the requirement that the user should be answering the question in accordance with the tense they are being asked ("Required X tense, answered using Y tense" type of error), the dataset was built. In total, the number of its items has reached 100 little dialogues (including at least one student's line and two of the teacher) that were classified according to the mistake type. Each error type and its example are shown in Table 4.

Mistake type	Example from the dataset
Required X tense, answered using Y tense	– T: Where did you go? – S: *I go toBeijing and Shanghai.
Infinitive/Gerund mistakes	– T: Which type of accommodation do you consider to be the most convenient? – S: *I like stay at the 5 star hotels.
Modal verb mistakes	– T: Don't you think solo travelling can be dangerous? – S: *In a way, because you must to pay attention always and take care of yourself.
Other verb mistakes	– T: How would you describe people in Italy? – S: *They really nice and friendly
Preposition mistakes	– T: Did you like the city? – S: I don't know. To be honest, I expected more. *The pictures in internet looked so beautiful..but the place was crowded and noisy and a little dirty.
Determiner mistakes	– T: They are nice, yes. Have you made some friends? – S: *Yes, I have met few locals. We still keep in touch.
Lexical mistakes	– T: Nice! You went to Croatia. How was it? – S: *Amazing! We spent a lot of time on the beach, did surf and swam.
Adjective/adverb mistakes	– T: And can you think of any disadvantages? – S: *Well, they are not fast as planes.
Noun mistakes	– T: Have you seen any sea animals? – S: *Yeah, there were many little crabs on the beach, and the childs tried to catch them.
Spelling mistakes	– T: Yeah, it has a different atmosphere. Have you tried some local food? – S: Yeah, it was delicious. *I tryed a lot of street food. It's not very healthy but I like it.

Table 4: Examples from the dataset according to the error type.

The resulting dataset representation and the various mistakes breakdown is shown in Figure 10. The majority of the mistakes in the dataset are related to the verbs, followed by lexical, article, and preposition ones. The least common ones are Spelling and Adjective/adverb mistakes.

The derived dataset that can be viewed in detail in Appendix A.1 will be used in the following subsection 3.5.

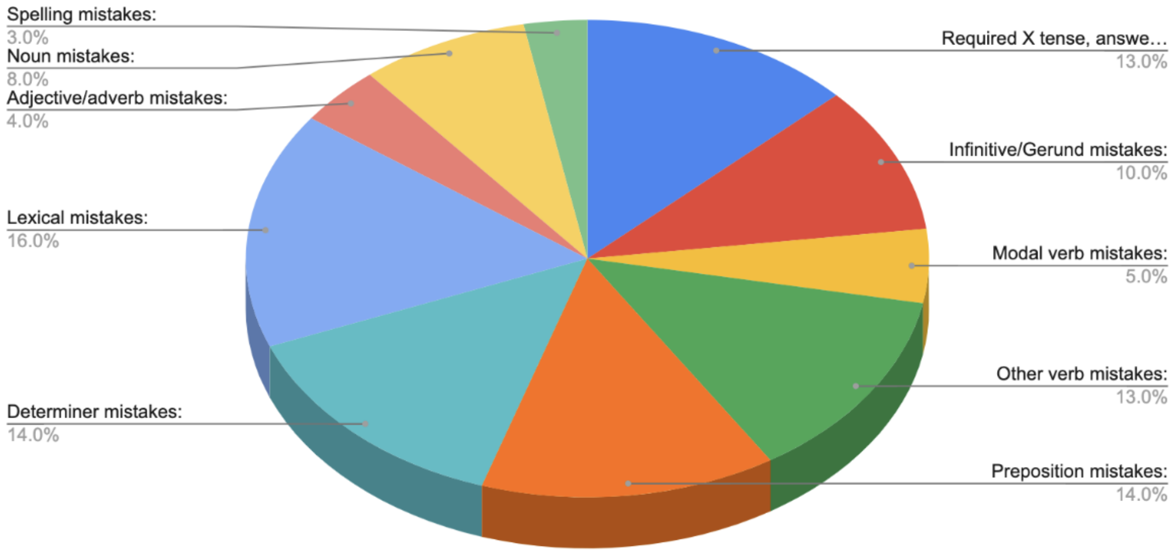


Figure 10: Mistakes distribution in the dataset.

3.5 Experiments

Several experiments for Automated Error Correction and Neural Question Generation were conducted within the scope of this project (Comparison of the error correction models, Comparison of the NQG models, Context altering experiment). Their purpose was to determine which existing models of each NLP task could be implemented in the chatbot in order to generate CF in a form of incorporated interrogative recast.

3.5.1 Automated error correction experiment

The experiment was designed to select an applicable model that would be able to perform automated grammar error correction of user's utterances and it includes investigation of existing and available open-source models and their consecutive evaluation.

Grammar error correction models used

Taking into consideration the purpose of building an error correction system mentioned earlier and the fact that it should be able to correct all the mistakes made within a sentence, even if they are multiple, it was decided to compare the performances of an existing error correction python library that has been created with a state-of-art transformer-based model.

One of the most known python libraries that is able to perform the tasks of error detection and correction is the *language-tool-python*[7]. It is an open-source library that is also used as a spell-checker in OpenOffice. Once installed, the library can be imported and called either locally or through a remote server via API, alongside with a possibility to select a proper language model. So far, there are 43 language models that are supported and available to download, including several variations of English: Australian, Canadian, GB, New Zealand, South African, and US. For this project a "US" model has been selected.

The Language Tool's algorithm of recognising the errors present in the input is following:

1. The introduced text is split into sentences.

2. Each sentence is split into words.

3. Each word is designated with its Part Of Speech (POS) tag (for example, "cities" = plural noun, "visited" = simple past verb)

4. The introduced text is matched against the incorporated Java rules and against the rules stored in the *grammar.xml* file.

Considering that the amount of transformer-based models has considerably increased over the last years, the choice of the second model was made in favour of Gramformer[8], that, as it turned out, had the most number of downloads among the others and can be customly integrated in any messaging platform, including the chatbot that is being developed.

This model was developed by the means of Happy Transformer, which was created on top of Hugging Face's Transformers library. Gramformer's functionality allows it to perform a text-to-text type of task so that the text introduced at the input can generate a new textual example that would be the resulting output. The model was trained and fine-tuned using datasets for GEC like C4[9], PIE[10] and JFLEG[11].

The results of the experiments are presented and analysed in the following section of this work.

[7]https://pypi.org/project/language-tool-python/
[8]https://github.com/PrithivirajDamodaran/Gramformer/
[9]https://github.com/google-research-datasets/C4_200M-synthetic-dataset-for-grammatical-error-correction
[10]https://github.com/awasthiabhijeet/PIE
[11]https://huggingface.co/datasets/jfleg

Comparison of the error correction models

The experiment was conducted using the wrongly-formed students' sentences from the dataset. The Language Tool library allows to parse every instance through its algorithms, detect possible issues and suggest replacements in case the errors were found. If a sentence has multiple mistakes, all of them would be taken into account and corrected. Every detected error has a RuleID, altogether this library includes around 5400 different rules, each of those has a personal identification (for example, *MISSING_TO_BEFORE_A_VERB* which is explained as: "*The verb needs to be in the to-infinitive form*").

The results of the students' sentences being processed with the Language Tool library are depicted in Figure 11.

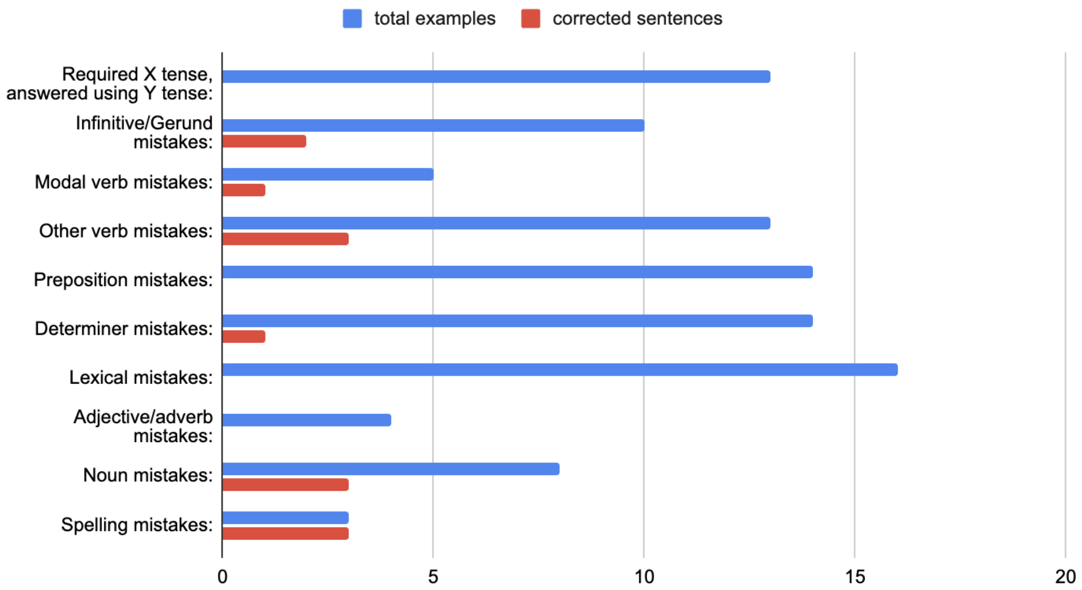


Figure 11: The distribution of the corrected sentences from the dataset after being processed by the Language Tool error correction algorithms.

The types of errors that clearly cannot be handled by this library, since none of the presented examples were corrected, are:

- Required X tense, answered using Y tense,
- Preposition mistakes,
- Lexical mistakes,
- Adjective/adverb mistakes.

While poor performance of the first error type (Required X tense, answered using Y tense) can be logically explained by the lack of the previous context indication of the proper tense use, the performance of the remaining three types is concerning. It would be quite problematic to build a robust error correction layer for a chatbot knowing some of the most representable types of errors (altogether preposition mistakes and lexical mistakes hold 30% of the whole error distribution in the dataset) are not going to be corrected.

The only type of error that has shown a complete 100% ability to be recognised and corrected are the Spelling mistakes. It is followed by Noun mistakes (37.50% of samples were corrected) and three remaining verb-related errors:

- Other verb mistakes (23.08% of samples were corrected),
- Infinitive/Gerund mistakes (20.00% of samples were corrected),
- Modal verb mistakes (20.00% of samples were corrected).

The proportion of the Determiner mistakes that have been successfully processed by the library is only 7.14%. An example of each error category that was corrected by means of the Language tool is shown in Table 5.

Error type	Example containing errors	Corrected example
Infinitive/Gerund mistakes	But we try go on a trip at least once a year .	But we try to go on a trip at least once a year.
Modal verb mistakes	In a way, because you must to pay attention and take care of yourself.	In a way, because you must pay attention and take care of yourself.
Other verb mistakes	It depend. Sometimes I am very hungry and sometimes I am not.	It depends. Sometimes I am very hungry, and sometimes I am not.
Determiner mistakes	I want to go to Maldives, it's a dream of mine.	I want to go to the Maldives, it's a dream of mine.
Noun mistakes	It was my first time travelling abroad, I was excited and trying so many new thing.	It was my first time travelling abroad, I was excited and trying so many new things.
Spelling mistakes	Because I can do whatever I like, meet new interesting people and be independant.	Because I can do whatever I like, meet new interesting people and be independent.

Table 5: Examples of the error correction done by the Language Tool.

Taking into account those ratios and the representativeness of each mistake kind from the dataset, it is clear to conclude that the Language Tool library has proved to be inconsistent and cannot guarantee the robustness of the GEC system for the future implementation of the chatbot.

Another way to correct the students' utterances is to use Gramformer, a Seq2Seq model based on a T5 transformer that was trained to detect, highlight and correct grammar, punctuation, and spelling errors.

Passing the dataset examples through Gramformer has resulted in 52% of them being corrected according to the mistake type, which is substantially higher than the results obtained by the Language Tool library (13%). The total distribution of the corrected and wrong-formed sentences is shown in Figure 12.

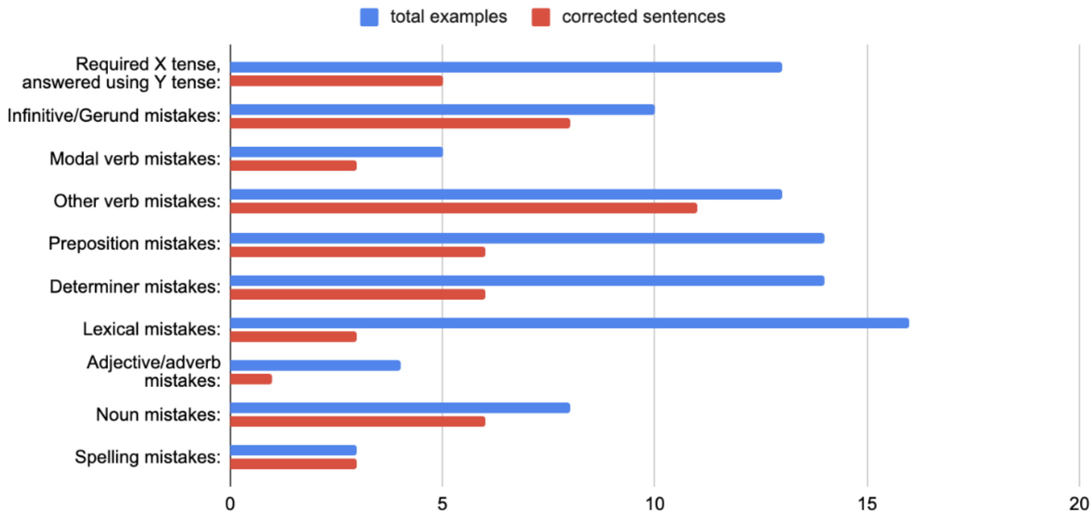


Figure 12: The distribution of the corrected sentences from the dataset after being processed by Gramformer.

An example of each error category that was corrected by means of Gramformer is shown in Table 6.

The Spelling mistakes category has maintained its correctability of 100%, every other category has shown improvements compared to the previous implementation. The pivot Table 7 illustrates the results of both experiments.

Based on those results, it is evident that lexical mistakes are the most troublesome kind of errors to automatically correct. It could be explained by the lack of its representation in the datasets used to train the model. Another apparent issue is the fact that none of the two experimented implementations were able to correct the "Required X tense, answered using Y tense" type of error, that is one of the requirements for the CF layer of the chatbot. Considering the fact that in order for this requirement to be fulfilled, the chatbot's utterance should be taken into account, the possible solution would be to design an algorithm based on Spacy's part-of-speech tags, that is described in details in subsection 3.6 to determine whether or not the user's utterance is in accordance with the question asked by the chatbot.

Error type	Example containing errors	Corrected example
Required X tense, answered using Y tense	I really like Japanese food, I wish I can go to Japan one day.	I really like Japanese food, I wish I could go to Japan one day.
Infinitive/Gerund mistakes	No, but I really want try. .	No, but I really want to try.
Modal verb mistakes	No, I can't to drive. I failed my driving exam 3 times	No, I can't drive. I failed my driving exam 3 times.
Other verb mistakes	They really nice and friendly.	They are really nice and friendly.
Preposition mistakes	When we went on the top of the Tower, the view was breathtaking!	TWhen we went to the top of the Tower, the view was breathtaking!
Determiner mistakes	Not always, but this time we had our vacation at same time so we decided to go somewhere together.	Not always, but this time we had our vacation at the same time, so we decided to go somewhere together.
Lexical mistakes	Well, you can choose where to go, what to do, who to speak.	Well, you can choose where to go, what to do, who to speak to.
Adjective/adverb mistakes	It is funner.	It is funnier.
Noun mistakes	It is hard to choose one, I liked all my trip to Europe.	It is hard to choose one, I liked all my trips to Europe.
Spelling mistakes	I liked the old city neihbourhood.	I liked the old city neighbourhood.

Table 6: Examples of the error correction done by Gramformer.

3.5.2 Neural Question Generation experiment

In this experiment four different NQG models participated to determine which one of them could be implemented in the RASA chatbot to generate an incorporated interrogative recast. Below the description of those models could be found.

Neural Question Generation models used

After studying the existing NQG models' implementations that are freely available for research purposes, it was decided to apply the following ones to examine whether or not it is feasible to adopt them to produce incorporated interrogative recasts.

1. AnwerQuest[12]

This model (Roemmele et al., 2021) has combined Question Answering (QA) and QG models with a purpose to improve reading comprehension type of task. Their QA model is a combination of BERT (a text encoding model) and BiDAF Shared-Norm (a Bi-directional Attention Flow with a shared normalisation approach). For the QG a transformer similar to Scialom et al. is used, that was later augmented. The model was trained using both SQUAD-1.1 and SQUAD-2.0 QA datasets, combined with the dataset made up of CNN news - NewsQA. The automatic and human evaluations conducted prove the advantages of the training data augmentation.

2. docTTTTTquery[13]

[12]https://github.com/roemmele/answerquest
[13]https://github.com/castorini/docTTTTTquery

Error type	Total number of examples	Corrected sentences Language Tool	%	Corrected sentences Gramformer	%
Required X tense, answered using Y tense	13	0	0.00 %	5	38.46%
Infinitive/Gerund mistakes	10	2	20.00 %	8	80.00%
Modal verb mistakes	5	1	20.00%	3	60.00%
Other verb mistakes	13	3	23.08%	11	84.62%
Preposition mistakes	14	0	0.00%	6	42.86%
Determiner mistakes	14	1	7.14%	6	42.86%
Lexical mistakes	16	0	0.00%	3	18.75%
Adjective/adverb mistakes	4	0	0.00%	1	25.00%
Noun mistakes	8	3	37.50%	6	75.00%
Spelling mistakes	3	3	100.00%	3	100.00%

Table 7: Total results of the error correction experiment.

The model was a part of Nogueira et al. (2019) research. It might also be referenced as docT5query or doc2query-T5. This project uses a T5 transformer based QG model that generates the questions by creating queries that are suitable for each document. Next, the generated queries are appended to the original documents. This model was tested on MS MARCO passages and according to the researchers the results have improved compared to other doc2query models.

3. MixQG[14]

Compared to other selected models this model (Murakhovs' ka et al., 2021) was trained using nine different QA datasets to include more all the different types of questions (yes/no, multiple-choice, extractive, and abstractive). The training was done by fine-tuning a pre-trained text2text framework based on T5 (Wolf et al., 2020). The MixQG was able to exceed the results of other equally sized models that were trained directly on the target dataset. On top of that further fine-tuning has also raised the scores.

4. ProphetNet[15]

A large-scale seq2seq pre-training model called ProphetNet (Qi et al., 2020) that was powered up with future n-gram prediction, that improves the prediction of the next tokens. As opposed to previous implementations of two-stream self-attention, this model uses an

[14]https://github.com/salesforce/QGen/tree/main/MixQG
[15]https://github.com/microsoft/ProphetNet

n-stream self-attention mechanism. All of the above allows to prevent the widespread problem of model overfitting. The model was trained with the base scale dataset used in BERT (16GB) and with the large scale one (160GB) similar to BART. ProphetNet has accomplished commendable results on question generation tasks.

Table 8 represents the pivot table with key information about the above models, including their types, datasets used in training, previous works the models are based on, and the implementation method used.

Model	Type	Dataset	Base	Implementation
AnswerQuest	traditional answer-agnostic seq2seq model	SQuAD, NewsQA	BERT + BiDAF	by installing the library from the github repo
docTTTTTquery	traditional answer-agnostic seq2seq model	MS MARCO	T5	through Huggingface library
MixQG	traditional answer-aware seq2seq model	SQuAD, NewsQA, TriviaQA, SearchQA, HotpotQA, NQ, NarQA, MCTest, BoolQ	T5 + BART	through Huggingface library
ProphetNet	pre-trained seq2seq model	BERT + BART ones	Transformer (Vaswani et al., 2017	through Huggingface library

Table 8: NQG models selected for the experiment.

The models presented in this section were compared in order to determine best suitable model. The results of this comparison are presented next.

Comparison of the Neural Question Generation models

Understanding the difficulty behind the idea of automatically created questions, it only made sense to compare the mentioned above models in order to determine their effectiveness and relevance to the goals set by this research. To do so we used the created earlier dataset and passed the user's input containing an error through all four deployed models to generate the questions.

As shown in Figure 13 only one model was fully successful at generating all of the questions (docTTTTTquery), it is followed by ProphetNet model (99%) that has failed to generate a question to a sentence *"They really nice and friendly"* that was given as an input. As a result, in the output the model has provided an empty string.

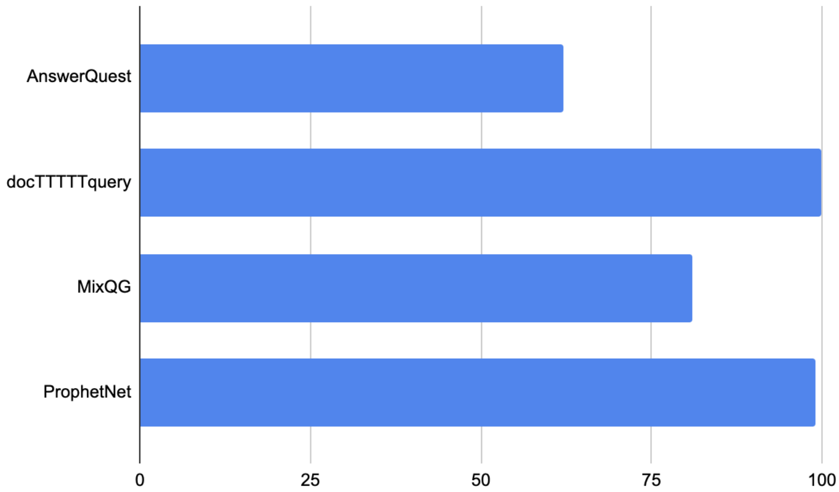


Figure 13: The distribution of the questions, generated by all selected models.

The model that seems to struggle the most is AnswerQuest. Unfortunately, out of 100 sentences given at the input it has only been able to generate 62 questions, meaning that the remaining 38% of them were impossible to generate a question to. The situation with MixQG is the most controversial. The received output covers all 100% of the given sentences, however, only 81% of them are questions. The rest has been split between the exclamatory (ending with a "!") and declarative sentences (ending with a "."), with 12% and 7% respectively. Some of the examples of these sentences are shown in Table 9.

Input	Output
We took the train to Munich. It was my first time travelling in the train and in the city	What a great way to spend a day!
Sometimes, we both like the same activities, we are very same.	Sometimes, we both like the same activities, we are very same.
It was my first try in diving, I loved it so much!	What a great experience it was to dive with the scuba team at the Y
When we went on the top of the Tower, the view was breathtaking!	What a view!

Table 9: Examples of some exclamatory and declarative sentences generated by MixQG model.

Based on the results from Figure 13 it is safe to exclude the AnswerQuest model from the subsequent experiments due to its incapacity of delivering robust output. Another reason why this model was discarded lies in the multiplicity of the output. It would require the implementation of additional algorithms and criteria to choose the final question. There are cases when instead of generating just one question, the model presents a list of questions from 0 to 3. For example:

[“*Where was the first time travelling?*”]
[“*What was beautiful and clean?*”, “*Are there any activities to do?*”]
[“*What do I prefer go on?*”, “*When is everyone free?*”, “*What is it difficult to find?*”]

The challenge of this experiment is to generate a question that is a part of an incorporated interrogative recast. Therefore it is necessary to determine what constitutes a good question in the terms of this work. According to Faruqui and Das (2018), a question is considered to be well-formed if:

- it is grammatical,
- it does not contain spelling errors;
- it is unequivocally a question.

On top of that, created questions should not only be relevant to the topic and allow the flow of the conversation, but also be natural and not repeat the information already communicated.

Taking all of this into consideration and to further explore the effectiveness of the models a set of special criteria was proposed. It can be viewed in Table 10. Three of those criteria based on their importance to the formation of a good question, were marked as “critical” (B, C, D).

Criteria	Full description	Short name
A	Questions that repeat the information already specified in the input	REPEAT
B	Grammatically wrong questions	GRAMMAR
C	Inserting made-up locations/names/objects in the question	MADE-UP
D	Preventing the natural flow of the conversation	NATURAL
E	Drastically changing the context	CONTEXT
F	Not well-formed questions in Total (A+B+C+D+E)	TOTAL
G	Critically wrong questions (B+C+D)	CRITICAL

Table 10: The proposed set of criteria to determine the NQG models’ functionality.

Based on these criteria each of the 3 remaining models have been analysed sentence by sentence. Each sentence could result in having from 0 to 7 of marked criteria. The example is presented in Figure 14. The boxes of TOTAL and CRITICAL were ticked when at least one of the previous criteria has been noted.

Student's sentence	docTTTTTquery	REPEAT	GRAMMAR	MADE-UP	NATURAL	CONTEXT	TOTAL	CRITICAL
Yes, I like solo travel.	do you like solo travel	☑	☐	☐	☐	☐	☑	☐
The best trip I had is the trip to Japan in 2017.	what is the best trip i have ever taken	☑	☐	☐	☐	☐	☑	☐
Yes, they showed me around the city, we have a lot of Mexican food.	what kind of food do mexicans eat	☐	☐	☐	☐	☐	☐	☐
Probably it was the trip I made with my father when I was little.	what was the first thing that triggered my interest in traveling	☐	☐	☐	☐	☐	☐	☐
Everything! I liked the food and the people is very nice. And I saw so many beautiful and interesting places.	what is so nice about korea	☐	☐	☑	☐	☑	☑	☑
I go to Italy last month	where did i go to italy last month	☐	☑	☐	☐	☐	☑	☑
Yes, we had a very good continental breakfast that we can choose in advance.	is breakfast included in the room rate	☐	☐	☐	☐	☐	☐	☐
I like stay at the 5 star hotels.	what hotels are 5 star	☐	☑	☐	☑	☐	☑	☑
Because you have a very comfortable room, don't need to cook a breakfast and clean.	why do you need to cook breakfast	☐	☐	☐	☐	☐	☐	☐
A little, but I like comfort. I think it is worth it and if you can afford	what is the cost of a tv	☐	☐	☐	☐	☑	☑	☐

Figure 14: An example of the selection process for the docTTTTTquery model based on the set of proposed criteria.

Later that information was used to create a pivot table (Figure 15) that assisted in a choice of the model that suits best to the needs of the chatbot.

Criteria	docTTTTTquery	MixQG	ProphetNet
REPEAT	15	20	34
GRAMMAR	12	11	18
MADE-UP	14	5	0
NATURAL	6	12	12
CONTEXT	26	17	15
TOTAL	62	44	60
CRITICAL	27	20	25

Figure 15: The results of the models' analysis based on the set of proposed criteria (where green cell is the best result, yellow is similar results, and red is unsatisfying result.

Numbers represent the amount of sentences that were detected within each of the criteria. For example, ProthetNet model generated 34 questions that repeat the user's unput. According to the table the worst model for the task is ProthetNet due to its high frequency of generation questions that duplicate the input and 18 grammatically wrong outputs. For example,

- Input: "I go to Italy last month" → Output: "the the the the the the the the the the the the the the the the the the the",
- Input: "No, everything was perfect! The best hotel I stayed! → Output: "was the hotel i stayed at the best i stayed at ?".

Nonetheless, it is worth mentioning that it is the only model that has avoided inserting fictional information in the questions. This issue seems to be more acute in the case of the docTTTTTquery model. The level of fabricated locations, that has nothing to do with the conversation, is concerning. For instance,

- Input: "We stayed in the hotel. It was very big and comfortable." → Output: "what is the name of the hotel in san diego",
- Input: "I liked the fact that there are so many different things to do and not leave the hotel." → Output: "what to do in orlando florida".

Another problem with the docTTTTTquery model is its inconsistency with the topic that might confuse a student. For example,

- Input: "I was in so many! Pisa, Venezia, Florencia, Roma." → Output: "where was i filmed in rome",
- Input: "Yes, very often! But I always try to find with the discount." → Output: "do you ever find a discount code on ebay".

Based on Figure 15 it is difficult to instantly choose between docTTTTTquery and MixQG. Both of the models have shown that they have some drawbacks. However, there is a substantial difference between the total number of not well-formed sentences (criteria TOTAL) for the models. Moreover, the amount of critical mistakes for the MixQG is also lower than that of the rest of the models. Hence, the MixQG model was selected as the most suitable one for its future implementation in the chatbot.

There is another curious feature of the MixQG model, that allows us to confidently proceed. And it is the fact that among the generated output there are sentences or questions that already can serve as an incorporated declarative (correcting a student while repeating his utterance) or an incorporated interrogative recast (correcting a student in the form of a question). The examples are presented in Table 11.

Another interesting characteristic to notice is the exclamatory sentences. All 7 of them perfectly fit into the conversation and add a sense of naturalness and this is exactly what differentiates artificial and natural communication. Some of the examples are shown below in Table 12.

Input	Output	Recast type
I like stay at the 5 star hotels.	I like to stay at the 5 star hotels.	incorporated declarative
They really nice and friendly	They are really nice and friendly.	incorporated declarative
No, we stayed there one week	We stayed in the same town for only one week.	incorporated declarative
The best trip I had is the trip to Japan in 2017.	What was the best trip I took in 2017?	incorporated interrogative
I liked the old city neihbourhood.	What was the name of the neighborhood?	incorporated interrogative
Well, they are not fast as planes.	Are they as fast as planes?	incorporated interrogative

Table 11: Examples of the MixQG model's output suitable for incorporated declarative and incorporated interrogative recasts.

Input	Output
We stayed in the hotel. It was very big and comfortable.	What a great place to stay in the mountains!
We took the train to Munich. It was my first time travelling in the train and in the city	What a great way to spend a day!
When we went on the top of the Tower, the view was breathtaking!	What a view!
It was wonderful! The best trip I have!	What a wonderful trip!

Table 12: Examples of exclamatory sentences generated by the MixQG model that fit the natural flow of the conversation.

Context altering experiment

After determining that the MixQG model suits the task better, another experiment to see whether adding more pre-context would somehow improve the model output was conducted. The hypothesis was that by expanding the model's input, it will generate more natural and context appropriate questions. It might also decrease the amount of critical and total errors.

The procedure of the input enhancement was done by expanding the dataset. To do that the sequencing of the examples was determined, so that each instance of dialog consisted of more than just a single student utterance and up to three when applicable. In order to detect the best enhancement tactics it was decided on two different approaches:

1. including both questions from the chatbot and the user's answers (Dataset+),
2. including only a set of the user's answers (Dataset_Answers).

Some examples of the enhanced datasets are shown in Table 13.

Initial Dataset	Dataset+	Dataset_Answers
- T: Yeah, I prefer going with my friends too. Where did you stay? - S: *We stayed in the hotel. It was very big and comfortable. - T: **Nice! You stayed at the hotel.** And how was the service there?	- T: When was the last time you went travelling? - S: *My family and I go to Italy last summer for a couple of weeks. - T: **Oh, you went there with your family.Do you often travel with them?** - S: *Sometimes, to be honest I prefer go on trips with my friends but it is very difficult to find the time when everyone is free or on holidays. - T: **Yeah, I prefer going with my friends too. Where did you stay?** - S: *We stayed in the hotel. It was very big and comfortable. - T: **Nice! You stayed at the hotel. And how was the service there?**	- S: *My family and I go to Italy last summer for a couple of weeks. - S: *Sometimes, to be honest I prefer go on trips with my friends but it is very difficult to find the time when everyone is free or on holidays. - S: *We stayed in the hotel. It was very big and comfortable.
- T: What do you think are the advantages of a ferry ride? - S: *I don't know, I guess you can take your car with you, or other heavy thing. - T: **Other heavy things? Like what?**	T: What do you think are perks of travelling by plane? - S: *Well, it fast and it can take you anywhere you want. - T: **Yes, it's fast. And have you ever taken a ferry?** - S: *No, but I really want try. - T: **I want to try it too. What do think are the advantages of a ferry ride?** - S: *I don't know, I guess you can take your car with you, or other heavy thing. - T: **Other heavy things? Like what?**	- S: *Well, it fast and it can take you anywhere you want. - S: *No, but I really want try. - S: *I don't know, I guess you can take your car with you, or other heavy thing.

Table 13: Examples from enhanced Dataset+ and Dataset_Answers.

These two new datasets were used to run the MixQG model using different input data. For the purposes of distinguishing the results the models have been called MixQG+ and MixQG_Answers respectively.

Some examples of enhanced input and the resulting new questions for MixQG+ are shown in Table 14 together with the MixQG results. After performing the step, the striking problem has appeared. As it turned out, adding the previous chatbot's outputs has resulted in the model generating the question that has already been asked by the bot. It is a setback that disrupts the natural flow of the conversation. Out of the 100 examples of the dataset, 69 questions have been mistakenly generated this way. Therefore, it is clear that the initial hypothesis in this case was wrong and the previous MixQG results have not been improved.

MixQG input	MixQG+ input	MixQG output	MixQG+ output
Everything! I liked the food and the people is very nice. And I saw so many beautiful and interesting places.	When was the last time you went travelling? My family and I go to Italy last summer for a couple of weeks. So you went to Italy. What did you like the most about this country? Everything! I liked the food and the people is very nice. And I saw so many beautiful and interesting places.	What did I like about my trip to the Balkans?	What did you like the most about Italy?
Yes, they showed me around the city, we have a lot of Mexican food.	When was the last time you went travelling?I just returned from Mexico couple of days ago. A couple of days ago? Wow! Was it a work trip? No, I was visiting my friends in Guadalajara. Did you have fun? Yes, they showed me around the city, we have a lot of Mexican food.	Did they show me around the city?	When was the last time you went travelling?
No, I can't to drive. I failed my driving exam 3 times	What do you think are perks of travelling by car? I think it's really nice if you like drive. And do you like driving? No, I can't to drive. I failed my driving exam 3 times.	Can I drive if I have a criminal record?	What perks do you have when travelling by car?

Table 14: Examples of enhanced MixQG+ input (consisting of both the user's and the bot's previous utterances) with the resulting output in comparison with the MixQG results.

To continue with the experiment, the extended input has been limited only to the user's utterances. Some examples of enhanced MixQG_Answers input and the resulting new questions are shown in Table 15 together with the MixQG results. As a result, the problem about duplicating the questions from the input that has been encountered with the Dataset+ input has disappeared and the majority of the generated output has changed.

MixQG input	MixQG+ input	MixQG output	MixQG+ output
Sometimes, we both like the same activities, we are very same.	No, I went with my young sister. Sometimes, we both like the same activities, we are very same.	Sometimes, we both like the same activities, we are very same.	Did I go to the same place with my sister?
Because you have a very comfortable room, don't need to cook a breakfast and clean.	I like stay at the 5 star hotels. Because you have a very comfortable room, don't need to cook a breakfast and clean.	Why do you need to cook breakfast and clean?	What is the best hotel for a 5 star hotel?
Everything! I liked the food and the people is very nice. And I saw so many beautiful and interesting places.	My family and I go to Italy last summer for a couple of weeks. Everything! I liked the food and the people is very nice. And I saw so many beautiful and interesting places.	What did I like about my trip to the Balkans?	What did I like about Italy?

Table 15: Examples of the enhanced MixQG_Answers input (consisting of only user's previous utterances) with the resulting output in comparison with the MixQG results.

The Figure 16 shows the summary table for all three variations of inputs for the MixQG model that was designed using the same set of criteria (Table 10).

Criteria	MixQG	MixQG+	MixQG_Answers
REPEAT	21	81	37
GRAMMAR	11	1	5
MADE-UP	5	5	5
NATURAL	11	0	1
CONTEXT	16	6	6
TOTAL	45	89	45
CRITICAL	20	3	10

Figure 16: The results of the models' inputs based on the set of proposed criteria.

While the aggravation of the MixQG+ model is quite noticeable, the obvious improvement of the MixQG_Answers model is debatable. Despite the reduced number of critical errors, the number of questions that duplicate the input has increased. When closely analysing the possible improvement of MixQG+ and MixQG_Answers model to the original MixQG (Table 16) by comparing each generated question individually, the distribution of improved and worsened questions is 21 to 25, which again, makes it hard to take an explicit decision about whether or not the enhancement of the pre-context plays a positive part in chatbot question generation. Therefore, it was decided to stick to the original way of presenting the context for the MixQG model without any enhencement techniques.

Type of change	MixQG+	MixQG_Answers
Better	12	21
Betterish (critical error replaced with a repeated question)	13	11
Same (relatively)	21	21
No change	3	22
Worse	51	25

Table 16: Results of analysing individual questions generated by enhanced models (MixQG+ and MixQG_Answers) based on the improvement criteria.

Inintial pronoun preservation

Another issue that is present throughout all of the models is the preservation of the initial pronouns in the output. Making sure that the pronoun use is properly aligned with the comunicative intention contributes to natural flow of the conversation, otherwise instead of the chatbot asking questions to its users in the second person, it would be asking itself questions in the first person. Some of the incorrect examples and proposed corrections are shown in Table 17. This problem would require the post-processing to be performed upon the QG model's output, before it could be used as a chatbot's utterance. This will be implemented in subsection 3.6.

Input	Output	Desirable output
I would like to go to Thailand, my friends say to me it is a beautiful country.	What country do I want to go to?	What country do you want to go to?
It is hard to choose one, I liked all my trip to Europe.	What is the best part of my trip to Europe?	What is the best part of your trip to Europe?
I try, because travel with my best friend is the best	How do I travel with my best friend?	How do you travel with your best friend?
Yes, because it was so good! And I am little lazy to go to the cafe in the morning.	why do i eat breakfast at cafe	why do you eat breakfast at cafe

Table 17: Examples of incorrect pronoun persons generated by NQG models (column "Output") and the proposed corrections (column "Desirable output").

The results of the conducted experiments show that existing NQG models could be used as means of generating not only incorporated interrogative recast, but also incorporated declarative and incorporated interrogative ones. However the models used in this project still lack fluency and tend to replicate information already mentioned by users when generating questions. It was also determined that in order to proceed with the obtained output, it is necessary to add post-processing. This supplementary step would fix the pronouns in the chatbot's utterances.

3.6 Corrective feedback integration

The final stage of this project is to incorporate an interrogative recast to the chatbot, introduced in subsection 3.3.1 of this work, that would be given to a student in case he commits an error. However, the chatbot's architecture presented in subsection 3.3.1 does not cover the feedback implementation of CF in the form of recast. Since recast is immediate, it should be generated after each student's input. The schematic representation of how the CF will be incorporated for each round of conversation is depicted in Figure 17.

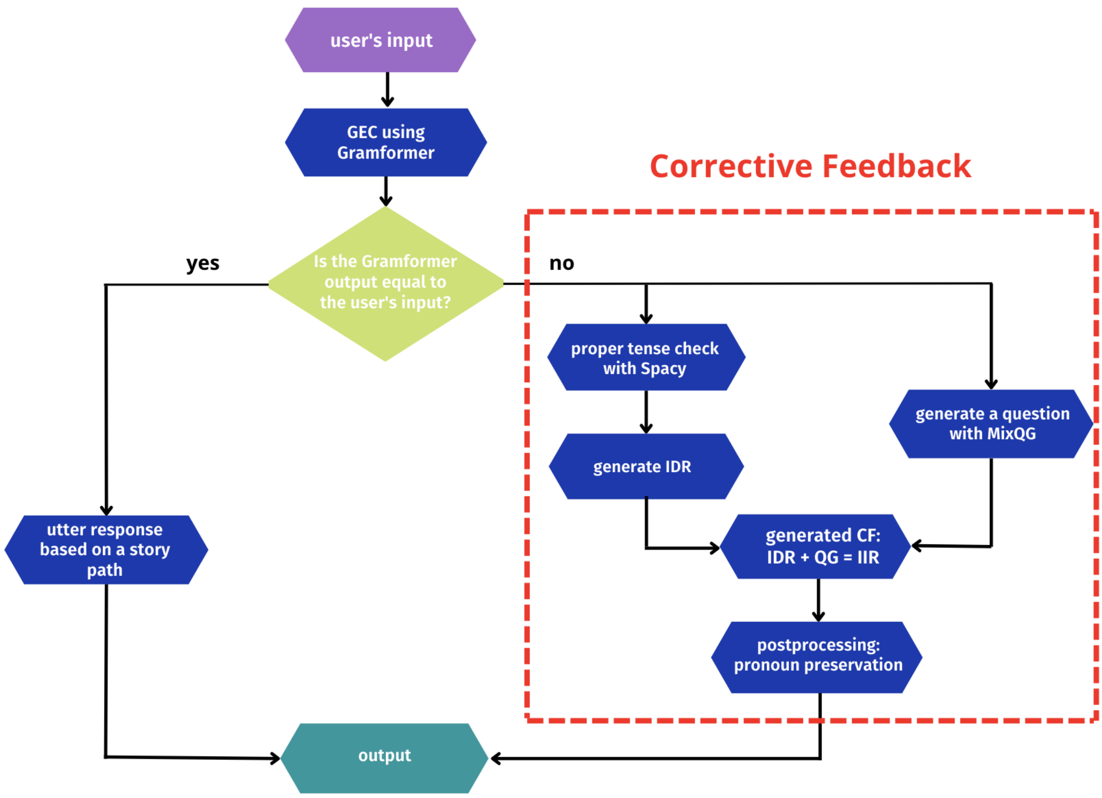


Figure 17: The algorithm of CF implementation.

Depending on the input's correctness the chatbot will either generate a response based on the classified intent and a story path or generate CF, that will be the final output given to the user.

CF in a form of incorporated interrogative request (IIR) is generated out of two components:

- incorporated declarative recast (IDR),
- a question, generated by the MixQG model.

Before generating an IDR, the requirement of the compliance of the verb tenses being used in a question asked by the chatbot and the answer given by a student, should be fulfilled (proper tense check with Spacy). It was accomplished by incorporating Spacy, a python library that supports part of speech tagging. In this case, the tags that are relevant to the project are *token.pos_ == 'VERB'* and *token.tag_ == 'VBD'* (where VBD being a verb in past tense). The algorithm identifies whether or not the question requires the corresponding answer to be in the past tense based on the manually introduced list of responses that do, such as *utter_when*, *utter_interesting_tell_more*, *utter_like_most*, etc. If the user's utterance, parsed with Spacy, is in the past tense, the corrected sentence is replaced (based on data of type dictionary that consists of present and past forms of irregular English verbs) by a new one, that is indeed in the past. Otherwise, if the time reference is correct, this Spacy step is skipped.

Next, by adding *"So, "* the corrected sentence becomes an IDR, that, after being combined with a question generated by the MixQG model, becomes an IIR and is almost ready to be presented to the user. Finally, the sentence should also undergo a transformation in a form of post-processing in order to adjust the characteristics of personal pronouns as it was mentioned in the subsection 3.5.2 under **Initial pronoun preservation** title:

Generated CF: "So, I like to travel by plane. What country do i want to go?" → Desirable output: "So, you like to travel by plane. What country do you want to go".

After completing the prototype's architecture and implementing the new structural changes to the previous chatbot's version, we can finally test the bot by taking on a role of the student. Some examples of generated CF that come as a result of trial conversations between a student (S) and a bot (B) can be seen below:

- S: I also liked the different museums and walking on the city
-B: So, you also liked the different museums and walking around the city. What did you like about the city?

or

- S: I stayed on a small hotel near the city center
- B: So, you stayed at a small hotel near the city center. The hotel was small and the staff was friendly.

In both examples the first part of IIR (IDR) has been generated correctly: the mistake has been corrected, the pronoun is replaced. The second part that was generated by the MixQG model can be characterised as a well-formed question in the first example. In the second one, the generated output is not a question, however it is still understandable and doesn't interrupt the flow of the conversation much.

While the bot shows great potential for effective IIR, there are still issues related to the human-bot interaction that remain to be addressed. For example:

- S: Hi

- B: What's up! How are you in there?

- S: good

- B: Sounds great! Are you ready to talk about travelling?

- S: yes

- B: Glad to hear that! Let's go! Do you like travelling?

- S: yes

- B: Me too! I love it! When was the last time you went travelling?

- S: last month

- B: So, last month, last month, the month was a busy month. In the last month, the month of the same month, the month of the same month the

As you can see, the beginning of the conversation in this example is quite natural until the user's utterance *"last month"* is processed. The bot's last response is quite incomprehensible, both IDR and a QG output. It can be assumed that the reason lies in the Gramformer's inability to process incomplete sentences, since after close examination it was discovered that its output after processing the phrase "last month" turned out to be "last month, last month, the month was a busy month." This hypothesis was proven after entering similar phrases and monitoring the prototype's behaviour. Another example is could be seen as following:

- B: Me too! I love it! When was the last time you went travelling?

- S: a few months ago

- B: So, a few months ago, a few months ago, a few months ago, a few months ago, a few months ago, a few months ago, a few months ago, a few months ago, a. When did the last time a new post was written about a new topic?

This time, the part generated by the MixQG model is closer to desirable, however the output of Gramformer model is still unacceptable ("a few months ago, a few months ago, a few months ago, a few months ago, a few months ago, a few months ago, a few months ago, a few months ago,"). This is an issue that has not arisen during the previous experiments, since all the examples from the dataset are well-formed and complete sentences. On the contrary, when providing a full-sentenced answer to the bot's question "When was the last time you went travelling?" the issue disappears and the conversation can be continued:

- S: hi
- B: Hello there! How's it going?
- S: all good
- B: Awesome! Are you ready to talk about travelling?
- S: yeah
- B: Cool! Let's go! Do you like travelling?
- S: yep
- B: Me too! I love it! When was the last time you went travelling?
- S: I went travelling a few months ago
- B: Oh wow! Tell me more about it
- S: I goed to Greece with my friends
- B: So, you went to greece with your friends. Did you go to Greece with friends?
- S: Yes, I went to Greece with my friends
- B: Cool! I've never been there myself. What did you like the most about it?
- S: I liked the arquitecture and the food
- B: So, you liked the architecture and the food. The food was good, the architecture was nice, the architecture was nice, the
- S: I don't understand
- B: I can help you by reminding you of some vocabulary and grammar. You can ask me things like:

•Accommodation vocabulary
•Sightseeing vocabulary
•Types of travelling vocabulary
•Transportation vocabulary
•Grammar

In this dialogue the integrated Gramformer's model was able to successfully correct both committed errors, however, the QG element of the CF has failed in the second IIR, since it interrupted the flow of the conversation, but, the bot was able to recover.

Another curious output was discovered while conversing the prototype:

- B: Cool! I've never been there myself. What did you like the most about it?
- S: I the most liked the central park
- B: So, you liked the most the central park. I liked the central park the most.

What happened here, is that even though the GEC module made an attempt at correcting the error, it is still syntactically incorrect. However, the sentence generated by the

MixGQ model is correct. That could be explained by the difference in the datasets used to train both models.

Since a chatbot built using the RASA tool can also be connected to other services like common messaging and voice channels (Facebook messenger, Telegram, Slack and so on) the final version of the prototype was integrated to the Telegram application by hosting it over https domain[16].

The designed prototype showed that it is capable of identifying the errors and generating CF in the form of IIR and maintain a conversation on the topic of Travelling. However, the current implementation is not robust enough to handle real user behaviour yet since learners rarely answer in a proper sentence manner.

4 Conclusions and future work

The goal of this project to design and implement a prototype of a chatbot to practice conversational English equipped with CF for Spanish speakers was achieved.

The created prototype is able to converse on the topic of Travelling and provide CF whenever the user makes a mistake. The objectives set in the subsection 3.1 of this work were reached. To do so, all types of CF were analysed across three different approaches (Lyster ans Ranta's classification, explicit vs explicit, delayed vs immediate), and, as a result, it was decided to choose IIR to be integrated in the prototype. A completely new dataset of student-teacher interactions was created to help conduct experiments that resulted in Gramformer being selected to perform automatic GEC and the MixQG model to generate a question representing a part of IIR. And finally, the ultimate version of the prototype with integrated CF was implemented and added to Telegram.

The prototype has demonstrated stable dialogues with errors being recognised throughout the conversation making it possible to assume that designing a chatbot capable of providing immedeate CF is feasible. Yet, there is ample room to improve the models. The NLU model of the chatbot has proven to be stable and able to recognise the majority of the intents (0.702 of accuracy) and entities (0.937 of accuracy).

However, some issues were also encountered along the way:

- When being used through Telegram, the first time the chatbot engages outside models, such as Gramformer and MixQG, the time of processing the user's input increases up to 31-33 seconds, and then decreases to approximately 22-24 seconds.

[16] https://t.me/Alfia_TFM_bot

- Short incomplete user's replies, for example *"last summer"*, are not processed correctly by Gramformer creating a very confusing for a learner output.
- Quite often the generated questions are not precisely questions, but affirmative sentences. There are times when those sentences completely repeat the user's message.
- The remaining issues that are peculiar to all the existing chatbots mentioned in the subsection 2.1.3: the novelty effect, typos and mispellings that lower the NLU model's accuracy, multiple sentences at once, etc.

Those issues shape an approach in future work dedicated to solving the problems that have been mentioned. The first issue could be solved by hosting the chatbot over more powerful machine instead of using a personal laptop. Finding alternative models of GEC and QG with better performance results or obtaining a dataset of real student-teacher interactions full off errors and their corrections could help solve the second issue. To improve the NLU model we could increase the training data buy adding more possible intent examples. Also, it would be appealing to focus on the chatbot's functionality and it's expansion, for instance, implementing other types of CF and develop the conversation around other topics like "Health", "Hobbies", ect.

References

Dimitrios Alikaniotis and Vipul Raheja. The unreasonable effectiveness of transformer language models in grammatical error correction. *arXiv preprint arXiv:1906.01733*, 2019.

Tahani Alsubait, Bijan Parsia, and Ulrike Sattler. Ontology-based multiple choice question generation. *KI-Künstliche Intelligenz*, 30(2):183–188, 2016.

Diana C Arroyo and Yucel Yilmaz. An open for replication study: The role of feedback timing in synchronous computer-mediated communication. *Language Learning*, 68(4): 942–972, 2018.

Emmanuel Ayedoun, Yuki Hayashi, and Kazuhisa Seta. A conversational agent to encourage willingness to communicate in the context of english as a foreign language. *Procedia Computer Science*, 60:1433–1442, 2015.

Emmanuel Ayedoun, Yuki Hayashi, and Kazuhisa Seta. Adding communicative and affective strategies to an embodied conversational agent to enhance second language learners' willingness to communicate. *International Journal of Artificial Intelligence in Education*, 29(1):29–57, 2019.

Minhui Bao. Can home use of speech-enabled artificial intelligence mitigate foreign language anxiety–investigation of a concept. *Arab World English Journal (AWEJ) Special Issue on CALL*, (5), 2019.

Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. Rasa: Open source language understanding and dialogue management. *arXiv preprint arXiv:1712.05181*, 2017.

John B Carroll. Characteristics of successful second language learners. *M., Burt, H., Dulay and M., Finocchiaro (Eds.), Viewpoints on English as a second language in honour of JE Alatis*, pages 1–7, 1977.

Jason A Chen, M Shane Tutwiler, Shari J Metcalf, Amy Kamarainen, Tina Grotzer, and Chris Dede. A multi-user virtual environment to support students' self-efficacy and interest in science: A latent growth model analysis. *Learning and Instruction*, 41:11–22, 2016.

Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 35(1):53–65, 2018.

Rod Ellis and Rod R Ellis. *The study of second language acquisition.* Oxford University, 1994.

Rod Ellis and Shawn Loewen. Confirming the operational definitions of explicit and implicit knowledge in ellis (2005): Responding to isemonger. *Studies in Second Language Acquisition*, 29(1):119–126, 2007.

Rod Ellis, Shawn Loewen, and Rosemary Erlam. Implicit and explicit corrective feedback and the acquisition of l2 grammar. *Studies in second language acquisition*, 28(2):339–368, 2006.

Hady Elsahar, Christophe Gravier, and Frederique Laforest. Zero-shot question generation from knowledge graphs for unseen predicates and entity types. *arXiv preprint arXiv:1802.06842*, 2018.

Zhihao Fan, Zhongyu Wei, Piji Li, Yanyan Lan, and Xuanjing Huang. A question type driven framework to diversify visual question generation. In *IJCAI*, pages 4048–4054, 2018.

Melanie Fleming, Pablo Riveros, Carl Reidsema, and Nicholas Achilles. Streamlining student course requests using chatbots. 2018.

L Fryer and R Carpenter. Bots as language learning tools. language learning and technology. *Language Learning & Technology*, 10(3):8–14, 2006.

Luke K Fryer, Mary Ainley, Andrew Thompson, Aaron Gibson, and Zelinda Sherlock. Stimulating and sustaining interest in a language course: An experimental comparison of chatbot and human task partners. *Computers in Human Behavior*, 75:461–468, 2017.

Luke K Fryer, Kaori Nakao, and Andrew Thompson. Chatbot learning partners: Connecting learning experiences, interest and competence. *Computers in Human Behavior*, 93: 279–289, 2019.

Andrew Gallacher, Andrew Thompson, Mark Howarth, P Taalas, J Jalkanen, L Bradley, and S Thouësny. *"My robot is an idiot!"–students' perceptions of AI in the L2 classroom.* Research-publishing. net, 2018.

Yoshiko Goda, Masanori Yamada, Hideya Matsukawa, Kojiro Hata, and Seisuke Yasunami. Conversation with a chatbot before an online efl group discussion and the effects on critical thinking. *The Journal of Information and Systems in Education*, 13(1):1–7, 2014.

Zhao Hong Han. Rethinking the role of corrective feedback in communicative language teaching. *RELC Journal*, 33(1):1–34, 2002.

Nuria Haristiani. Artificial intelligence (ai) chatbot as language learning medium: An inquiry. In *Journal of Physics: Conference Series*, volume 1387, page 012020. IOP Publishing, 2019.

Jeremy Harmer. *The practice of English language teaching*. Pearson longman, 2007.

Azirah Hashim. Crosslinguistic influence in the written english of malay undergraduates. *Journal of Modern Languages*, 12(1):60–76, 1999.

Michael Heilman and Noah A Smith. Good question! statistical ranking for question generation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 609–617, 2010.

Jiyou Jia, Yuhao Chen, Zhuhui Ding, and Meixian Ruan. Effects of a vocabulary acquisition and assessment system on students' performance in a blended learning class for english subject. *Computers & education*, 58(1):63–76, 2012.

He-Rim Kim and Glenn Mathes. Explicit vs. implicit corrective feedback. *The Korea TESOL Journal*, 4(1):57–72, 2001.

Na-Young Kim. Chatbots and korean efl students' english vocabulary learning. *Journal of Digital Convergence*, 16(2):1–7, 2018.

Na-Young Kim, Yoonjung Cha, and Hea-Suk Kim. Future english learning: Chatbots and artificial intelligence. *Multimedia-Assisted Language Learning*, 22(3):32–53, 2019.

Shaofeng Li. Student and teacher beliefs and attitudes about oral corrective feedback. *Corrective feedback in second language teaching and learning: Research, theory, applications, implications*, pages 143–157, 2017.

Michael Pin-Chuan Lin and Daniel Chang. Enhancing post-secondary writers' writing skills with a chatbot. *Journal of Educational Technology & Society*, 23(1):78–92, 2020.

Roy Lyster and Leila Ranta. Corrective feedback and learner uptake: Negotiation of form in communicative classrooms. *Studies in second language acquisition*, 19(1):37–66, 1997.

Penny Mac Donald. We all make mistakes!. analysing an error-coded corpus of spanish university students written english. *Complutense Journal of English Studies*, 24:103–129, 2016.

Lisa Margreat, J John Paul, Thusnavis Bella Mary, et al. Chatbot-attendance and location guidance system (algs). In *2021 3rd International Conference on Signal Processing and Communication (ICPSC)*, pages 718–722. IEEE, 2021.

Lidiya Murakhovs' ka, Chien-Sheng Wu, Tong Niu, Wenhao Liu, and Caiming Xiong. Mixqg: Neural question generation with mixed answer types. *arXiv preprint arXiv:2110.08175*, 2021.

Mao Nakanishi, Tetsunori Kobayashi, and Yoshihiko Hayashi. Towards answer-unaware conversational question generation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 63–71, 2019.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. The conll-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, 2014.

Rodrigo Nogueira, Jimmy Lin, and AI Epistemic. From doc2query to docttttquery. *Online preprint*, 6, 2019.

Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. *arXiv preprint arXiv:2001.04063*, 2020.

Afsaneh Rahimi and Hossein Vahid Dastjerdi. Impact of immediate and delayed error correction on efl learners' oral production: Caf. *Mediterranean Journal of Social Sciences*, 3(1):45–45, 2012.

Stephen Richardson and Lisa Braden-Harder. The experience of developing a large-scale natural language processing system: Critique. In *Natural Language Processing: The PLNLP Approach*, pages 77–89. Springer, 1993.

Melissa Roemmele, Deep Sidhpura, Steve DeNeefe, and Ling Tsou. Answerquest: A system for generating question-answer items from multi-paragraph documents. *arXiv preprint arXiv:2103.03820*, 2021.

Alla Rozovskaya and Dan Roth. Grammatical error correction: Machine translation and classifiers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2205–2215, 2016.

Alla Rozovskaya, Dan Roth, and Vivek Srikumar. Correcting grammatical verb errors. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 358–367, 2014.

Vasile Rus, Brendan Wyse, Paul Piwek, Mihai Lintean, Svetlana Stoyanchev, and Cristian Moldovan. The first question generation shared task evaluation challenge. 2010.

Troy A Smith and Daniel R Kimball. Learning from feedback: Spacing and the delay–retention effect. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 36(1):80, 2010.

Pavel Smutny and Petra Schreiberova. Chatbots for learning: A review of educational chatbots for the facebook messenger. *Computers & Education*, 151:103862, 2020.

Xingwu Sun, Jing Liu, Yajuan Lyu, Wei He, Yanjun Ma, and Shi Wang. Answer-focused and position-aware neural question generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3930–3939, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Yi Fei Wang, Stephen Petrina, and Francis Feng. Village—virtual immersive language learning and gaming environment: Immersion and presence. *British Journal of Educational Technology*, 48(2):431–450, 2017.

Joseph Weizenbaum. Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45, 1966.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45, 2020.

Ying Xu, Dakuo Wang, Penelope Collins, Hyelim Lee, and Mark Warschauer. Same benefits, different communication patterns: Comparing children's reading with a conversational agent vs. a human partner. *Computers & Education*, 161:104059, 2021.

Hui-Chun Yang and Diego Zapata-Rivera. Interlanguage pragmatics with a pedagogical agent: The request game. *Computer Assisted Language Learning*, 23(5):395–412, 2010.

Qinghua Yin and Müge Satar. English as a foreign language learner interactions with chatbots: Negotiation for meaning. *International Online Journal of Education and Teaching (IOJET)*, 2020.

Yongjing Yin, Linfeng Song, Jinsong Su, Jiali Zeng, Chulun Zhou, and Jiebo Luo. Graph-based neural sentence ordering. *arXiv preprint arXiv:1912.07225*, 2019.

Wei Yuan, Tieke He, and Xinyu Dai. Improving neural question generation using deep linguistic representation. In *Proceedings of the Web Conference 2021*, pages 3489–3500, 2021.

Zheng Yuan and Ted Briscoe. Grammatical error correction using neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–386, 2016.

Ruqing Zhang, Jiafeng Guo, Lu Chen, Yixing Fan, and Xueqi Cheng. A review on question generation from natural language text. *ACM Transactions on Information Systems (TOIS)*, 40(1):1–43, 2021.

A Appendices

A.1 Dataset

Error category 1. Required X tense, answered using Y tense

- T: What cities have you been to?
- S: *I was in so many! Pisa, Venezia, Florencia, Roma.
- T: **Yes, you have been to many places.**

- T:What was the most memorable part of this trip?
- S: *A picnic on the beach. We had some local cheese and wine, and watch the sunset together.
- T: **Oh, you watched the sunset. How romantic!**

- T: Where did you go?
- S: *I go to Beijing and Shanghai.
- T: **You went there? Which city did you like most?**

- T: And you have nothing to complain about this hotel?
- S: *No, everything was perfect! The best hotel I stay!
- T: **The best hotel you stayed at? Why?**

- T: Have you ever had a bad experience staying at an airbnb?
- S: *One time, yeah. The air conditioner break in the middle of summer. It was terrible.
- T: **Oh no, I'm sorry it broke. Did you have to change your accommodation?**

- T: So you liked all of your trips. Which countries in Europe have you been to?
- S: *I was in Hungary, Germany, Finland, Italy and France.
- T: **You have been to all of these countries! How cool!**

- T: Oh, you miss going to different places. What is your favourite cuisine?
- S: *I really like Japanese food, I wish I can go to Japan one day.
- T: **I wish I could go there too.**

- T: When was the last time you went travelling?
- S: I don't remember, probably 2 years ago.
- T: And where did you go?

- S: *I go to Croatia with my friends.
- T: **Nice! You went to Croatia. How was it?**

- T: When was the last time you went travelling?
- S: *My family and I go to Italy last summer for a couple of weeks
- T: **Oh, you went there with your family. Do you often travel with them**

- T: Yeah, I prefer going with my friends too. Where did you stay?
- S: We stayed at the hotel. It was very big and comfortable.
- T: So you stayed at the hotel, was the breakfast included in the price?
- S: *Yes, we had a very good continental breakfast that we can choose in advance.
- T: **So you could choose it. Have you tried local breakfast?**

- T: When was the last time you went travelling?
- S: I just returned from Mexico a couple of days ago.
- T: A couple of days ago? Wow! Was it a work trip?
- S: No, I was visiting my friends in Guadalajara.
- T: Did you have fun?
- S: *Yes, they showed me around the city, we have a lot of Mexican food.
- T: **Nice! You had Mexican food. What dish did you like the most?**

- T: How about you tell me about the best trip you had instead?
- S: *The best trip I had is the trip to Japan in 2017.
- T: **Oh, it was a trip to Japan. Tell me more about it!**

- T: Have you ever tried couchsurfing?
- S: *No, I didn't. I think it can be dangerous.
- T: **You haven't? Why do you think it's dangerous?**

Error category 2. Infinitive/gerund mistakes

- T: Do you always travel with the same company?
- S: *I try, because travel with my best friend is the best.
- T: **So you like travelling with your best friend.
What was the favourite trip that you've gone to together?**

- T: What activity did you like the most?
- S: *It was my first try in diving, I loved it so much!
- T: **Your first try diving? Was it scary?**

- T: Do you miss it? Travelling?

- S: Of course! *I miss go to different places and try new food.
- T: **Oh, you miss going to different places. What is your favourite cuisine?**

- T: Do you always travel with your best friend?
- S: Unfortunately not, he is busy. *But we try go on a trip at least once a year.
- T: **You try to go on a trip at least once a year? Where are you planning to go next?**

- T: What new things have you tried?
- S: *I have tried waterski and SUP.
- T: **You have tried waterskiing? Cool! Did you like it?**

- T: When was the last time you went travelling?
- S: *My family and I go to Italy last summer for a couple of weeks
- T: Oh, you went with your family. Do you often travel with them?
- S: *Sometimes, to be honest I prefer go on trips with my friends
but it is very difficult to find the time when everyone is free or on holidays
- T: **Yeah, I prefer going with my friends too. Where did you stay?**

- T: Did you go there by yourself?
- S: *Yes, I like solo travel.
- T: **You like solo travelling? Why?**

- T: Which type of accommodation do you consider to be the most convenient?
- S: *I like stay at the 5 star hotels.
- T: So you like staying at the hotels? Why?
- S: **Because you have a very comfortable room, you don't need to cook breakfast and clean.**

- T: Have you ever tried camping?
- S: *Yeah, we went camping with my classmates and some teachers
when I was in school.
- T: So, you went camping when you were at school. Did you like it?
- S: *not really, sleep on the ground was not comfortable. And I hate mosquitoes.
- T: **Yes, sleeping on the ground is uncomfortable.**

- T: What do you think are perks of travelling by plane?
- S: *Well, it fast and it can take you anywhere you want.
- T: Yes, it's fast. And have you ever taken a ferry?
- S: *No, but I really want try.

- T: **I want to try it too. What do think are the advantages of a ferry ride?**

Error category 3. Modal verbs mistakes

- T: What do you think are perks of travelling by car?
- S: *I think it's really nice if you like drive.
- T: I like driving too. I think it's fun.
- S: *No, I can't to drive. I failed my driving exam 3 times.
- T: **You can't drive? I'm sure you will be able to pass it eventually.**

- T: What else do you think is dangerous when it comes to travelling?
- S: *I think you need be very careful with valuable things like passports and money.
- T: **Why do you think you need to be careful about it?**

- T: Don't you think solo travelling can be dangerous?
- S: *In a way, because you must to pay attention always and take care of yourself.
- T: **Yes, you must pay attention. I agree.**

- T: So would you recommend this hotel to me?
- S: Of course! *You should definitely to go there! You will like it!
- T: **You think I should go there? Okay, I will!**

- T: Where are you planning to go once it is possible again? (travelling)
- S: *I like to go to Costa Rica when we can travel again.
- T: **You would like to go there? Why?**

Error category 4. Other verb mistakes

- T: Was it your first time in Europe?
- S: Yes! *I never was in Europe before!
- T: **I've never been to Europe either. Did you have fun?**

- T: Are you a light breakfast eater?
- S: *It depend. Sometimes I am very hungry and sometimes I am not.
- T: **It depends on what?**

- T: How would you describe people in Italy?
- S: *They really nice and friendly.
- T: **They are nice, yes. Have you made some friends?**

- T: And what do you think are the disadvantages of travelling by plane?
- S: *I hate when the flights delay.

- T: **You hate it when the flights are delayed? Did it happen often to you?**

- T: How was it?
- S: *It was wonderful! The best trip I have!
- T: **The best trip you've ever had? Why?**

- T: Why do you like this type of accommodation?
- S: *Because it feel like having a home, with a kitchen and a living room and everything.
- T: **Yeah, it feels like having a home.**

- T: Do you only stay at aribnbs all the time?
- S: *I try to, but if there will be no affordable options, I book a hostel.
- T: **If there are no affordable options, you mean**

- T: So you practised surfing. Are you good at it?
- S: *I am surfing only for 2 years. I have a lot to learn.
- T: **You have been surfing for 2 years? Wow! That's cool!**

- T: You liked both London and Edinburg? Was it your first time there?
- S: It was my first time in Edinburg. *But I was in London before.
- T: **You have been to London before? When?**

- T: When was the last time you went travelling?
- S: *My family and I go to Italy last summer for a couple of weeks
- T: So you went to Italy. What did you like the most about this country?
- S: Everything! *I liked the food and the people is very nice.
And I saw so many beautiful and interesting places.
- T: **Yeah, the people are very nice there. Have you made some friends?**

- T: When was the last time you went travelling?
- S: *I just returned from Mexico couple of days ago.
- T: A couple of days ago? What were you doing there?
- S: I was there on holidays.
- T: What city did you go to?
- S: *I was in Mexico City 3 days, and then in Cancun 5 days.
- T: You were in Cancun for 5 days? What did you like the most about this place?
- S: *The beach was beautiful and clean, and there was so many activities to do.
- T: **There were many activities? Like what?**
- S: I went diving and windsurfing. It was a lot of fun.

- T: Have you ever tried couchsurfing?

- S: *No, I didn't. I think it can be dangerous.
- T: You haven't? Why do you think so?
- S: *because you never know a person that host you very well.
- T: **Yes, you're right. You never know a person that hosts you.**

- T: Oh wow! Did something happen?
- S: Yes, I was walking near the river and it was slippery, so I fell and hurt my leg.
- T: Were you walking alone?
- S: *no, luckily my parents was with me and helped me.
- T: **Your parents were there? That's good.**

- T: What do you think are perks of travelling by plane?
- S: *Well, it fast and it can take you anywhere you want.
- T: **Yes, it's fast. And have you ever taken a ferry?**
- S: *No, but I really want try.
- T: And I want to try going on a helicopter ride.

Error category 5. Prepositions mistakes

- T: What was the most memorable part of the trip for you?
- S: *When we went on the top of the Tower, the view was breathtaking!
- T: **you went all the way to the top? Amazing!**

- T: Did you like the city?
- S: I don't know. To be honest, I expected more. *The pictures in internet looked so beautiful..but the place was crowded and noisy and a little dirty.
- T: **Yeah, the pictures on the internet often mislead us. So you are not recommending this place?**

- T: Was it a long trip?
- S: *No, we stayed there one week.
- T: **You mean for one week? Did you have time to go hiking?**

- T: Was it a business trip?
- S: No, I don't have business trips. *I went there for holidays.
- T: **You went there on holidays? Did you have fun?**

- T: What do you think are disadvantages of it?
- S: *Sometimes I had to wait the check in with suitcases and everything.
- T: **You had to wait for the check in? For how long?**

- T: What do you like doing together?
- S: *Many things, like shopping, going to gastronomic tours, and hiking.
- T: **Going on gastronomic tours? What kind of cuisine is your favourite?**

- T: Is the location important for you?
- S: Yes, I don't want to live very far from all the popular attractions.
*And I don't like to go with public transport.
- T: **You don't like to go by public transport? Why not?**

- T: Yeah, I prefer going with my friends too. Where did you stay?
- S: *We tayed in the hotel. It was very big and comfortable.
- T: **Nice! You stayed at the hotel. And how was the service there?**

- T: When was the last time you went travelling?
- S: I don't really remember, it was a very long time ago.
*It is difficult to travel now because covid.
- T: Yeah, because of the covid many countries have stopped accepting tourists.
Were you travelling a lot before that?
- S: *Yes, at least 2 times in a year.
- T: **Wow! Twice a year! That must have been fun! Do you miss it? The travelling?**

- T: When was the last time you went travelling?
- S: *I just returned from Mexico couple of days ago.
- T: A couple of days ago? What were you doing there?
- S: I was there on holidays.
- T: What city did you go to?
- S: *I was in Mexico City 3 days, and then in Cancun 5 days.
- T: **You were in Cancun for 5 days? What did you like the most about this place?**
- S: *The beach was beautiful and clean, and there are so many activities to do.

- T: How about you tell me about the best trip you had instead?
- S:* Probably it was the trip I made with my father when I was little.
- T: The trip you took with your father? Why did you like it so much?
- S: *We took the train to Munich. It was my first time travelling
in the train and in the city.
- T: **Is travelling by train your favourite way of transportation?**
- S: *Yes, I prefer the train because it is fast and comfortable.
- T: I prefer trains too.

- T: Have you ever tried camping?
- S: *Yeah, we went camping with my classmates and some teachers when I was in school.
- T: **So, you went camping when you were at school. Did you like it?**
- S: *not really, sleep on the ground was not comfortable. And I hate mosquitoes.
- T: Yes, sleeping on the ground is a pain.

Error category 6. Article mistakes

- T: Was breakfast included in the price?
- S: *Yes, we had buffet every morning. I loved it!
- T: **You had a buffet? How cool!**

- T: And did you always have breakfast at the hotel?
- S: *Yes, because it was so good! And I am little bit lazy to go to the cafe in the morning.
- T: **A little bit lazy? Why?**

- T: Have you ever tried staying at a hostel?
- S: *Of course, I stayed a lot of times at the hostel when I was a student.
- T: **Oh, you stayed at a hostel. Did you like this type of accommodation?**

- T: What is your favourite place in Italy?
- S: *I like Venice, it is very unique and has different atmosphere.
- T: **Yeah, it has a different atmosphere. Have you tried some local food?**

- T: What was your favourite part of the trip?
- S: *We went hiking to the National Park and climbed mountain, the view from the top of the mountain is amazing!
- T: **Oh, you climbed a mountain! How high was it?**

- T: Would you like to go there again one day?
- S: *yeah, if I have the time.
- T: **if I have time, I would like to go there too**

- T: And where did you stay there?
- S: *I stayed in the hostel, not very far from city centre.
- T: **Do you always stay not far from the city centre?**

- T: Do you always travel with your sister?
- S: *Not always, but this time we had our vacation at same time

so we decided to go somewhere together.
- T: **Is it difficult to have your vacation at the same time?**

- T: They are nice, yes. Have you made some friends?
- S: *Yes, I have met few locals. We still keep in touch.
- T: **You've met a few locals! That's great!**

- T: You mean for one week? Did you have time to go hiking?
- S: *Yes, we went on 6 hour hike near the coast line. Best hike ever!
- T: **A 6-hour hike? Wow, were you exhausted?**

- T: You try to go on a trip at least once a year? Where are you planning to go next?
- S: *I want to go to Maldives, it's a dream of mine.
- **T: to the Maldives? Why?**

- T: When was the last time you went travelling?
- S: *I just returned from Mexico couple of days ago.
- T: **A couple of days ago? What were you doing there?**
- S: I was there on holidays.

- T: Which type of accommodation do you consider to be the most convenient?
- S: *Personally, I like to book airbnb.
- T: **Why do you like to book an airbnb?**
- S: *Because you can choose a house or an apartment of your like.
- T: Yes, I agree. You can choose a house or an apartment of your liking. **Is the location important for you?**

- T: Which type of accommodation do you consider to be the most convenient?
- S: *I like stay at the 5 star hotels.
- T: So you like staying at the hotels? Why?
- S: *Because you have a very comfortable room, don't need to cook a breakfast and clean.
- T: **Yes, not cooking breakfast is convenient. Do you think 5 star hotels are on a quite expensive side?**
- S: *A little, but I like comfort. I think it is worth it and if you can afford.
- T: I'm glad you can afford it.

Error category 7. Lexical mistakes

- T: Nice! You went to Croatia. How was it?
- S: *Amazing! We spent a lot of time on the beach, did surf and swam.

- T: **So you practised surfing. Are you good at it?**

- T: Do you always stay at the hotels?
- S: *Not always, I like camping also.
- T: **I like camping too. What is your favourite thing about camping?**

- T: What did you like the most about it?
- S: *I liked the fact that there are so many different things to do and not leave the hotel.
- T: **you mean, without leaving the hotel? Yeah, I agree**

- T: What do you think are the perks of solo travelling?
- S: *Well, you can choose where to go, what to do, who to speak.
- T: **And who do you usually speak to**?

- T: Don't you think that hotels charge too much money?
- S: *Yes, very often! But I always try to find with the discount.
- T: **to find a hotel? Do they offer you big discounts?**

- T: Have you taken many pictures?
- S: *Yeah, I even finished the memory on my phone!
- T: **You ran out of memory? You must really like photography!**

- T: And how was the service there at the hotel?
- S: *It was okay, but the people work there don't speak English very well.
- T: **The people who work there? What languages could they speak?**

- T: Where would you like to go if you had a chance?
- S: *I would like to go to Thailand, my friends say to me it is a beautiful country.
- T: **Your friends told you that? What else did they say?**

- T: Any place/activity you can recommend?
- S: *I absolutely recommend you to go ride London's Eye. The view is amazing!
- T: **I would love to go for a ride, thanks.**

- T: And which city did you like the most?
- S: I don't know. *I think I liked London and Edinburg both.
- T: **You liked both London and Edinburg? Was it your first time there?**

- T: You don't like to go by public transport? Why?
- S: *It is confusing and I always lost.

- T: **you get lost? Maybe you could download Google maps.**

- T: How about you tell me about the best trip you had instead?
- S: *The best trip I had is the trip to Japan in 2017.
- T: Oh, it was a trip to Japan. Tell me more about it.
- S: *My best friend and me spent 1 month travelling all over Japan.
- T: **My best friend and I travelled to Japan too. Did you like their food?**
- S: Oh yeah, I loved it. Sushi is my favourite food.

- T: How about you tell me about the best trip you had instead?
- S: *Probably it was the trip I made with my father when I was little.
- T: **The trip you took with your father? Why did you like it so much?**
- S: *We took the train to Munich. It was my first time travelling
in the train and in the city.
- T: Is travelling by train your favourite way of transportation?

- T: Did you go there by yourself?
- S: *No, I went with my young sister.
- T: Do you often travel with your younger sister?
- S: *Sometimes, we both like the same activities, we are very same.
- T: **Oh, I see, you are alike. How old is she?**

- T: Which type of accommodation do you consider to be the most convenient?
- S: *Personally, I like to book airbnb.
- T: Why do you like to book an airbnb?
- S: *Because you can choose a house or an apartment of your like.
- T: **Yes, I agree. You can choose a house or an apartment of your liking.
Is the location important for you?**
- S: *Of course, I like to live more close to the city centre near main attractions.
- T: So, the closer to the city centre the better?

- T: Which type of accommodation do you consider to be the most convenient?
- S: *I like stay at the 5 star hotels.
- T: So you like staying at the hotels? Why?
- S: *Because you have a very comfortable room, don't need to cook
a breakfast and clean.
- T: Yes, not cooking breakfast is convenient.
Do you think 5 star hotels are on a quite expensive side?
- S: *A little, but I like comfort. I think it is worth it and if you can afford.
- T: **I'm glad you can afford it.**

Error category 8. Adjective/adverb mistakes

- T: Why do you think travelling with friends is better?
- S: *It is funner.
- T: **Oh, it's more fun. What kind of activities do you consider to be fun?**

- T: And can you think of any disadvantages?
- S: *Well, they are not fast as planes.
- T: **Yeah, they are not as fast as planes.**

- T: Did you go there by yourself?
- S: *No, I went with my young sister.
- T: **Do you often travel with your younger sister?**
- S: *Sometimes, we both like the same activities, we are very same.
- T: You are alike?
- S: Yes, we are alike

- T: Which type of accommodation do you consider to be the most convenient?
- S: *Personally, I like to book airbnb.
- T: Why do you like to book an airbnb?
- S: *Because you can choose a house or an apartment of your like.
- T: Yes, I agree. You can choose a house or an apartment of your liking.
Is the location important for you?
- S: *Of course, I like to live more close to the city centre near main attractions.
- T: **I like to live closer to the city centre too.**

Error category 9. Noun mistakes

- T: Have you seen any sea animals?
- S: *yeah, there were many little crabs on the beach, and the childs tried to catch them.
- T: **They were trying to catch them? Have you tried as well?**

- T: What do you think are the advantages of a ferry ride?
- S: *I don't know, I guess you can take your car with you, or other heavy thing.
- T: **Other heavy things? Like what?**

- T: Do you want to learn driving?
- S: *I am a little scared, there is so many traffic in my city.
- T: **I see, there is so much traffic, it's understandable.**

- T: What was so special about the trip?
- S: *It was my first time travelling abroad, I was excited and trying so many new thing.
- T: **What new things have you tried?**

- T: What was the best trip you ever had?
- S: *It is hard to choose one, I liked all my trip to Europe.
- T: **So you liked all of your trips. Which countries in Europe have you been to?**

- T: And you never take other means of transportation?
- S: *I take, I take plane and car and bus.
- T: **You do? You take planes?**

- T: I like camping too. What is your favourite thing about camping?
- S: *I love camp fire in the evenings, telling stories and having dinner.
- T: **So you love campfires. Can you start a fire by yourself?**

- T: How about you tell me about the best trip you had instead?
- S:* Probably it was the trip I made with my father when I was little.
- T: The trip you took with your father? Why did you like it so much?
- S: *We took the train to Munich. It was my first time travelling in the train and in the city.
- T: Is travelling by train your favourite way of transportation?
- S: *Yes, I prefer the train because it is fast and comfortable.
- T: **So you prefer trains. And can you think of any disadvantages?**

Error category 10. Spelling mistakes

- T: What was your favourite sightseeing spot?
- S: *I liked the old city neihbourhood.
- T: **Why this neighbourhood?**

- T: Yeah, it has a different atmosphere. Have you tried some local food?
- S: Yeah, it was delicious. *I tryed a lot of street food. It's not very healthy but I like it.
- T: **You tried street food? Is it your favourite kind of food?**

- T: Did you go there by yourself?
- S: *Yes, I like solo travel.
- T: You like solo travelling? Why?
- S: *Because I can do whatever I like, meet new interesting people and be independant.

- T: **So, for you it is important to be independent**?
- S: Yes, I value freedom a lot.