VT1

Base System for a Language Learning Chatbot

Submitted by:

Janick Michot

Matr.-Nr.: 16561276

Supervised by:

Cieliebak Mark

Submission date:

31. January 2023

Abstract [EN]

The ChaLL project aims to combine and leverage latest technologies to create a voice-based chatbot that provides learners with vital speaking opportunities in both focused and unfocused conversations and with personalized feedback. Prior to ChaLL's actual project launch, this work investigated in a possible system architecture and in the adaptation of chatbot responses according to learners' CEFR levels. While the former was purely conceptual work, the latter also involved the implementation of initial prototype solutions. Both aspects were preceded by research on comparable solutions, student proficiency levels, and the establishment of criteria for measuring language complexity.

Although there is literature on text simplification, the question of context-sensitive responses according to a CEFR level posed a novel challenge. This work's contribution is a review of related literature, a collection of requirements for this task, and an approach for personalized lexical adaption including implementation. This configurable and extensible approach uses CEFR level-dependent word lists for complex (simple) word identification and substitute selection. The substitute candidates are generated using BERT's masked-language modelling.

This research has yielded versatile results for the ChaLL project. Nevertheless, many aspects require further investigation, and evaluation. Finally, this work should be understood as a collection of ideas rather than an exact blueprint for ChaLL.

Abstract [DE]

Das Projekt ChaLL versucht neueste Technologien zu kombinieren und zu nutzen, um einen sprachbasierten Chatbot zu entwickeln, der Lernenden in freien oder aufgabespezifischen Dialogen und mit personalisiertem Feedback eine interaktive Möglichkeit zum Spracherwerb ermöglicht. Noch vor dem eigentlichen ChaLL-Projektstart wurde in dieser Arbeit eine mögliche Systemarchitektur für ChaLL und die Anpassung von Chatbot-Antworten entsprechend dem CEFR-Sprachniveau der Lernenden untersucht. Während es sich bei Ersterem um eine rein konzeptionelle Arbeit handelte, ging es bei Letzterem auch um die Umsetzung erster prototypischer Lösungen. Beidem gingen Recherchen zu ähnlichen L2-Anwendungen, zu den Leistungsniveaus von Lernenden und zur Festlegung von Kriterien für die Messung von Sprachkomplexität voraus.

Die Frage nach kontextsensitiven Chatbot-Antworten wurde bislang noch nicht untersucht. Der Beitrag dieser Arbeit ist nun eine Übersicht relevanter Literatur, eine Problemdefinition mittels Anforderungen und ein konfigurierbarer und erweiterbarer Ansatz für die personalisierte lexikalische Adaption. Dieser Ansatz verwendet Wortlisten pro CEFR-Level für die Identifizierung komplexer (einfacher) Wörter und für die Auswahl eines Ersatzwortes. Ersatzkandidaten werden mithilfe von BERT und Masked-Language Modelling generiert.

Diese Arbeit ergab vielfältige Ergebnisse für das ChaLL-Projekt. Dennoch müssen viele Aspekte weiter untersucht und bewertet werden. Schlussendlich sollte diese Arbeit eher als Ideensammlung, denn als exakter Bauplan für das ChaLL-Projekt verstanden werden.

Table of Contents

Table of Figures

List of Tables

List of Abbreviations

AS-unit	=	Analysis of Speech Unit
AE	=	Auto-Encoder
AR	=	Auto-Regressive
BFF	=	Backends-For-Frontend
CEFR	=	Common European Framework for Reference
CWI	=	Complex Word Identification
CALF	=	Complexity, Accuracy, Lexis and Fluency
CALF	=	Complexity, Accuracy, Lexis, and Fluency
CALL	=	Computer-Assisted Language Learning
E2E	=	End-to-End
LS	=	Lexical Simplification
NLP	=	Natural Language Processing
OOD	=	Open-Domain Dialog
PHZH	=	Pädagogische Hochschule Zürich
L2	=	Second Language
SLA	=	Second Language Acquisition
Seq2seq	=	Sequence-to-Sequence
STT	=	Speech-To-Text
TBTL	=	Task-Based Language Learning
TC	=	Text Complexification
TOD	=	Task-Oriented Dialog
TS	=	Text Simplification
TST	=	Text Style Transfer
TTS	=	Text-To-Speech
TTR	=	Type-Token-Ratio TTR
ZPD	=	Zone of Proximal Development

1 Introduction

This work and the associated problem definition in this chapter are based on the project "Towards a Voice-based Chatbot for Language Learners (ChaLL)" conducted by the research partners University of Zurich, Zurich University of Applied Sciences (ZHAW) and Zurich University of Teacher Education (PH-ZH). This work can therefore be considered as a sub-project, which is intended to do preliminary work for this research project. Thus, this project is technically dependent on the ChaLL project, but still has its own goals and research questions to answer. Accordingly, this work includes multiple references to the ChaLL funding application to describe the relevance, problem definition and technical aspects.

1.1 Background and Relevance

Language is an important instrument in scientific communication, business world, cultural interchanges, political issues etc. and therefore, language is widely recognized as key to success in life (Oroujlou & Vahedi, 2011). The reasons to study a foreign language are innumerable and accordingly, second language acquisition (SLA) is of the utmost importance. Hedge (2011, p. 228) characterizes speaking as one of the core skills developed in SLA and after listening the second most used skill in everyday communication. The function of language is to communicate, hence speaking is crucial. Nevertheless, a common mistake in second language (L2) learning is not paying adequate attention to speaking. Exemplary pronunciation is considered by many to be an expendable skill achieved by few, but clear and intelligible speech is an essential part of language acquisition (Thomson & Derwing, 2015). The fundamental importance of speaking lies in the intelligibility and ability to communicate (Arteaga, 2000). Pronunciation in speaking is key for L2 learners to enhance their communication skills and avoid complete communication breakdowns (Morin, 2007). Thus, the ability to communicate effectively is critical to SLA in particular and more broadly to be successful in today's global world.

1.1.1 Problems

To perform well on the international stage, students must acquire high-level foreign language skills. However, for many adult L2 learners, speaking and pronunciation does not improve even when they are exposed to L2 from native speakers for an extended period (Bajorek, 2017). Speaking competencies consequently need to be trained systematically and from an early state in the L2 process. Although language production is a highly complex process, it is often not adequately addressed in the classroom. The main challenges in teaching speaking skills to primary school students include a lack of speaking opportunities, the use of the first language (L1) as the language of instruction, a lack of

extended conversational practice, a lack of focus on speaking skills and motivation and attitudes towards language learning (Grimm et al., 2015; Pfenninger & Lendl, 2017). These challenges can lead to difficulties in acquiring the range of speaking skills needed for effective communication and may also lead to language learning anxiety and a reluctance to communicate in the second language. Traditionally, spoken interaction has been more difficult to practice outside of the classroom compared to other language skills, which exacerbates these problems.

1.1.2 Computer-assisted Language Learning

To address the problems of language learning, computer-assisted language learning (CALL) provides novel opportunities for L2 students to improve their speaking in personalized and effective ways. CALL encompasses both pedagogical issues and instructional innovations and is an important field of language education (Beatty, 2013). While CALL was initially defined as «the search for and study of the computer applications in language teaching and learning» (Levy, 1997, p. 1), the term CALL is no longer limited to formal contexts. In this sense Beatty (2013, p. 7) describes CALL as «any process in which a learner uses a computer and, as a result, improves his or her language» (Beatty, 2013, p. 7). Although CALL is not yet fully integrated into language learning (Chen et al., 2021), it has become an important component in language learning context. Anyway, as technology evolves and new devices come on the market, there are always opportunities for pedagogical advances in CALL and the development of new technical solutions for language learners. The goal of CALL, therefore, is to find ways to make the most of technology for language learning in a way that is pedagogically sound (Chen et al., 2021).

Considering today's possibilities resulting from advances in language processing, computer-assisted L2 learning, according to Golonka et al. (2014), offers a potential that no human can do: They have access to nearly unlimited knowledge, they allow focused and individual interactions, they don’t lack in patience and have unlimited time, they can send immediate and individual feedback on every utterance, they can adapt to student-led pacing and they have perfect consistency. Hence computer-assisted tools seem to have the potential to mimic communication with real speakers without the need of any human. The potential certainly depends on the choice of technologies and the implementation. Overall, CALL software can provide L2 learners with advanced, cost-effective and learner-centered tools (Bajorek, 2017).

1.1.3 Towards a Voice-based Chatbot for Language Learners (ChaLL)

According to the funding application ChaLL is envisioned as set of technologies to provide «learners with a constantly available native speaking “partner” in the form of a chatbot

which affords them context- and time-independent interactive speaking opportunities designed to develop specific speaking skills». Thus, the ChaLL project attempts to use and combine latest technologies to overcome the problems of teaching English to primary school students. As stated in the funding application ChaLL is backed by a «systematic approach to language learning with a syllabus based on the most recent SLA and task-based language teaching (TBLT) research findings» and to practice both grammatical and lexical features as well as freely and spontaneously speech, ChaLL «will provide both focused/closed and unfocused/open speaking tasks». The available language learning apps do not have a systematic teaching method and do not offer learners the opportunity to express themselves spontaneously and interactively. Thus, as described in in the funding application, ChaLL aims to «improve language learning in a fundamental way by bringing together advanced approaches in Automatic Speech Processing with context-sensitive didactic concepts for the first time» and finally to create «a safe environment in which all learners are encouraged and enabled to foster their speaking competence through "repeated exposure and use" (Kormos, 2011, p. 55), at their own pace, and with enough motivational impetus so as to facilitate real-world L2 communication».

1.2 Research Questions

This work embraces the idea behind ChaLL and aims to take the first steps towards a voice-base chatbot that will provide learners with vital speaking opportunities. The development of such a system encompasses different research areas, like Speech-to-Text (STT) for language learners, error detection in learner's utterances and generation of appropriate feedback, detecting skill-levels and adapting the chatbots responses accordingly. So, the system will be based on different Natural Language Processing (NLP) technologies, but as of yet, it is unclear whether current technologies can solve the different sub-tasks to a sufficient degree. Due to the complexity of ChaLL and the multitude of research areas that are addressed, it is not possible to examine all components equally in this work. Therefore, emphasis is placed on the overall system architecture and the task of adapting the chatbots response to a given skill-level (refer to Chapter 1.3 for more details about the focus of this work). Consequently, not all individual technologies are examined, but mainly how different technologies can be combined, to be able to implement the idea of ChaLL. This leads to the following research question of the presented work.

Research Question: How to design a system architecture for a second language learning system and adapt chatbot responses to the learner's language proficiency level to optimize the effectiveness of second language acquisition?

The main research question can be decomposed into several sub-questions according to the principle of dividing a comprehensive problem into sub-problems:

RQ1: What are some alternative L2 learning solutions, which elements of these solutions can be incorporated, and how can the proposed solution differentiate itself from these alternatives?

RQ2: What frameworks are utilized for classifying language learners according to proficiency levels and how to quantify and evaluate second language performance to establish the distinctions between these levels?

RQ3: What are the architecture requirements, and consequently, how to design an architecture to effectively support the implementation of a system like ChaLL?

RQ4: What previous work has been done in the field of text adaptation? What are the requirements for adapting chatbot responses based on a language learner's proficiency level, and to what extent is it feasible to adjust responses in accordance with a learner's skill level?

RQ5: What are the next steps to further develop the suggested system architecture and the response adaption for language learning towards the idea of ChaLL?

1.3 Scope and Delimitation

As stated, this work is related to the project ChaLL, that has not yet officially started (at the time of submission of this work) but forms the substantive background of the presented work. There are several preliminary works, including the funding application, on which this work is built. Ideally, findings from this work will flow back into the project ChaLL. Once this work was started, Quazel (see 2.1.3) was found to be a platform that already covered a considerable part of the requirements for ChaLL. Consequently, contact was sought with the developers of Quazel, which influenced the focus of this work. Initially, the goal was not simply to conceptualize the base system, but to implement it. However, since the basic system is quasi in place when collaborating with Quazel, the architecture part was limited to a conceptual investigation and instead the adaptation of chatbot responses according to a language level was included in the scope.

Although the conceptual investigation of the architecture requires to consider all the technological aspects of ChaLL, they are not all studied in detail. This would otherwise go beyond the scope of this work. Instead, an architecture is designed, with which dependencies and complications between the individual components can be shown. Collectively, a macroscopic rather than a microscopic viewpoint is taken when designing the architecture. As for the task of adapting dialog systems responses according to the learner's skill-

level, this does not involve identifying the skill level or generating a response by a dialog system.

The research questions to be answered are rather explorative than descriptive. Many aspects of the work have emerged only in the course of the work and apart from the funding proposal, little was known about the research subject in the beginning. The work is therefore approached in a relatively unbiased manner, with the aim of gaining insights, structuring the research subject and thus providing useful preliminary work for the ChaLL project. Therefore, an explorative strategy is chosen for this work.

Due to the changing focus of the work and the limited time, no effort could be made to acquire data and train data-driven models accordingly for task of response adaptation. Finally, the code developed in this work for response adaption will be a prototype and not production ready. That is, to move from the result of this work to production, it requires additional work.

1.4 Structure of the Thesis

As description of the structure of this work, this chapter also shows the scientific approach in this work. The work encompasses a prior analysis of L2 learning and proficiency and subsequent implementation of the architecture design and response adaptation.

The prior analysis of the initial situation consists of several investigations aimed at identifying other actors, L2 proficiency levels/classification, and previous research in the field of text adaption. Starting with a competitive analysis, the process involves identifying other actors, examining their products from technical and didactic perspectives, and utilizing this information to identify requirements, highlight opportunities, and ensure the unique value proposition of the system (RQ1, see 2.1). Secondly, the analysis of (measuring) proficiency levels, is required to formulate the problem of adapting the chatbot response according to a given learner's skill level (RQ2, see 2.2 and 2.3). Finally, a survey of related literature in the field of text adaptation is used to discuss and compare different approaches and to investigate what approaches would be suitable for user-sensitive chatbot response adaptation (RQ3, see 4.1). All analyses are based on a systematic literature review.

The findings from the previous analyses are then used to define requirements for both the architecture (RQ4, see 3.1) and the response adaption (RQ5, see 4.2). Based on the requirements and using an incremental approach, the architecture is designed in a stepwise manner by adding complexity and logic at every iteration (RQ4, see 3). This results in a modular and extensible architecture design that can be used as a starting point for

the ChaLL implementation. The architecture is expressed and presented both in text form and by means of sketches.

Using the knowledge from the technical analysis and based on the definition of L2 performance criteria and the classification of L2 learners, approaches for response adaptation are prototypically realized (RQ5, see 4.3). As mentioned in the delamination part of the first chapter, due to time-constraints pre-trained language models or static approaches are used instead of training a model.

The goal of this work is not to build a complete system, but rather, this work is intended to provide the first steps towards a voice-base chatbot that will provide learners with vital speaking opportunities. Therefore, an important part of this work is to critically review the results and choose the next steps (RQ6, see 3.3 and 4.4). This is also in view of the fact that the next semester work will possibly be based on this work.

2 Analysis of Language Learning

The analysis in this chapter consists of three parts, starting with a review of other actors. Subsequently, the focus in the second part is on L2 learners and reference levels used to classify learners according to their level of proficiency. Finally, criteria for automatic measurement of L2 proficiency are examined and defined for this work.

2.1 Other actors

Because of the variety in scope and purpose of language learning applications, there is no general comment about language learning applications being beneficial or not (Godwin-Jones, 2014). It all depends on the implementation. Between a simple vocabulary game completed in 5 minutes and an immersive 3D multiplayer environment played over a long period of time, there is a big difference in complexity and user engagement. Similarly, language learning applications vary in sense of their usage. There is a variety of games between educational games for classroom purpose and an online multiplayer game played in free time. Given the vast differences in scope and purpose, this chapter examines various approaches from other actors separated into L2 apps, online games, and educational games. All the observed applications can be grouped under the definition of CALL.

2.1.1 Online Games

Consistent with Beatty's definition of CALL (see 1.1.2) is the concept of naturalistic CALL which refers to «students' pursuit of some leisure interest through a second or foreign language in digital environments in informal learning contexts, rather than for the explicit purpose of learning the language» (Chik, 2013, p. 835). This definition applies in particular to online games. Whilst educational games are designed to be used in the classrooms, commercial online games are not intended as L2 learning tools. Nevertheless, they also provide opportunities in learning a foreign language and thus, there has been an increase in interest in using online games for SLA (Godwin-Jones, 2014). The fact that digital gaming is a vital part in the lives of many young people provides numerous opportunities to connect with population with limited interest in formal education (Godwin-Jones, 2014) and thus, the interest in combining gaming and L2 learning is reasonable.

Some games use STT, which allows players to enter game-specific voice commands. Here speech recognition is used to convert input via the microphone into specific words or phrases to control game elements. Thus, playing a game with foreign language voice commands included allows to use spoken language, but in a very rigid and limited way. Other

games include voice chat in addition to multiplayer functionality, which allows communication between one or more players. Jabbari & Eslami (2019) states that playing online multiplayer games in the target language «improve receptive L2 vocabulary knowledge and transform L2 learners into more resourceful communicators who venture to utilize various discourse management strategies to communicate effectively in their interactions». Unlike traditional textbooks and classrooms, players are taught cultural and linguistic knowledge in an environment that includes a variety of players with different linguistic skill level ranging from novices to experts (Godwin-Jones, 2014). In response to game events, player interactions and speech input, players are constantly formulating, repeating, revising or rephrasing statements (Godwin-Jones, 2014). Due to the repetitive nature of games, players can reinforce their vocabulary and language structures with increasing level of difficulty and complexity. Last, players are in a protected environment where they are constantly rewarded with increasing game inventory or player level. This strengthens the motivation of the gamer.

Online games may provide opportunities to use the target language for communication between players or as voice commands, but they are not intended for interactive speech and do not follow the core principles of SLA. Looking at the multiplayer games with voice chat function, the communication is indeed computer-assisted, but ultimately it takes place between individuals. Therefore, again no structure regarding SLA is guaranteed. Nonetheless, game playing can be a potential resource for long-term language maintenance or serve as an entry point to spark interest in learning new languages, as gaming has strong motivational factors (Godwin-Jones, 2014).

From a practical and pedagogical point of view, online games are less interesting for this work. The technological aspects are also of little relevance. Certain components like TTS can be found in online games, but more specific NLP-technologies used for structured SLA are not common. Games are particularly interesting when it comes to graphic design and user engagement issues of foreign language learning applications. These are topics that are not necessarily relevant during the development of the base system, but they are still worth considering.

2.1.2 Educational Games

The use of games in foreign language learning encounters numerous pedagogical and practical obstacles. This includes, according to Godwin-Jones (2014), how to incorporate language learning into gameplay and, conversely, how to incorporate gameplay and related activities back into the classroom. To address this issue, educational games, as opposed to online games, are designed specifically for educational purposes. Educational games are characterized by combining and integrating both teaching and entertainment

in simulations, and neither of these features can be omitted. Primary educational games are designed to convey specific content, yet the play aspect cannot be neglected, as it is the source of motivation in accomplishing tasks and ultimately promotes the enjoyment during play time (Godwin-Jones, 2014).

An important term in the context of educational games is gamification, which Deterding, Dixon, Khaled and Nacke (2011), refers to as «the use of game design elements in non-gaming contexts». Other authors emphasize more the importance of emotions in their definition, such as Hamari, Koivisto and Sarsa (2014) and their definition of gamification as a «process of enhancing services with motivational affordances in order to invoke gameful experiences and further behavioral outcomes». Therefore, the relevance of emotions during play time and during the learning process seems crucial to grasp the benefits of learning through game-based resources.

In addition to the digitalized versions of textbook exercises this category also includes learning apps created in collaboration with educational institutions for primary school students[1] [2] [3]. Unfortunately, most games in this category do not focus on practicing speaking. Serving Soda is a Swiss-made online educational game that focuses on spoken English output with loose objectives and has a pleasing user interface. While motivating, serving soda does not promote any particular oral abilities (Goh & Burns, 2012). It grants students points if they employ a complex term or grammatical structure. However, according to the ChaLL funding proposal «hitting a target the students do not know how to reach in the first place (learners might not understand what appropriate vocabulary is) will cause dopamine levels to spike (gamification), instead of increasing the motivation to engage in learning for the purpose of communication».

Apart from apps that are created by educational institutions, searching the web shows a multitude of less formal apps than can be used in educational settings. Often simple applications with low implementation effort are preferred and thus, digital versions of simple games can be found repeatedly[4]. Akinator[5] for exampleis a game where an online genie guesses the character, object, or animal a user is thinking of. While playing the game, students must answer questions until the Akinator guesses correctly or incorrectly, which promotes reading comprehension in particular. Another digitalized version of a

[1] https://www.thelanguagemagician.net/; https://anton.app/de/

[2] https://www.gse.harvard.edu/apps/early-literacy

[3] https://www.readingrockets.org/literacyapps/language-and-communication

[4] https://preply.com/en/blog/tut-res-esl-games-for-online-teaching/

[5] https://en.akinator.com/

simple game is Taboo[6], where one player describes something, and the other players are guessing. This enhances the vocabulary as well as listening and describing things. The goal of Scattergories[7] is to come up with words that start with a particular letter from a variety of different categories. This game is well suited for intermediate or above learners and focuses on vocabulary revision. As the system will provide focused tasks, these simple games might be reconsidered when designing tasks.

Online games are useful for teaching functional language, boosting motivation, revising target language, building routines and refreshing focus in language learning[4]. Further, online games can be helpful for teaching emergent language as students see the value of the language and start using it right away. Unlike traditional activities such as fill-in-the-blank exercise, games tend to be more engaging and motivating for students as well. For young learners, routines included in online games provide a sense of security, predictability, and something to look forward to. Finally, games can bring an element of competition and energy to a class when needed.

The main difference between these online tools and the ChaLL program is the approach to language learning. While according to the ChaLL funding application the online tools often rely on outdated, limited, and teacher-centered methods, the primary goal of the ChaLL program is to use a communicative, comprehensive, and student-centered TBLT approach. This approach focuses on providing real-world input, appropriate corrective feedback, scaffolding, and a combination of motivating open and closed tasks to ensure maximum language learning growth.

2.1.3 L2 Apps

While educational games attempt to incorporate new technologies into the classroom, the social pervasiveness of mobile digital devices and utilization of apps seems more evident in social practice. Mobile devices as enabler for new language learning approaches appears to be promising for both classroom and self-access learning. In the domain of SLA, various mobile-friendly apps can be downloaded for different operating system. Among the best known are Duolingo[8], Rosetta Stone[9], Busuu[10] and Babbel[11]. Theses apps provides great flexibility in both time and place of use. Moreover, they promise user-friendly platforms where you can access individualized learning materials.

[6] https://playtaboo.com/playpage

[7] https://scattergoriesonline.net/new-game.xhtml

[8] https://en.duolingo.com/

[9] https://www.rosettastone.eu/

[10] https://www.busuu.com/

[11] https://uk.babbel.com/

But academic research has disproven claims that app-based learning can improve oral competence (Lord, 2016). Although L2 apps are effective in fostering explicit receptive reading, vocabulary and grammar in L2, its efficiency in fostering oral competency is uncertain. For instance, Lord (2016) discovered that while novice Spanish learners in the classroom and those using Rosetta Stone achieved comparable performance in non-speaking tasks, those who used Rosetta Stone had more difficulties speaking than those who received teaching in the classroom. Additionally, Loewen et al. (2020) discovered that L2 learners' performance in the oral test components in Turkish was worse compared to the performance in the reading, writing and grammar exam components after one year of using Duolingo. A finding that was confirmed by the self-reports of the study participants about their difficulties in speaking (Loewen et al., 2020).

As conveyed in the ChaLL funding proposal L2 apps often utilize speech recognition software, but speaking practice is rarely a significant aspect of these apps and usually only consists of repetition exercises, which does not provide adequate opportunities for developing spoken skills in a second language. Other L2 programs that place more emphasis on speaking tend to only require learners to say individual words based on pictures or written prompts (Speak Easy[12]) or to read out loud pre-written dialogues (Peaksay[13]).

Another limitation indicated by Bajorek (2017) is the lack and form of targeted feedback. Whilst Rosetta Stone, Duolingo and Babel provide feedback about their utterances, most of the feedback is only binary. In addition, Rosetta Stone provides feedback as waveforms and pitch contours, but according to Bajorek (2017) the waveforms are cryptic and therefore hard to understand. Unexplained waveform feedback adds no value in improving utterances for non-linguists. In his research, Bajorek (2017) identified Babbel as the application with the best targeted feedback, due to well-performing voice recognition abilities, the explicit pronunciation instructions, and the integration of speech into vocabulary.

Despite the innovative potential, traditional L2 apps unfortunately do not help significantly in developing speaking skills. They deliver content in a modern gamified experience that engages learners. Regarding their content, they focus more on traditional L2 pedagogical lesson. Thus, L2 apps are good for learning basic vocabulary and grammar, but because of the very limited components for oral skills, they will not turn a learner into a fluent speaker. To improve app-based learning, more recent advancements with emphasis on speech opportunities needs to be implemented.

[12] https://www.gamestolearnenglish.com/speak-easy/

[13] https://www.peaksay.com/

In contrast, Quazel was created specifically to help learners improve their speaking skills and become fluent in their desired language (fluency development). It offers learners the opportunity to engage in spontaneous conversations with AI partners tailored to their abilities and provides the necessary tools and resources to support language learning. To use Quazel, learners simply need to speak into the microphone on their device in the language they are learning, and then Quazel will respond and continue the conversation using the latest language models. The Quazel user interface also provides learners with resources to look up vocabulary and get ideas for what to say. While learners are free to talk about any topic, tasks give learners goals to work towards. Tasks provide learners with specific topics to discuss with their conversation partners. When a task is completed, it can be checked off and once all tasks are completed, learners will be presented with additional broader topics to discuss.

As shown in Figure 1 Quazel's user interface allows speaking or typing the input, it has an option to speak the output slower, and it offers different types of help including help for the task, analysis of the own output, and translations into L1. While Figure 1 shows a topic from a selection of specific task topics, there is also the option to speak completely freely. Quazel seems to be designed primarily for English natives and does not have a specific mode for English language learners.

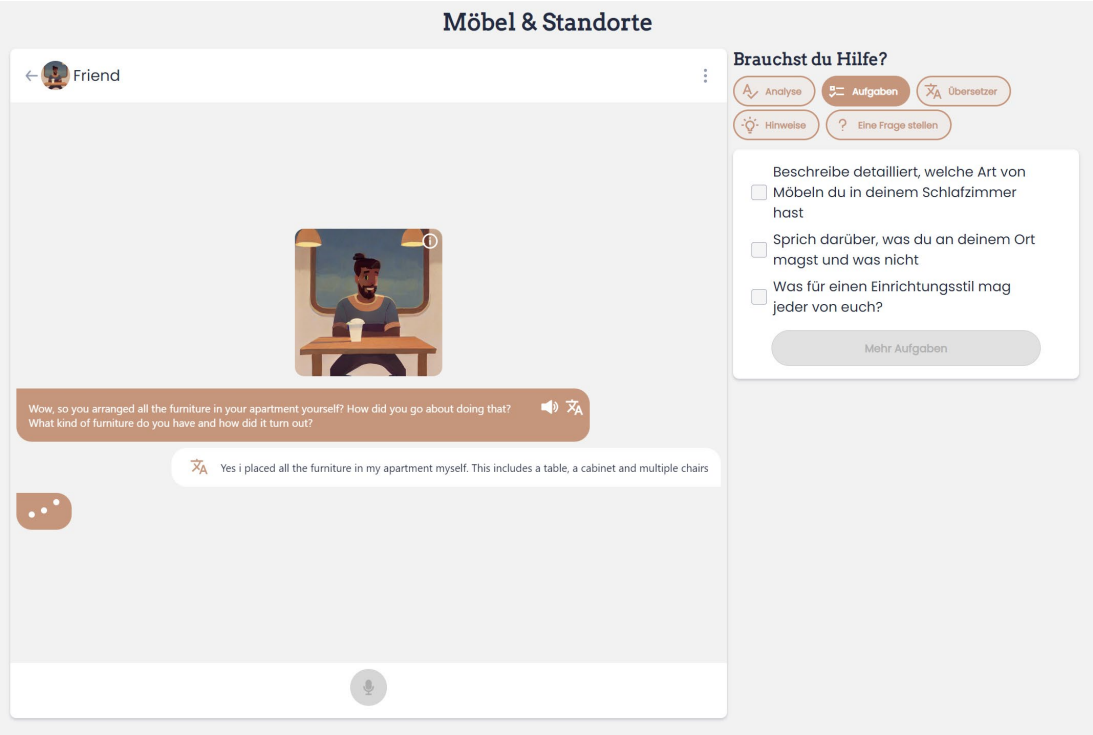

Figure 1: Quazel User Interface (accessed 16.01.2023)

Quazel seems to be programmed to correct grammatical or lexical errors in learner output by translating them into similar-sounding words. However, the responses produced by this correction are not logical or coherent and do not support targeted language development (L. Sauer, personal communication, October 7, 2022). Thus, scaffolding as the «temporary interactional support that is given to learners while their language system is under construction» (Wilson, 2008) seems to be neglected in Quazel. Event tough, Quazel appears to have already developed a considerable part of the ChaLL idea, thus enabling testing of the program and the generation of ideas.

2.1.4 Summary

CALL comes in various forms and serves different purposes. Online games, which are not specifically designed as language learning tools, can provide opportunities for language learning using voice commands or voice chat functions. While they may not follow the principles of second language acquisition (SLA) and may not be suitable for interactive speech, they can be useful for maintaining language skills or sparking interest in learning a new language. Educational games, on the other hand, are designed for the use in classrooms and thus, they tend to be more effective in teaching language skills (particularly vocabulary and grammar). But still, they lack in effectiveness in teaching pragmatic or cultural knowledge. L2 apps, which are designed specifically for language learning, can vary in their focus and approach, but generally they aim to teach language skills through structured lessons and exercises. Overall, the effectiveness of a language learning application depends on its implementation and the needs and goals of the learner.

To facilitate SLA, it is important to create a positive and supportive learning environment that helps to lower the affective filter (Du, 2009). The affective filter is a concept in SLA that refers to the emotional and attitudinal factors that can influence a learner's ability to acquire a second language (Du, 2009). Emotional and attitudinal factors, such as negative thoughts and feelings, can act as barriers to SLA (Du, 2009). On the other hand, a positive and supportive learning environment, characterized by fun, safety and relaxation, can facilitate language learning by putting learners in a more receptive mindset (Du, 2009). Therefore, educational games and L2 apps are an excellent tool to help students engage with the target language.

With Quazel there is already a tool that comes close to the idea of ChaLL. But while Quazel only offers free communication and focuses on fluency development, ChaLL concentrates on interactive speaking consisting of fluency, accuracy and complexity. Unlike Quazel, ChaLL aims to be a language learning tool that promotes both conscious, targeted language development and unconscious, open learning with a stronger didactic focus (L. Sauer, personal communication, October 7, 2022).

2.2 L2 Proficiency Levels

Following the funding application, the target group of ChaLL and this work are language learners in primary school in Switzerland. This chapter attempts to subdivide this target group according to language proficiency using reference levels. This classification of language learners is important for later instrumenting response adaptation.

2.2.1 CEFR-Levels

Addressing different types of language learners with different skill levels requires a classification of learners into categories that are in accordance with their language level. The Common European Framework for Reference (CEFR) is such a comprehensive framework that categorizes learners into different levels according to their language abilities (Council of Europe, 2001). These levels range from A1 to C2 and include numerous descriptors in various categories such as listening, reading, writing, spoken interaction and spoken production (Council of Europe, 2001). Each descriptor is defined as "Can Do" statement, to describe what L2 learners can do at each proficiency level. As such, CEFR focuses on what learner know rather than what they do not know. The CEFR defines three categories, each with 2 subcategories, leading to the following six proficiency levels in progressing order:

- *Basic User*: A1 (Breakthrough), A2 (Waystage)
- *Independent/Intermediate User*: B1 (Threshold), B2 (Vantage)
- *Proficient User*: C1 (Effective Operational Proficiency), C2 (Mastery)

The framework is not developed for a specific language and instead, the CEFR allows language-independent proficiency assessment of learners due to transparent, unbiased, and clear guidelines (Council of Europe, 2001). CEFR has been proven successful in formal tests and rating learner performances and therefore has become a key reference framework for teachers (Alderson, 2007). While the language-independence makes it a widely used reference, Alderson (2007) criticizes CEFR for using vague and imprecise language. Examples can be found in the global scale table[14] that summarizes the set of proposed common reference levels. This holistic tables states that A1 learners should be able to understand "basic phrases", B1 learners can handle "clear standard input" and C1 learners can work with "longer" texts. But "basic phrases", "clear standard input" and "longer" texts are not further quantified, leaving some questions regarding the precision of CEFR. Other criticisms include the overlapping and inconsistent levels as well as the top-down design without much adaptation to the educational context (Alderson, 2007).

[14] https://www.coe.int/en/web/common-european-framework-reference-languages/table-1-cefr-3.3-common-reference-levels-global-scale

2.2.2 Lehrplan21

In Switzerland, the EDK (2011) defines the basic skill that must be achieved by learners in various subjects, including the second national language and English. Due to Switzerland's multilingualism, the languages to be learned and the order in which they are taught vary among the cantons. But the EDK aligned a strategy that every student in Switzerland should start learning a second national language and English by the seventh grade. Independently of the language first introduced, the levels to be reached at the end of eighth and eleventh grade are identical for both languages and compulsory for all cantons. These levels and the related basic skills for foreign languages are based on CEFR (Buechel & Lichtenauer, 2019).

To harmonize the different language regions in Switzerland the EDK defined basic skills that must be reached by learners and educational regions have developed curricula in accordance with these basic skills (Buechel & Lichtenauer, 2019). The regions where German is language of instruction have developed the Lehrplan21[15], a «curriculum common to 21 cantons, which some cantons may partly adapt to their own specific needs» (Buechel & Lichtenauer, 2019). Lehrplan21 is based on common competence-oriented principles from CEFR, but because the CEFR-levels were «too vague for the definition of teaching aims», additional mid-levels (A1.1, A1.2, A2.1 etc.) were introduced (Buechel & Lichtenauer, 2019). While the curriculum is based on a communicative approach that promotes authentic situations and offers teachers freedom of methods, the standards listed for every level of school are more precise (Buechel & Lichtenauer, 2019). The expected level of learners at the end of compulsory schooling are B1.1 in writing and B1.2 in reading, speaking and listening (Buechel & Lichtenauer, 2019).

Since ChaLL aims to focus on interactive speaking, most relevant definition of descriptors for this work can be found in the speaking parts of CEFR and Lehrplan21. Table 1 combines the "Can Do" from the CEFR self-assignment grid with the "Can Do" from the Lehrplan21. The dependence of the Lehrplan21 on the CEFR is evident. Many of the aspects from the CEFR self-assignment grid reappear in a more detailed formulation in the Lehrplan21. Serving as holistic overview about what is expected from learners at each language level, it does not provide sufficient information about how a basic skill can be measured or evaluated.

[15] https://www.lehrplan21.ch/

CEFR self-assignment grid[16]		Lehrplan21[17]	
A1	Ich kann mich auf einfache Art verständigen, wenn mein Gesprächspartner bereit ist, etwas langsamer zu wiederholen oder anders zu sagen, und mir dabei hilft zu formulieren, was ich zu sagen versuche. Ich kann einfache Fragen stellen und beantworten, sofern es sich um unmittelbar notwendige Dinge und um sehr vertraute Themen handelt.	können mit ganz einfachen Worten Kontakt aufnehmen und sich verständigen (z.B. sich begrüssen, verabschieden, bedanken, etwas bestellen).	A1.1
		können sich in vertrauten Situationen auf einfache Art verständigen, wenn die Gesprächspartner/innen Rücksicht nehmen und behilflich sind (z.B. Spiel, Freizeit, Schule). können einfache Fragen zu vertrauten Themen stellen und mit einzelnen Wörtern, Ausdrücken oder kurzen Sätzen antworten (z.B. Datum, Zeit, Befinden, Farbe).	A1.2
A2	Ich kann mich in einfachen, routinemässigen Situationen verständigen, in denen es um einen einfachen, direkten Austausch von Informationen und um vertraute Themen und Tätigkeiten geht. Ich kann ein sehr kurzes Kontaktgespräch führen, verstehe aber normalerweise nicht genug, um selbst das Gespräch in Gang zu halten.	können in alltäglichen Situationen mit einfachen Worten Informationen austauschen und einholen (z.B. Rollenspiel, Gruppenarbeit). können zu vertrauten Themen einfache Fragen stellen und beantworten, kurz etwas dazu sagen oder auf Gesagtes reagieren (z.B. Unterricht, Einkauf).	A2.1
		können einfache Aussagen zu vertrauten Themen machen und darauf reagieren (z.B. etwas erklären, Verständnis prüfen). können zu alltäglichen Aktivitäten Fragen stellen und beantworten (z.B. Freizeit, Reisen, Unterricht). können ausdrücken, ob sie einverstanden sind oder lieber etwas anderes möchten (z.B. Vorschlag, Abmachung). können vertraute Personen um einen Gefallen bitten und auf Bitten reagieren (z.B. etwas ausleihen, Wunsch äussern).	A2.2
B1	Ich kann die meisten Situationen bewältigen, denen man auf Reisen im Sprachgebiet begegnet. Ich kann ohne Vorbereitung an Gesprächen über Themen teilnehmen, die mir vertraut sind, die mich persönlich interessieren oder die sich auf Themen des Alltags wie Familie, Hobbys, Arbeit, Reisen, aktuelle Ereignisse beziehen.	können zu vertrauten Themen auf einfache Art Informationen austauschen (z.B. Mode, Film, Musik). können ihre Meinung sagen und nach der Meinung von anderen fragen (z.B. Diskussion, Interview, Gruppenarbeit). können einfache Telefongespräche führen.	B1.1
		können mit Gleichaltrigen längere Gespräche über gemeinsame Interessen führen, falls diese sich um gegenseitiges Verstehen bemühen (z.B. Ferienbekanntschaft, Austauschpartner/in). können spontan Fragen stellen zu besonderen Ereignissen oder Erlebnissen (z.B. Ferien, Fest, Unfall). können in Diskussionen oder bei Entscheidungen die eigene Haltung argumentativ einbringen, Vorschläge machen und die Meinungen anderer kurz kommentieren (z.B. Projektarbeit, Wahl der Lektüre, Streitgespräch) können sich in alltäglichen Situationen beschweren (z.B. defektes Produkt).	B1.2

Table 1: Compression of Basic Skills between CEFR and Lehrplan21 [16] [17]

[16] https://www.coe.int/en/web/portfolio/self-assessment-grid

[17] https://zh.lehrplan.ch/index.php?code=a|1|21|3|1|1

2.3 Measure L2 Proficiency

While competencies have previously only ever been described as “Can Do”, this chapter introduces criteria of how L2 performance can be measured. Based on a recommendation by L. Sauer (personal communication, November 28, 2022) and in accordance with the criteria stated by Ellis and Barkhuizen (2005, p. 139) L2 performance can be measured along the dimensions of accuracy, complexity and fluency:

- Complexity: «The extent to which learners produce elaborated language» (Ellis and Barkhuizen, 2005, p. 139)
- Accuracy: «How well the target language is produced in relation to the rule system of the target language» (Ellis and Barkhuizen, 2005, p. 139)
- Fluency: «The production of language in real time without undue pausing or hesitation» (Ellis and Barkhuizen, 2005, p. 139)

While accuracy, complexity and fluency can be measured on both oral and written production, fluency for example needs to be «operationalized differently» (Ellis and Barkhuizen, 2005, p. 145). A main differentiation between spoken and written language lies in the definition and identification of a unit on which to base the analysis (Ellis and Barkhuizen, 2005, p. 147). Whilst this is easier in written language, since the unit is given by a sentence, it is more problematic in spoken language (Ellis and Barkhuizen, 2005, p. 147). Hence, with ChaLL as a tool for interactive speech, the first question is how to segment spoken text into units in a principled way.

2.3.1 Units for Measuring Spoken Language

Ellis and Barkhuizen (2005, p. 147) argue that without clear definition of an utterance, it is «impossible to achieve accurate and comparable analyses». Foster (2000, p. 365) therefore proposed analysis of speech units (AS-unit) and defined it as «a single speaker’s utterance consisting of an *independent clause* or *sub-clausal unit*, together with any *subordinate clause(s)* associated with it». Rather than intonational or semantic the unit segmentation should primarily be syntactic, as its easier to identify and thus is more reliable (Ellis and Barkhuizen, 2005, p. 147). The definition of AS units by Foster et al. (2000, p. 365) includes the following components:

- An *independent clause* is a clause that includes a finite verb.
- A *sub-clausal unit* consist of a segment of speech that can be elaborated into a full clause by recovering elided elements or a minor utterance (“Thank you”)
- A *sub-ordinate-clause* consists minimally of a finite or non-finite verb plus at least one other element. For example, “If you win the award, …”.

In spoken language, to be sure whether a subordinate clause is attached to an independent clause, the pause should not be too long (Ellis and Barkhuizen, 2005, p. 147). While the AS-unit needs to be further defined for various tasks related to spoken language like error detection or response generation, this can be neglected for response adaptation since the chatbot's output already comes in written form. Thus, the functional units are given by the sentence. There should not be a more complex segmenting into units, than splitting the response into sentences.

2.3.2 Complexity, Accuracy, Lexis and Fluency (CALF)

The criteria proposed by L. Sauer (personal communication, November 28, 2022) for this work are summarized in Table 2. This table is based on the criteria from Ellis and Barkhuizen, (2005, pp. 137 - 164). Because the lexical part has been detached from complexity the proposed categories follow a framework for operationalizing and measuring L2 performance through complexity, accuracy, lexical complexity, and fluency (CALF). To be able to make a statement about the performance under consideration of the CEFR levels, a threshold would have to be defined for each criterion and CEFR level. But defining exact thresholds is subject to further research and is not part of this work. Also, additional criteria per category are conceivable.

2.3.2.1 Complexity

Following Ellis and Barkhuizen (2005) syntactical complexity measuring can be further subdivided into interactional, grammatical and lexical complexity, with the latter handled as its own main category (see 2.3.2.4). The interactional complexity can be defined by the *number of turns* and the *mean length of the turn*. Grammatical complexity on the other hand is defined by the *amount of subordinations*, the *use of specific linguistic features* and the *mean number of verb arguments*. Measuring subordinations works well in written language, but less so in oral texts, which contain a larger number of subclausal units (Ellis and Barkhuizen, 2005, p. 155). Therefore, Ellis and Barkhuizen (2005, p. 155) supplemented the measure of grammatical complexity by the other two criteria, with *mean number of verb arguments* allowing a statement about the communicative style of the speaker. A low mean indicates a verbal style, whereas a high mean indicates a nominal style, which is considered more complex (Ellis and Barkhuizen, 2005, p. 155).

2.3.2.2 Accuracy

According to Ellis and Barkhuizen (2005, p. 153), the *percentage of error-free clauses* and the number of *errors per 100 words* are commonly used measures in language learning to assess accuracy. However, the authors note that determining what constitutes an error can be difficult and that certain questions, such as whether self-corrections should be

counted as errors, may depend on the specific research objectives. Further *self-corrections* should be considered in relation to the overall number of errors, because the more errors a speaker makes, the more self-correction opportunities he has. Finally, *target-like verbal morphology* and *target-like use of plurals* relates to grammatical accuracy. They might be relevant for focused tasks, that have been designed to elicit production of a particular linguistic feature (Ellis and Barkhuizen, 2005, p. 153).

2.3.2.3 Lexis

As for lexical complexity, which unlike Ellis and Barkhuizen (2005) is treated as a separate category, the criteria are divided into *lexical density* and *lexical diversity*. Lexical diversity measures the variety of words used, whereas lexical density measures the proportion of content words used in a text. *Lexical density* might be calculated as the ratio of functional words to lexical words. As a criterion for measuring *lexical diversity* the *type-token ratio* (TTR) can be used, which is calculated by dividing the number of unique words (types) in a text by the total number of words (tokens) in the text (Ellis and Barkhuizen, 2005, p. 155). Unfortunately, this measure is influenced by the length of the text because it is easier to get a high value in a short sentence than in a longer one.

To address the issue of text length, several measures have been proposed, one of which is the *Guiraud Index* (Guiraud, 1954), which is also a type/token-based measure that aims to be independent of the text length. The Guiraud index is calculated by dividing the number of unique words (types) by the square root of the total number of words (tokens). This method results in a higher lexical richness for longer texts than a simple TTR. However, according to Daller et al (2003, p. 200), neither TTR nor the Guiraud index are valid measures of lexical richness in later stages of L2 acquisition. An improved version of the Guiraud index is the *Advanced Guiraud Index*, which also takes frequency into account as a factor (Daller et al., 2003).

2.3.2.4 Fluency

Ellis and Barkhuizen (2005, p. 157) define two principal categories of fluency measure: *temporal variables* and *hesitation phenomena*. *Temporal variables* are related to the speed and pauses of speaking. *Hesitation phenomena* refer to the tendency of language learners to pause or hesitate when speaking in the target language and thus relates to dysfluency, which describes speech that is disordered or difficult to understand. Dysfluency can manifest in a variety of ways, including stuttering, repetitions, prolongations, blocks, and interjections. Ellis and Barkhuizen (2005, p. 157) define *False starts*, *Repetitions*, *Reformulations* and *Replacements* as dysfluency that can be measured.

Aspect	Sub-category	Measure	Source
Complexity	Interactional	Number of Turns: The total number performed by each speaker is counted. This can be expressed as a proportion of the total number of turns in the interaction. Mean Length of Turn: The total number of words/pruned words divided by number of turns	Ellis and Barkhuizen (2005, pp. 153)
	Grammatical	Amount of Subordination: The total number of separated clauses divided by total number of (AS) units. Use of some specific linguistic Feature: The number of different verb forms used. Mean Number of Verb Arguments: The total number of verb arguments (subject, direct object, indirect objects, adjectival complements, prepositional phrases) divided by total number of finite verbs.	Ellis and Barkhuizen (2005, pp. 152)
Accuracy		Number of self-corrections: The number of self-corrections as a percentage of the total number of errors committed. Percentage of error-free clauses: The number of error free clauses divided by the total number of independent clauses, sub-clausal units and subordinate clauses multiplied by 100. Errors per 100 words: The number of errors divided by the total number of words produced divided by 100. Percentage of target-like verbal morphology: The number of correct finite verb phrases divided by the total number of verb phrases multiplied by 100. Percentage of target-like use of plurals: The number of correctly used plurals divided by the total number of obligatory occasions for plurals multiplied by 100. Target-like use of vocabulary: The number of lexical errors divided by the total number of words in the text.	Ellis and Barkhuizen (2005, pp. 154)
Lexis	Diversity	Type-token ratio (Advanced) Giraud's Index (sophisticated version of type-token)	Ellis and Barkhuizen (2005, pp. 154) Guiraud (1954)
	Density	Ure Lexical Density: Proportion between the total number of lexical items and the total number of words. Halliday Lexical Density: Proportion between the total number of lexical items and the total number of clauses. Lexical items often describe the real content of a sentence including noun, verbs, adjectives and adverbs.	Ure (1971) Halliday (1989)
Fluency	Temporal variables	Speech rate: Numbers of syllables/words per second/minute Number of pauses: - Filled (ehm/eh/uhm) - Unfilled (silent) Pause length: - Total length of pauses above threshold (typically at 0.3s) - Mean length of all pauses beyond the threshold Length of run: Mean number of syllables between two pauses of a pre-determined length (threshold). This measure discounts dysfluencies.	Ellis and Barkhuizen (2005, pp. 145)
	Hesitation phenomena / Dysfluency	False start: Utterance is begun and then abandoned or reformulated. Repetition: Words/Phrases/Clauses are repeated without any modification. Self-correction/reformulation: Words/Phrases/Clauses that are repeated with some modification. Replacements: Lexical items that are immediately replaced by other lexical items.	Ellis and Barkhuizen (2005, pp. 145)

Table 2: Measuring criteria for Complexity, Accuracy, Lexis and Fluency based on a proposal by L. Sauer (personal communication, November 28, 2022)

3 System Architecture

This chapter deals with the first focus of the work, the system architecture. While an end-to-end system that directly generates the appropriate speech output from speech input is conceivable, in this work an approach is pursued in which the idea of ChaLL consists of several individual components. Developing the base system for a language learning chatbot heavily depends on the underlying architecture. One main challenge behind this work is therefore to design an architecture in which the individual technologies, some of which have yet to be developed or adapted, can be combined to achieve the idea of ChaLL.

In the first part of this chapter, the requirements for a system like ChaLL are collected. The definition of the requirements is based on information from the funding application and additional explorative research including the analysis of other actors. Considering the requirements, in the second part the suggested system architecture is explained gradually, starting with a highly simplified version displayed in Figure 3 and the final architecture in Figure 7. The proposed architectural design should be seen as a blueprint that helps to simplify the problem, that serves as a basis for future discussions and that serves as a collection of ideas. To build a production-ready system from it, complementary studies are inevitable.

3.1 System Architecture Requirements

Following the research question, the goal is to design a basic system for language learning with focus on level dependent response adaption (see Chapter 4). For other technical aspects, please refer to chapters from the funding proposal in the Appendix A. Ignoring all component-specific requirements, the overall architecture imposes the following general requirements. This list is not intended to be exhaustive and additional requirements will certainly arise.

3.1.1 Modularity

First, the architecture should be as modular as possible. As of yet, it is for certain components unclear how they are implemented and how they interact with other components. Many aspects will only become apparent when the actual implementation begins or when the system is further developed later. To prevent the architecture from having to be fundamentally adapted or even becoming unusable, it should be as modular as possible from the outset. Breaking the system into multiple components divides the complexity into loosely coupled and highly maintainable services. Every service can be developed, tested, and deployed independently of others. Finally, this breakdown reinforces the principle of separation of concerns.

3.1.2 Extensibility

Further, the architecture should be extensible so that it is possible to expand the feature set. An extensible architecture is designed from the outset for customization and enhancement. Modularity helps with extensibility, as new functionality can be achieved either by adding new components to the scope or by extending existing components. In both cases, separated modules help reduce the effort required to add new features.

3.1.3 Decoupled, Reusability

In addition to modularity, the individual components should also be as independent of each other as possible. Decoupling would mean that the components know as little as possible about each other and can therefore be developed and tested independently. Conversely, decoupled services would allow the use of components developed in other settings by simply following the defined interfaces. However, the decoupled components should be composed in some way.

3.1.4 Consistent Data Storage

Consistent data flow and storage need to be considered using a modular decoupled architecture. Opposing approaches are possible: either every component has access to a data store or only specific services. Regarding separation of concerns and maintainability, it might be more practical to have a single unit that handles data storage. This unit could simultaneously be responsible for calling all other services and providing them with the information they need.

3.1.5 Performance

Performance requirements outline system constraints like response time and latency. Since the acceptance of a system like ChaLL depends on response time, latency needs to be considered when developing the architecture. In a modular system, the total response time is composed of the summed times of the individual calls of the components. To keep this time as low as possible, the time should only be influenced by process steps that lead to the result of a request. Any processing related to data persistence or observation should not hinder the result.

3.1.6 End-to-End versus Pipeline

This point is less of a requirement and more of a basic strategy. In terms of architecture, the two approaches End-to-End (E2E) and pipeline are opposed to each other. In between, there are various mixed solutions. E2E learning involves training a single, complex model, such as a Deep Neural Network, to perform the entire task, without the use of

intermediate steps typically found in traditional pipeline designs: «This elegant although straightforward and somewhat brute-force technique [E2E] has been popularized in the context of deep learning. It is a seemingly natural consequence of deep neural architectures blurring the classic boundaries between learning machine and other processing components by casting a possibly complex processing pipeline into the coherent and flexible modeling language of neural networks» (Glasmachers, 2017).

As mentioned in the introduction to this chapter, the goal is not to develop an E2E system that directly generates speech output from speech input, but nevertheless both opposing strategies should be considered. If not for a first version of ChaLL, then perhaps for a future one. And if not for the entire system, then perhaps for individual components such as Response Adaption. Although it is assumed that all components can be learned as E2E, for certain components it may be more practical to split them into multiple parts as a pipeline. This may be for reasons of controllability or configurability.

Regarding the entire architecture, Figure 2 shows the two opposing strategies, as well as their assumed advantages and disadvantages. The advantages and disadvantages correspond mostly to the requirements discussed in this chapter.

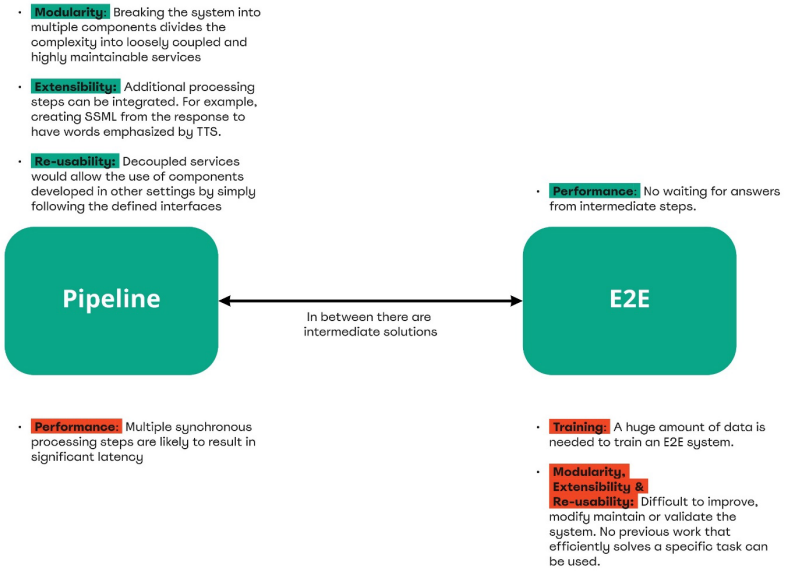

Figure 2: Pros and Cons between Pipeline and End-to-End as opposing strategies

3.2 System Architecture Results

In the following, the architecture is explained gradually, starting with a highly simplified version displayed in Figure 3 and the final architecture in Figure 11. The sketches build on each other and show the additions per chapter and iteration.

3.2.1 Service Pipeline Architecture

To achieve modularity, the system is to consist of individual components. This can be accomplished using a service-oriented architecture, an architectural style that structures an application as a collection of individual services. This architecture style is achieved by reducing the number of dependencies between services and using carefully defined interfaces trough APIs. Furthermore, this form of architecture would ensure expandability. New services can be developed separately and then added as part of the workflow. They only must adhere to the defined interfaces. Thus, logic of the language learning system can be easily supplemented or adapted using a service-oriented architecture.

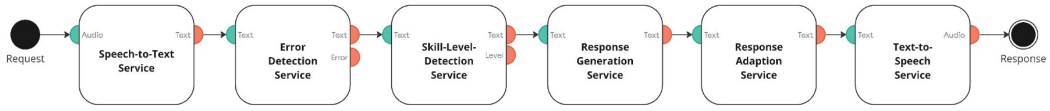

Figure 3: Basic Service Architecture Pipeline showing the simplified generation of a chatbot response to a learner's utterance.

Starting with a highly simplified version of service-oriented architecture, Figure 3 shows what the minimum viable pipeline for speech-based learning with user-sensitive responses might look like. The components in this figure align with the components from the ChaLL proposal in Appendix A. The basic workflow is as follow: (1) First the user utterances need to be voice recognized with STT. Based on the transcribed text both (2) the skill-level, and (3) errors in the learners' utterance can be detected. (4) Next the dialog system generates a response. After receiving dialog system response, (5) the text can be adapted according to the detected skill level. Lastly, the (6) text is synthesized. This fundamental and simplified setup serves as baseline for further developing the architecture.

Each of the services in Figure 3 can be understood as a component that is executed sequentially. New services can be added between two other components. For example, the translation of the simplified response into another language would be done by adding and invoking a new service between text adaptation and TTS. In this way, the result of the translation would be synthesized and not directly the simplified text. Because the individual services are piped one after another, this architecture can be referred as service pipeline architecture. Each service within this pipeline can be adapted, replaced or removed, and new services can be added. Each service is responsible for a specific part of the logic, and thus this architecture follows the principle of separation of concerns.

Using this simple pipeline-workflow visualized in Figure 3 does not consider parallelization. All services are executed sequentially, meaning that each component is waiting until

the previous has finished. Some services need to run sequentially because they are dependent on the result of other components, but many others are independent. Detecting users' skill level and error detection could, for example, be run in parallel. Therefore, an exclusively sequential pipeline is not practical in different ways. First, the latency between request and response increases when all services are executed sequentially. Latency is a decisive factor in whether the application is accepted or not. While sequential steps may be acceptable for the simplified version in Figure 3, it presents a major problem in terms of additional steps. Storing user utterances into a database, for instance, should not hinder the workflow. Instead, this should be done in parallel. Moreover, this is an example where there is no necessity to wait for a result at all. The utterance can be stored, while the workflow continues. Therefore, secondly, not every step necessarily contributes to the final results. Some processing steps are only for observation or persistence of data (utterances, vocabulary, user, etc.). In these cases, the service must be called, but there is no need to wait for their response.

3.2.2 Central Service Bus

Interfaces ensures the communication in a service-oriented architecture. But ideally, not all services communicate directly with each other. Instead, it may be better to have a central service that controls all the communication along the pipeline. This central service determines which components are linked to each other, in which order and whether the calls are synchronous are asynchronous. Further, the bus-like service handles all service-requests and -responses. In this way, no component except the central service needs to know anything about the others. A bus-like infrastructure, therefore, would reduce dependencies to a minimum. Establishing a central service bus makes it possible to connect numerous decoupled services and allow them to communicate via the bus and without dependence on or knowledge of other services.

The idea of a bus-like infrastructure is visualized in Figure 4. Building on the fundamental architecture, this architecture shows what centralized control and decentralized logic might look like. This would also streamline the integration of multiple client applications, e.g., a mobile app in addition to a web application. Each client application communicates with the bus, where the request is proceeded by calling the corresponding services. Thus, the client application does not have to know every single service and all the communication is done via the bus. Client-specific settings could still be addressed by different calls or request queries. If this is not sufficient, the division of a single central unit into several Backends-For-Frontend (BFF) would be conceivable. Anyway, using a central bus architecture fosters modularity, without abandoning a consistent workflow.

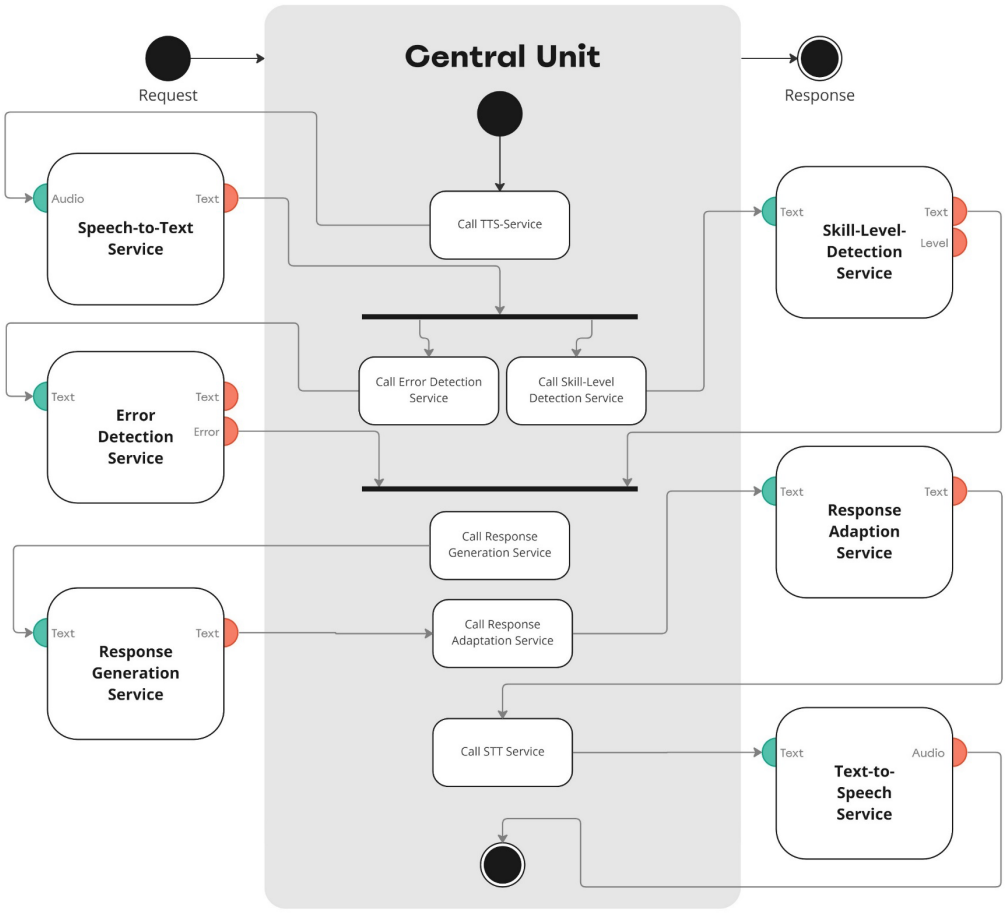

Figure 4: Service Pipeline Architecture showing the central bus-like orchestration when generating a chatbot response to a learner's utterance.

3.2.3 Data Storage and Delivery

Besides orchestration, the central unit may also be responsible for various tasks related to data storage. This would eliminate the need for each component to take care of data exchange with a database itself. Instead, the central unit can handle both the persistence and delivery of information. This enhances data integration. Not only does it prevent data from residing on different data sources, but it also prevents each component from maintaining its own calls to a database. Instead, this is managed centrally in one place. In this way, the skill-level recognition service, for example, only needs to detect the level and return a level based on a given utterance as a parameter. Any previous and further processing related to data storage is done by the central component, and the skill-level service does not need to know anything about the database. This includes, for example, saving the utterance before the request and updating user skill level after the request. However, this does not necessarily mean that individual services do not have direct access to a data store. Altogether, consistent access and delivery of data helps in terms of single responsibility, simple workflow, modularity and expandability.

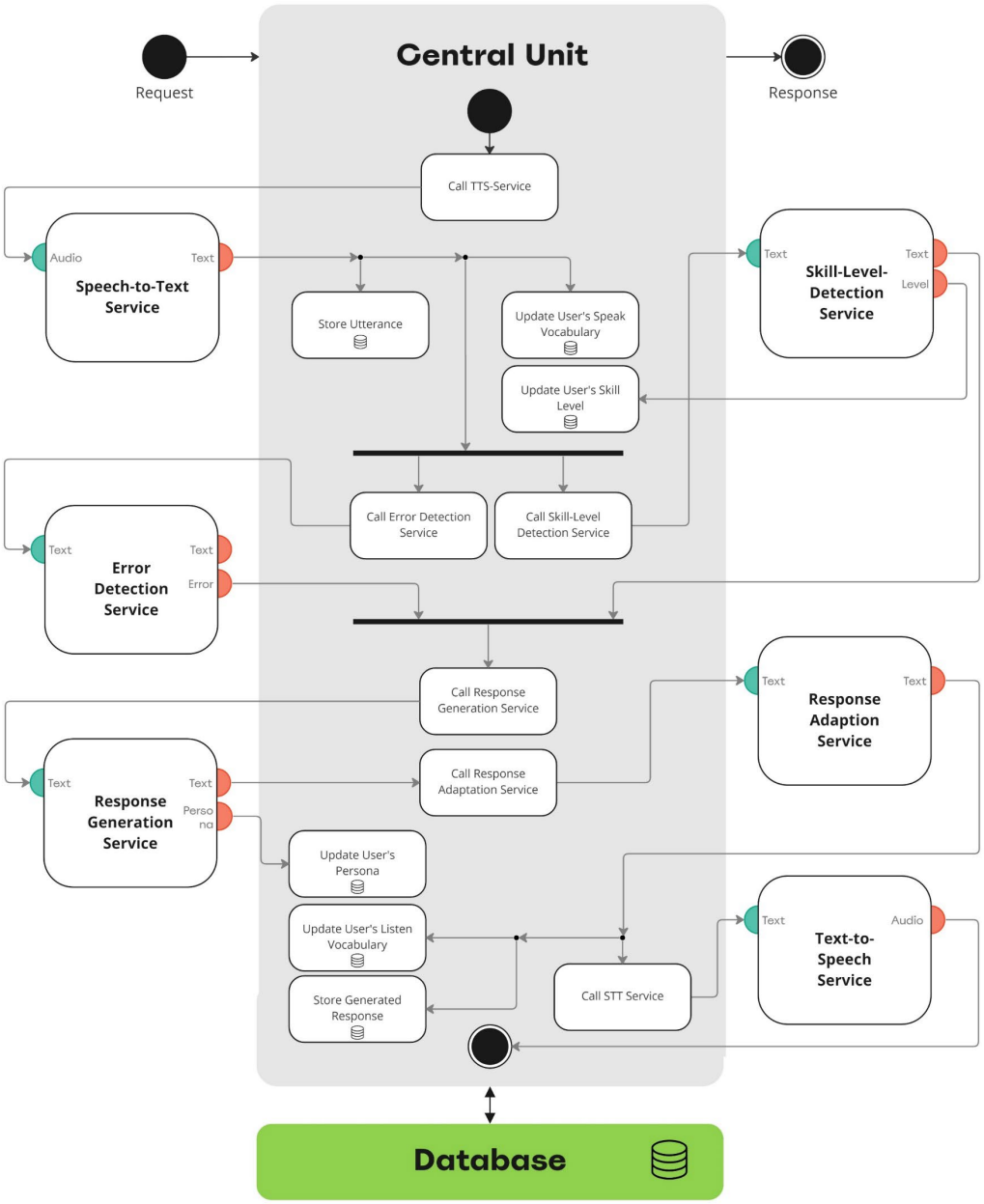

Figure 5: Service Pipeline Architecture showing the centralized data access when generating a chatbot response to a learner's utterance.

Design and implementation of a database structure are not part of this work. The following is a collection of possible information that can be stored in a database. The list is not meant to be complete, nor are all entities mandatory. Rather, the list serves as a collection of ideas related to data storage. The same ideas can be found in Figure 5. In addition to the previous sketch, this sketch shows the central data access. In this version, the only component that has access to the database is the central processing unit. Required information is passed to the services and the results of the services are stored by the central processing unit after receiving a result. The data access activities in Figure 5 are provided to illustrate procedures and logic and should not be considered necessary or final.

User: To store user-specific information beyond a session, this must be done via persistent storage. To be adaptive, storing user related data is key in a user-centric system like ChaLL. User-related information range from personal data up to users' learning profile, like skill level or tasks passed. Interests in the form of a persona are also relevant for an engaging conversation and to build proximity to the learner.

Utterances: Storing each utterance can be used by the observer (see next chapter) or for latter analysis. Both the audio format and the synthesized version of utterances could be interesting.

(Interim) Results: Storing intermediate and final results may have no immediate benefit, but this data can be of value later. Imagine a scenario where the system performs very well in computing a response to a given utterance. In the current architecture, this is achieved by many different individual services. The storage and mapping of both utterances and results can be transferred into a parallel corpus with request and response, and perhaps this corpus can be used to train a model that performs multiple tasks altogether. This is very optimistic and far ahead. But the same could maybe be applied for intermediate results, e.g., pairing dialog responses to final responses.

Vocabulary: Another possible information that could be stored is the vocabulary that the learner uses in his utterances. This information could then be used to detect the skill-level and to adapt the user's answers to this level. Lexical simplification to match the proficiency level would allow responses to be created in the learners' words. To force the learner to encounter new words, perhaps this would also allow the opposite. Instead of replacing the answer with words he has already used, the words could be replaced with synonyms he has never used. The idea of storing the user's vocabulary could perhaps include not only the vocabulary used in utterances, but also the vocabulary used in the system responses. In this way, both the words that the user said and the words that the user heard would be stored.

3.2.4 Observer Service

The central unit introduced in chapter 3.2.2 is responsible for generating a response to learners' utterances. Based on the same utterances, error detection is applied and optionally feedback is generated. Identifying errors in learners' utterances and providing appropriate and targeted feedback to help L2 learners make progress is an important function of a user-centered language learning system, but it is not the only form of feedback possible. Feedback on errors in learners' utterances means reacting to something negative, but considering the motivational factors, positive feedback would also be interesting. Conceivable positive feedback would be, for example, when a new level in the use

of vocabulary is reached or when a complex sentence structure above current learner's level has been used. Feedback boosts learners' confidence, motivation to learn and ultimately their performance. Regarding targeted feedback, both positive and negative feedback should therefore be considered.

Another differentiation of feedback is active and passive feedback. While passive feedback is given in response to an utterance, active feedback must be prompted by the learner. Feedback for errors in utterances is most probably passive, but something like prompting an analyzation of the vocabulary used in a task could be active. This differentiation of feedback is strongly dependent on the design of the frontend. Feedback can optionally be displayed once after a session or can be provided during a conversation, either by passively supplying feedback or by introducing a mechanism that allows the learner to request feedback. Further, feedback can be in written or spoken form. If you think of ChaLL as a language-based tool for improving speech production, feedback could even be packaged into a conversation itself. Assuming the learner has completed a session, a teacher-style conversation could follow to discuss positive and negative aspects. Such a conversation does not necessarily have to consist only of a monologue by the teacher, but could also consider questions from the learner. For example, "how good is my English?", "how can I improve?" etc.

Because the extent and form of targeted feedback is not yet clear, everything related to monitoring and supporting the learner is bundled under a service called "Observer". As the name suggests, the goal is to track learner's activities and to create targeted feedback based on these observations. For simplicity, a single observer is used to start, but perhaps it would be even better to have more than one observer, each taking on a particular type of observation. Regardless of the number of observers, the central unit maintains a list of all dependent observers and notifies them on certain events by calling the methods of the observers. Frequency of exchange between central unit and observer would depend on how many such events are defined. Basically, the triggering of these events could be defined by state change. This idea would follow that the central unit holds an internal state for every request, like "RecognizingSpeech" or "GeneratingResponse". Now, to automate the exchange, the notification of the observer could be triggered whenever this state changes. Thus, the observers are always informed about current processing state and can do their observing activities accordingly. States and status updates would be a simple and extensible way for communication between the observer and the central unit.

Furthermore, the observers should have access to the database. The key factor of an observer-like service is evidence. Evidence is what an observer collects during the sessions, used to rate learners' performance and refers to in feedback. Depending on the scope

and type of observation, the data included in the request might be sufficient, but in most cases additional data from the database is required. Therefore, contrary to the consistent data access and delivery mentioned in chapter 3.2.3 here, it makes sense to have direct database access. In this way, the exchange of information between the central unit and the observer could also be standardized. All specific information can be provided by the observer himself via database access. Based on the data collected and by applying clear and objective criteria, automated, targeted and evidence-based feedback is possible.

Figure 6 shows the architecture with the separate observer included. For now, the only observing task in the sketch is identifying errors in the utterance and generate targeted feedback. As described, one way in which the observer and the central processing unit might interact is using states and the updates on status changes. The states are highlighted in blue. The graph is to be understood, that the observer is informed every time such a state is updated within the processing of a request. Thus, the observer can intervene arbitrarily, and no specific calls need to be defined.

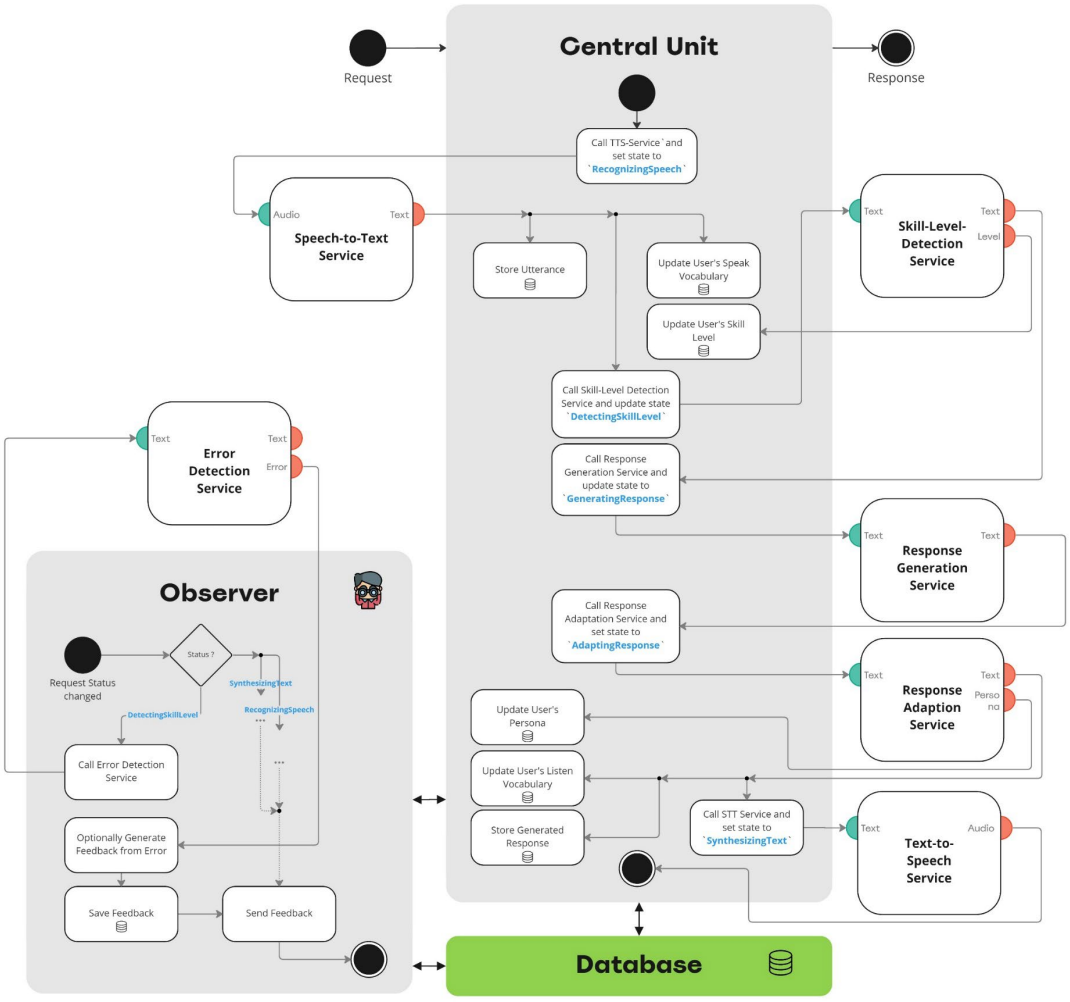

Figure 6: Service Pipeline Architecture showing the separate Observer Service when generating a chatbot response to a learner's utterance.

The way feedback is given back to the learner is not evident in Figure 6. Depending on the form and extent of the feedback, one possibility could be to add feedback to the final response of the central processing unit. This would require the observer to provide feedback to the central processing unit. This could be achieved either by expecting results on each status update call, or by making a final request to the observer just before the response is generated and sent. A more sophisticated and complex way of sending feedback to the frontend could be to use a WebSocket, with the central processing unit acting as a proxy between frontend and observer. In this way, the frontend still only needs to communicate to the central processing unit, but at the same time the observer can send feedback at any time. WebSocket connection might be initialized when starting a new conversation. The second approach using a WebSocket is added to the final sketch in Figure 11. As visualized, the central processing unit acts as a proxy for the WebSocket connection between the frontend and the observer, so that feedback can be sent directly from the observer to the frontend.

3.2.5 Open-Domain versus Task-Oriented Conversation

Until now, the service response generation is not further specified in the introduced architecture. The goal of this service is to create a response to a given utterance. However, the answer depends not only on the utterance, but also on the selected mode. ChaLL aims to provide both speaking practice in focused and unfocused talk-based conversations. Now, speech input in focused conversation needs to be handled differently from an unfocused input. In unfocused conversations, the learner is encouraged to use language freely and spontaneously as they would in real-world communicative activities, while the learner can practice and automatize specific grammatical and lexical features with focused conversation. For the architecture, this means that these two forms of conversation must be processed differently.

Focused conversation belongs to task-oriented dialog (TOD), where users solve structured tasks in a guided manner with the aim to achieve a specific goal (Zhang et al., 2020). In a language learning setting, achieving a goal means to solve an exercise, like for instance ordering a menu in foreign language or describing the furniture and locations. This are only examples of many tasks that can be offered as exercises in TOD. During this conversation, the learners are obligated to use topic related vocabulary to resolute a task.

In contrast, unfocused conversation belongs to open-domain dialog (OOD), where the aim is to «establish long-term connections with users by satisfying the human need for communication, affection, and social belonging» (Huang et al., 2020). Regarding user motivation and retention, novel transformer-based chatbots allows consistent persona across multiple chat sessions (Zhang et al., 2018) and the acquisition of new knowledge

on arbitrary topics during conversations online (Dinan et al., 2018). For open-domain conversation, a consistent persona including learners' interests and the chat history of past learning sessions ensure user engagement and trust.

Although recent approaches (Young et al., 2022) try to combine both task-oriented and open-domain dialog, most dialog systems are designed to be either focused or unfocused. Likewise, it is nearly impossible to handle both open-domain and task-specific conversation with a single chatbot. The two chatbot technologies pursue different goals, are built to carry a specific conversation, and use different knowledge bases to do so. From an architectural point of view, this means that two separate dialog systems are required to handle both modes of conversations. While this could be done in a single service, it is more appropriate to have two services. This allows to separate focused and unfocused concerns and makes it easier to maintain both dialog systems individually. Doing this split would result in Figure 7, where two new service for TOD and OOD are visualized.

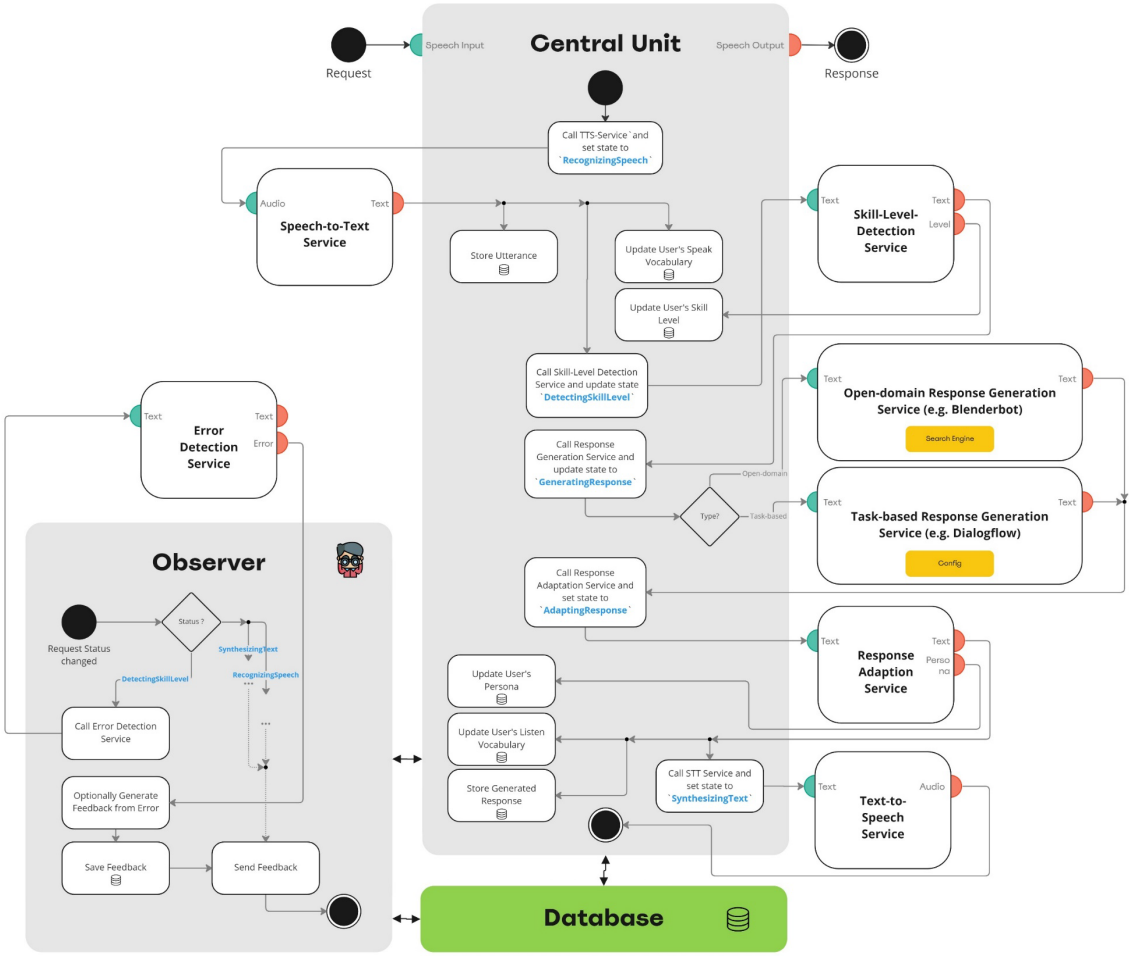

Figure 7: Service Pipeline Architecture showing the split between TOD and OOD when generating a chatbot response to a learner's utterance.

As stated in the funding proposal, it is intended to utilize existing chatbot technology to implement and explore context-sensitive responses. Efforts to give state-of-the-art transformer-based chatbots a consistent persona across chat sessions (Zhang et al., 2018) and the ability to acquire knowledge on arbitrary topics during online conversations must be particularly investigated to ensure high user motivation and retention (Dinan et al., 2018). A consistent persona, a memory of past conversations, and knowledge about a user's interest are considered key in ensuring user engagement and trust. An example of this type of chatbot is BlenderBot 2.0 (Komeili et al., 2022), which has a consistent persona, memory of past conversations, and the ability to acquire knowledge on a topic online.

Whether BlenderBot 2.0 is also suitable for task-based dialogs needs to be investigated. Maybe a more task-oriented dialog system like Dialogflow[18] is more appropriate for this purpose. Regardless of the choice of chatbot for task-based dialogs, there needs to be a way to configure tasks, as well as the possibility of tracking user's task progress and ultimately task completion. The finale response could look like the JSON example in Figure 8. It lacks the exact content, but it shows the basic breakdown of the response. The sketch in Figure 7 and its predecessors do not show how to start a conversation or session. But the process of starting and stopping would certainly involve loading and saving data related to the task and the user, as well as handling some sort of session.


```
{
    "user": {
        "persona": {
            ...
        },
        "skillLevel": "A1"
        ...
    },
    "task": {
        "name": "Furniture & Locations",
        "subtasks": [
            { "completed": true, "description": "Describe where the cabinet is placed" },
            { "completed": false, "description": "Answer how many furniture are in the room" }
        ]
        "completed": false,
        ...
    },
    "session": {
        "sessionId": "",
        "startAt": "",
        "updateAt": ""
        ...
    },
    "dialog": {
        "history": [
            { "sender": "chall", "text": "Wow. So you arrange all the furniture in your apartment..." },
            { "sender": "user", "text": "Yes. I played all the furniture..." }
        ]
    }
}
```


Figure 8: Example response in JSON format

[18] https://cloud.google.com/dialogflow/docs

3.2.6 Frontend

The frontend is the part of the system that focuses on the visual aspects and with which the learner interacts. Thus, the frontend lies above the backend and defines how to interact with the learning system via user interfaces. The frontend summarizes the code that is executed on the client side and that sends requests to the backend, where the logic is defined. More precisely, the frontend communicates through clearly defined API with the central unit of the backend. There might be different types of client applications like mobile or browser application. Depending on the clients' peculiarities, multiple centrals units may be required, where BFF could be a solution. Then, instead of a single central unit, we would have several that receive and process the client-specific requests by calling the separate services.

The development and design of the frontend is not part of this work, and here the focus is on the exchange between backend and frontend. Thus, the following two sketches are only representative for the variety of different approaches for the frontend. Since the frontend will not be pursued further in this work, these two sketches serve as illustrations when referring to the frontend from now on. Figure 9 shows a sketch of a simplified language learning frontend based on Quazel, which has many of the aspects of ChaLL integrated. This sketch shows a focused dialog, with the task displayed on the right. In the same place, other features are hinted at, and settings are implied, which will not be discussed further in this work.

To be voice-based, a language learning system is required to understand learners' speech adequately. Only when the system correctly captures what the learner is saying, a meaningful response can be generated. The process of transcribing oral speech into text also includes learners' errors because they are essential in further processing steps, such as recording the skill level. Ensuring that STT is as accurate as possible is critical since all subsequent challenges depend on it. Depending on the extent of the speech input and the form of visualizing current speech, TTS maybe needs to be separated from the pipeline. Instead, TTS can be called directly (or indirect using hub as proxy, see Figure 11) to synthesize current microphone input via WebSocket. In this way, intermediate results can be displayed to the learner so that they can see what they are saying, and furthermore, the synthesized result can then be manually adjusted and finally sent. This is how the voice input is solved in Quazel's frontend. Further, the questions remain if response and feedback generation require the synthesized text only or also the voice-based input.

Likewise, the same considerations can be made for the response: Are both the synthesized version and the text returned or should STT be completely separated from the response and controlled by the frontend. Controlling STT in the frontend could maybe be

used to speed up speech output using a stream and intermediate results. Quazel displays the text result in a chat-like interface and plays the synthesized response simultaneously.

Whether the conversation history should be displayed and the multistep (voice recording with separate send button) input is required, is left to frontend considerations. Figure 10 shows an alternative version that gives more credit to speaking, including pronunciation. Instead of displaying the conversation history, this sketch includes a 2D head that visualizes the speaking partner. Now, the idea behind this sketch is that the avatar could lip-sync the spoken text. This would enhance the social presence of an interlocutor and ideally promote pronunciation by allowing lip movements of the avatar to be perceived and adopted by the learner.

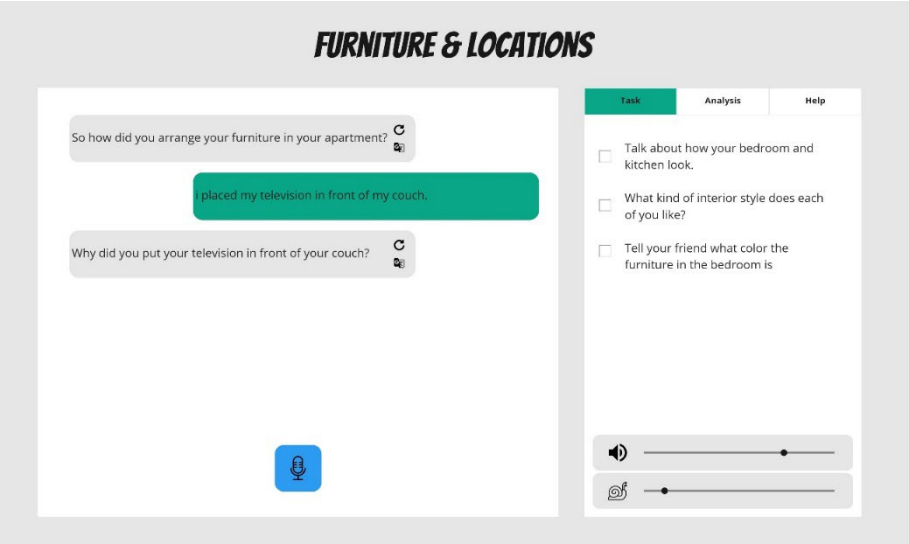

Figure 9: Quazel-like frontend sketch

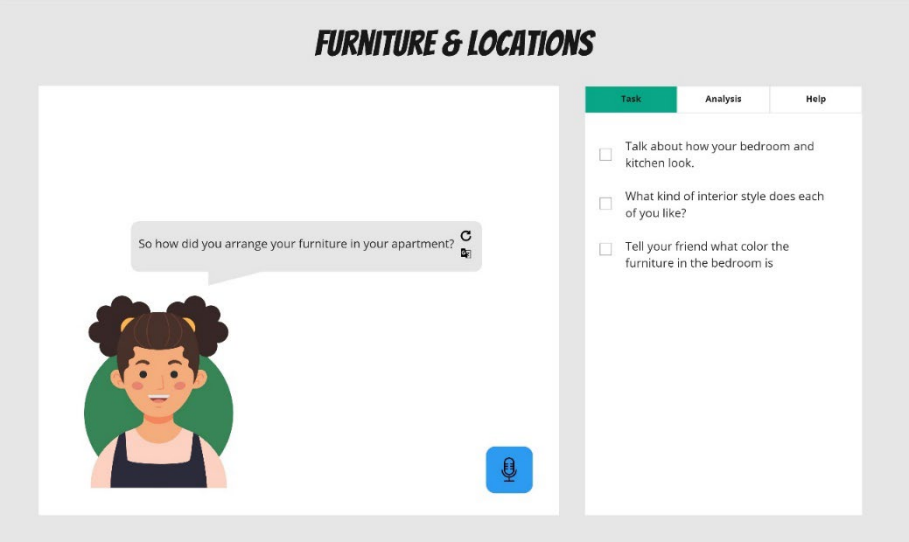

Figure 10: Quazel-like frontend sketch with lip-synced avatar

3.3 Discussion and Future Work

The contribution to the system architecture is summarized in Figure 11. A service pipeline architecture with a central unit that orchestrates and controls requests is recommended. The central unit ensures consistent data access and data flow. All requests from the frontend are handled by this unit in a bus-like manner. This promotes centralized control and decentralized logic. All activities related to the supervision and support of learners are delegated to a separate service called “Observer”. This service is informed by the central unit on status changes and has access to the database itself. This allows the observer to gather evidence during a session, from which feedback can be formulated and generated. The idea in Figure 11 is that feedback is exchanged between the frontend and the observer using a WebSocket and the central unit as a proxy. Likewise, a WebSocket could be used for speech recognition if the spoken text should be immediately visible in the frontend. As with many aspects of architecture, these are only one of several possible solutions. What and how can be adopted from the proposed architecture will only become apparent in the actual implementation. So far, the contribution of this chapter is purely conceptual and no effort for implementation has been made. Therefore, it is obvious that subsequent work includes the selection of appropriate technology and the actual implementation of the ChaLL system. However, to begin with implementation, additional work is required along many process steps that have been performed in this thesis.

This includes additional elicitation of system requirements. Even if the requirements were defined based on the ChaLL application, they correspond to more general system requirements rather than systematically elaborated project-specific requirements. In other words, additional requirements specification would be important. Regarding functional requirements, it would be interesting to elicit requirements from the user’s point of view. For example, learners and teachers could be approached with a similar L2 application, such as Quazel. Either let them use the application independently and freely or ask them to perform certain actions. In a subsequent survey, features could be collected and divided into basic factors (basics: absolutely needed), performance factors (satisfiers: assumed by learners) and excitement factors (delighters: surprises, gamification, etc.). This would ensure that the application meets the expectations of the end users.

Even though possible frontend designs based on Quazel were presented, developing the frontend requires further investigation. For example, a similar type of user assessment could be used to find the relevant frontend features. These features can then be visualized using mockups, which in turn can be presented to the target audience for another feedback loop. In addition to basic features that must be present (e.g., STT, TTS and re-

sponse generation) and the satisfiers which ensure the learning effect (e.g., response adaptation and error detection), users particularly might perceive additional features in the frontend as delighters. Since ChaLL is primarily aimed at children and young people, the design of the frontend should not be neglected. The comparison of other actors (see 2.1) has shown that games are ahead of L2 apps and educational games in terms of user motivation. The frontend design probably accounts for a significant part of this motivation.

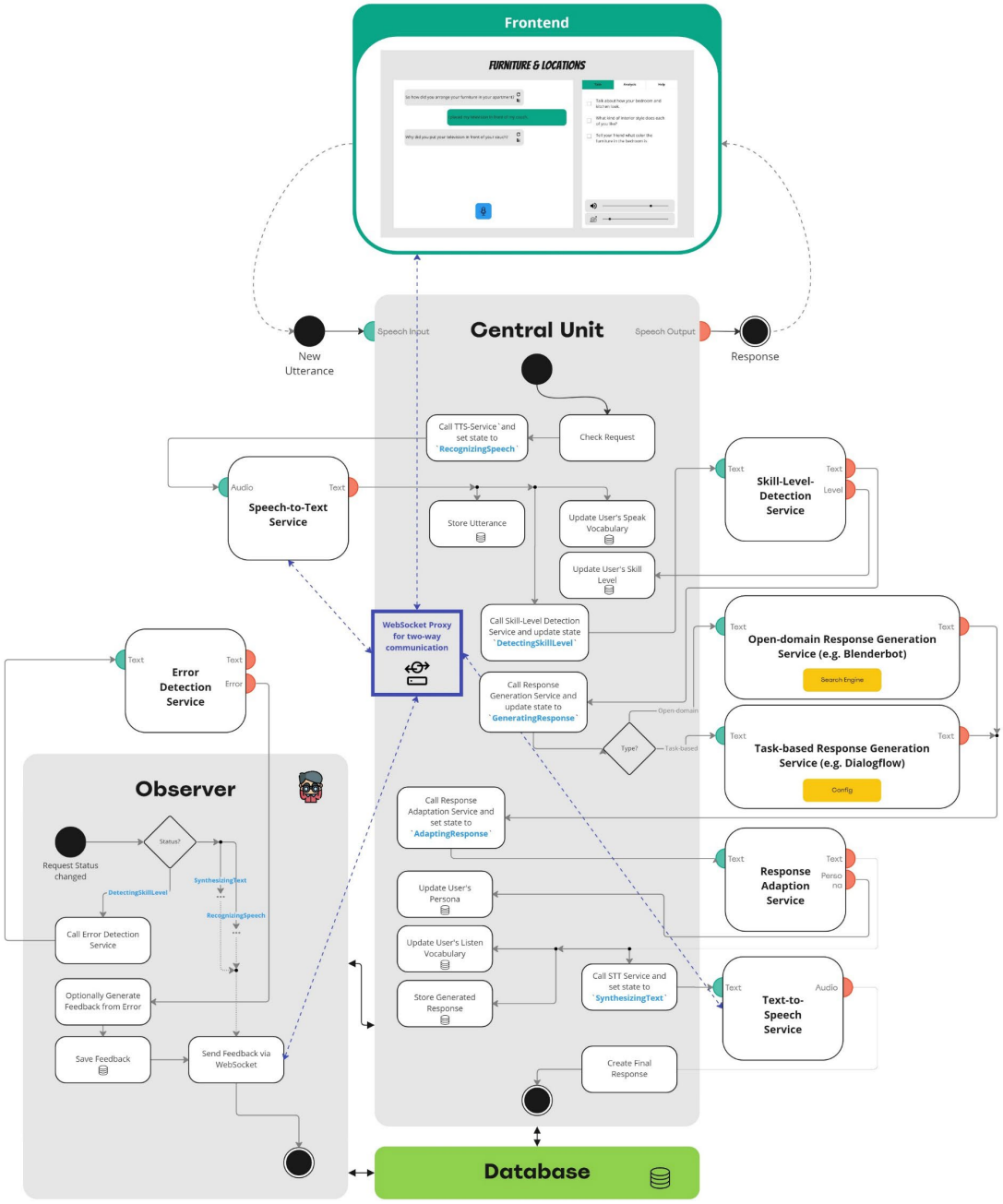

Figure 11: Service Pipeline Architecture showing the split between TOD and OOD when generating a chatbot response to a learner's utterance.

Another aspect that has only been touched in this work is data storage. It was shown what information could be stored and what the data flow and access might look like, but neither the technologies to be used nor a suitable data structure were discussed. Both are important aspects when it comes to ChaLL development. In addition, the data flow between services must be specified by defining interfaces. This, together with the centralized database access, ensures data consistency and strengths other requirements such as modularity, extensibility and reusability.

Overall, the requirements are rather superficially defined. Separate requirements and quantifiable goals would have to be defined for each service. These requirements must be specified considering all other services. It is useless, for example, if a requirement of STT is to fix any errors in utterances, when subsequently Error Detection Service checks the same text for errors. Service-specific requirements must therefore be defined, taking the entire system into account. The result of this chapter should help to maintain an overview and to understand the interrelationships.

In summary, the efforts on architecture in this thesis are only conceptual in nature. The ideas and findings have neither been implemented nor evaluated in any way. The result described in this chapter should therefore be understood as a first elaboration towards the architecture of ChaLL and not as an exact blueprint. Anyway, there is no single universal approach to architecture. Therefore, the actual ChaLL implementation may differ from the proposed pipeline architecture. To what extent the ideas can be implemented becomes apparent. In the future, the proposed pipeline architecture may even become obsolete if enough data can be collected to train an E2E system. Altogether, the results of this work should be helpful as a starting point for the ChaLL project.

4 Chatbot Response Adaptation

The second focus of this work is on adapting chatbot responses according to learners' proficiency level. Assuming the learner's skill level can be determined, this would allow responses to be adapted according to that level. In a learner-centric system, this seems crucial to address a broad target group. The system is intended to appeal to novices, as well as advanced language learners. Thus, the language used for novices must be easier understandable than for advanced learners. Otherwise, too complicated language might deter beginners and too simple language might bore advanced ones. Both seem to have a negative effect on the learning success. Consequently, an important question is, how to produce context-sensitive responses that adhere to a specific learner's CEFR level. Adapting the dialog system's response language according to a CEFR level has according to the funding proposal to date not been studied and poses a novel challenge.

Assuming that the skill level of both the learner and the chatbot response can be evaluated, three forms of response adaptation seem conceivable: No Adaption, Text Simplification and Text Complexification. However, if this assumption is supplemented by the fact that different parts of a response can correspond to different levels, the text adaptation is no longer unidirectional. Rather, some parts might need to be simplified, while other needs to be more complex. However, to introduce text simplification and text complexification in the following examples, text subdivision into multiple parts with different levels is omitted. The following text example was taken from Quazel (see Figure 1) and simplified/complexified using ChatGTP with the command "can you rewrite the following text according to CEFR level A1/B2":

Original Sentence: *"Wow, so you arranged all the furniture in your apartment yourself? How did you go about doing that? What kind of furniture do you have and how did it turn out?"*

- Response Level = Learner`s Skill Level → No adaption is required
- Response Level > Learner`s Skill Level → Text Simplification:
  "You put all the furniture in your apartment by yourself? How did you do it? What kind of furniture do you have? How does it look?"
- Response Level < Learner`s Skill Level → Text Complexification:
  "Impressive, you were able to arrange all the furniture in your apartment on your own? Can you give me some details on the process you followed? And can you tell me what types of furniture you have and how it all looks in the end?"

The idea behind response adaption is to modify the content and structure of a text so that it is easier (more challenging) to read and understand, while preserving the basic idea and approximating the original meaning. Among others, particularly foreign language learners benefit from sensitive language. The goal would be to make a text more accessible and easier to comprehend for novice learners and more challenging for advanced learners, without changing the intended meaning of the text. There are various ways that can be used to make a text simpler (more complex):

1. Replacing complex (easy) words and phrases with simpler (more complex) synonyms
2. Breaking up (Combining) long sentences into shorter (longer), simpler (more complex) sentences
3. Removing (Adding) unnecessary words and phrases
4. Using simpler (more complex) grammar and syntax
5. Providing (Omitting) definitions or explanations for complex terms

These forms of adaptation can be found in the technical investigation that follows. Further, this chapter attempts to define the chatbot response adaptation task using requirements. In the results part of this chapter, first attempts of an implementation of solutions to adapt a chatbot response according to a CEFR level are presented. This chapter concludes with a discussion and outlook.

4.1 Technical Analysis of Text Adaption

Since there are no existing solutions for response adaptation according to a proficiency level yet, this chapter examines the two possible forms of adaptation separately: Text Simplification (TS) and Text Complexification (TC). Different methods are studied for both forms of adaptation. Although, it is attempted to find all the relevant approaches for the use case of this work, the list does not claim to be exhaustive. Finally, metrics and datasets are summarized with which text adaptation is automatically evaluated and which are used in various methods.

4.1.1 Text Simplification (TS)

The process of automatically produce a simplified version of sentences and phrases is not new and has been approach by means of rule-based and statistical approaches. Rule-based approaches are typically carried out by replacing complex lexical and syntactic units with simpler ones, where each operation is performed separately in a pipeline manner (Spring et al., 2021). Statistical approaches apply machine translation methods to translate an input sentence to a simplified version (Al-Thanyyan & Azmi, 2021).

While the relationship between source and target sentences is normally one-to-one (1:1) in machine translation, automatic TS usually requires many-to-many (n:m) alignments with unaligned parts in between due to sentence splitting and compression, different order of information or additional explanations (Spring et al., 2021). Further, a single word or phrase in the original text may be replaced with multiple words or phrases in the simplified version, and vice versa. For example, a complex word or phrase in the original text may be replaced with a simpler synonym, or a long sentence may be broken down into multiple shorter sentences. Also, TS involves more elaborate transformations than machine translation, such as sentence splitting (Scarton & Specia, 2018).

4.1.1.1 Handcrafted Rule-based Text Simplification

Chandrasekar et al. (1996) were the first to introduce a rule-based approach to syntactic simplification. The goal of this approach was to decrease sentence length before using a parser. They divided the simplification process into two parts: (a) analyzing the sentence to create a structural representation and (b) using a series of rules to identify parts of the sentence that can be simplified (Al-Thanyyan & Azmi, 2021). Their research mainly focused on simplifying relative clauses, appositives and coordinated clauses (Al-Thanyyan & Azmi, 2021). They utilized handcrafted simplification rules, however, due to the basic mechanisms used to detect phrases and attachments, the system had limitations in its ability to handle certain types of sentences. Sentences that included long distances, crossed dependencies, or ambiguous phrases were not handled well by the system.

To address these weaknesses, hand-crafted systems for TS typically rely on transfer rules that work on the output of a parser (Siddharthan, 2014). Many TS systems are using a phrasal parser tree as the representation (Siddharthan, 2014). For example, De Belder & Moens (2010) used a rule-based system to simplify apposition, relative clauses, subordination and coordination by using the Stanford Parser (Klein & Manning, 2003) as their representation. Other systems write transformation rules using dependency parses as the representation. For instance, Bott et al. (2012) used dependency parsing for TS that can simplify relative clauses, coordination and participle constructions for Spanish.

4.1.1.2 Text Simplification as Sequence Labelling

Nisioi et al. (2017) presents the first attempt at using sequence-to-sequence (Seq2Seq) neural networks to model TS. Unlike previous automated TS systems, the neural TS systems are able to perform both lexical simplification and content reduction simultaneously. An extensive human evaluation of the output has shown that the neural TS systems achieve almost perfect grammaticality and meaning preservation of output sentences and a higher level of simplification than the state-of-the-art automated TS systems.

In 2017, Alva-Manchego et al. treated TS as a sequence labelling problem and identified simplification transformations at the word or phrase level. They employed the MASSAlign token-level annotation algorithm (Paetzold, Alva-Manchego, and Specia 2017) to generate annotated data, which was used to train an LSTM model to predict simplification transformations such as deletions and replacements. The output was generated by omitting words labelled for deletion, and by using the Paetzold and Specia (2017a) lexical simplifier for replacements. This approach was influenced by the abstractive sentence compression model of Bingel and Søgaard (2016), which proposed a tree labelling approach for removing or paraphrasing syntactic units in the dependency tree of a sentence using a Conditional Random Fields predictor.

4.1.1.3 Controllable Sentence Simplification

TS is often considered as «an all-purpose generic task where the same simplification is suitable for all» (Martin et al., 2019), but as for example in the use case of this work, different audiences benefit from different simplifications. Controllable TS refers to the process of making a text easier to read and understand by simplifying grammar and structure while keeping the underlying information the same but allowing for explicit control over the simplification process (Martin et al., 2019). The goal of controllable TS is to provide a simplified text that can be tailored to the specific needs of different target audiences.

Martin et al. (2019) proposed a controllable sentence simplification method that uses a discrete parameterization mechanism and a Seq2Seq model to adjust various attributes of the simplified text, such as length, amount of paraphrase, lexical complexity, and syntactic complexity. The authors show that the model can control these attributes, improving the performance of out-of-the-box Seq2Seq models. Their study focuses solely on sentence simplification, in which the input for the model is a single sentence and the output can be either one sentence or split into multiple sentences. To parametrize the Seq2Seq model they added control tokens (length=`NbChars`, paraphrasing=`LevSim`, lexical complexity=`WordRank` and syntactic complexity=`DepTreeDepth`) to the beginning of the source sentence: «The control token value is the ratio of this control token calculated on the target sentence with respect to its value on the source sentence» (Martin et al., 2019). The following is an example of the parameterization of the number of characters (22/71 = 0.3) and the Levenshtein similarity (Martin et al., 2019):

- Source: *<NbChars_0.3> <LevSim 0.4> He settled in London, devoting himself chiefly to practical teaching*
- Target: *He teaches in London.*

Martin et al. (2019) demonstrated that incorporating control tokens into Seq2Seq models improves performance for sentence simplification and that each control token effectively influences the generated simplifications. Finally, their approach can be extended with other attributes and thus can be adapted to different audiences (Martin et al., 2019).

4.1.1.4 Controllable Text Simplifications for Specific Target Audiences

Scarton and Specia (2018) describe a simplification method that involves adding artificial tokens to the beginning of sentences in a parallel corpus to encode the target language. Scarton and Specia (2018) enhanced the input to the encoder in a multilingual neural machine translation system, inspired by Johnson et al. (2017), by adding information about the intended audience and the predicted simplification transformations that should be performed. Specifically, they included an artificial token at the beginning of the input sentences to indicate the grade level (1, 2, 3 or 4) of the simplification and/or one of four possible text transformation types: identical, elaboration, splitting, or joining (Scarton & Specia, 2018). During testing, the text transformation level was either provided as an oracle label or predicted using a simple features-based Naive Bayes classifier (Scarton & Specia, 2018):

- number of tokens / punctuation / content words / clauses
- ratio of the number of verbs / nouns / adjectives / adverbs / connectives to the number of content words

Using the standard neural architecture in OpenNMT and data from the Newsela corpus, the researchers found that adding this extra information resulted in improved BLEU and SARI (Scarton & Specia, 2018). Further, the approach allows for zero-shot machine translation, where a single model can translate between language pairs that it has not seen during training (Scarton & Specia, 2018).

4.1.1.5 Controllable Text Simplification with Lexical Constraint Loss

Nishihara et al. (2019) proposed a method for controlling the grade level of a sentence in TS tasks. The study uses the grade levels of the US education system as a measure for sentence complexity and not only considers the sentence level but also the word level complexity: «Sentence level complexity is determined by adding the target grade level as input, while word level complexity is determined by adding weights to the training loss based on words that frequently appear in sentences of the desired grade level» (Nishihara et al., 2019). They use two methods to calculate the weight of the words, TF-IDF and pointwise mutual information (PPMI). TF-IDF is computed by looking at the probability that a word appears in a set of sentences of grade level *l* and PPMI is calculated by looking at the strength of co-occurrence between the word and the level. Both methods have a

range of [0,1) and Laplace smoothing is applied. By doing so, Nishihara et al. (2019) consider words «which frequently appear in text of a specific grade level». The experiment results show that the proposed method improves both BLEU and SARI metrics.

4.1.1.6 Controllable Text Simplification with Explicit Paraphrasing

While the TS domain is predominated by Seq2Seq models that are trained end-to-end to perform simplifications as lexical paraphrasing, deletion, and splitting simultaneously, the Maddela et al. (2020) suggest a hybrid approach for controllable TS with explicit paraphrasing. In this hybrid approach, they combine linguistically motivated rules for splitting and deletion with a neural paraphrasing model to generate different paraphrasing styles. The ability to control the degree of each simplification operation applied to the input texts seems interesting for content-sensitive response adaption.

Figure 12 shows an overview of Maddela et al. (2020) hybrid approach. For the rule-based part they used the state-of-the-art tool DisSim (Niklaus et al., 2019) that focuses on structural simplification by splitting and deleting. To simplify a complex English sentence, DisSim uses 35 manually crafted grammar rules to break it down into a set of hierarchically organized sub-sentences. These sub-sentences are candidates for further processing, except for those that are very short or long (Maddela et al., 2020). To increase the variety of candidates generated by DisSim, they supplemented a neural deletion and split module trained on a TS corpus. Conclusively, given an input sentence, the candidate generation creates a set of intermediate simplifications that have underdone splitting and deletion.

Maddela et al. (2020) designed a neural ranking model to score all the candidates that have undergone splitting and deletion. The model was trained on a standard TS corpus of pairs of complex sentences and their corresponding manually simplified versions (Maddela et al., 2020). The scoring function for each candidate during training was a length-penalized BERT score, which uses BERT embeddings to find soft matches between word pieces instead of exact string matching (Maddela et al., 2020). The ranking model was trained in a pairwise setup, minimizing ranking violations through hinge loss (Maddela et al., 2020). Features used in the model included the number of words, compression ratio, Jaccard similarity, applied rules, and number of rule applications for the input and candidate sentences (Maddela et al., 2020).

Finally, they feed the top-ranked candidates to a lexical paraphrasing model for the final output (Maddela et al., 2020). The top-ranked candidate from the previous step is paraphrased using a Transformer encoder-decoder model initialized with a BERT check-

point (Rothe et al., 2020). The model controls the extent of lexical paraphrasing by specifying the percentage of words to be copied and uses an attention-based copy mechanism to encourage paraphrasing while staying faithful to the input (Maddela et al., 2020).

Tuneable settings in both the candidate generation (controlling operation focus and candidates selection based on length or splits) and paraphrase generation (copy ratio to control the degree of paraphrasing) provide control over how much of the input is changed and thus, Maddela et al. (2020) showed that «the model can control various attributes of the simplified text, such as number of sentence splits, length, and number of words copied from the input».

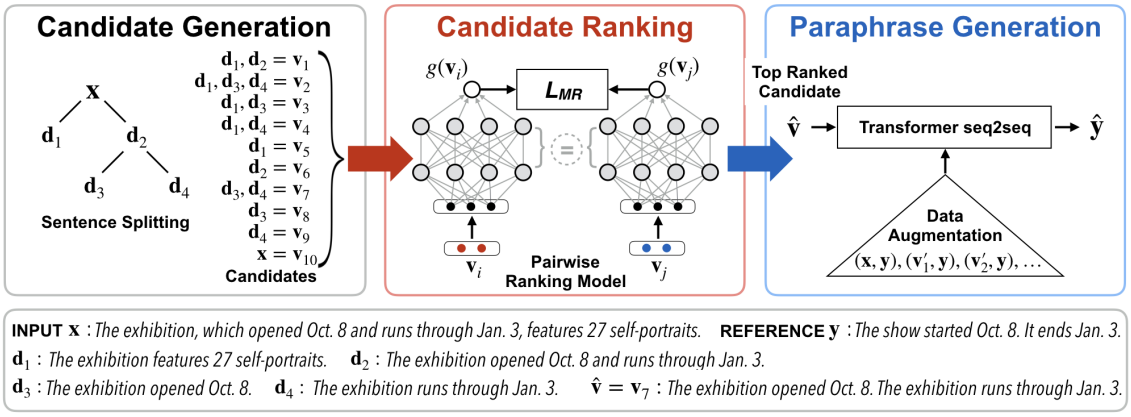

Figure 12: Overview of the proposed model for text simplification, which can perform a controlled combination of sentence splitting, deletion, and paraphrasing by (Maddela et al., 2020).

To understand the errors generated by their hybrid model Maddela et al. (2020) defined to following categories of errors: **Hallucinations**, where the model introduced information not in the input, **Fluency Errors**, where the model generated ungrammatical output, **Anaphora Resolution**, where it was difficult to resolve pronouns in the output and **Bad Substitution**, where the model inserted an incorrect simpler phrase.

4.1.1.7 Text Simplification for Specific CEFR-Level

Spring et al. (2021) were able to demonstrate multi-level TS for standard German language into levels A1, A2 and B1 by establishing strong baselines on a generic simplification task. Further, they were able to boost model performance for specific levels of simplification using source-side labels and a pretraining/fine-tuning strategy. They used source-side-labels as additional information that is provided to the translation system about the source text in the form of tags indicating the desired CEFR level of the target segment (<b1>, <a2>, and <a1>). They used a collection of news articles from the Austria Press Agency that are simplified into two language levels (B1 and A2). The results showed that using source-side labels and pretraining generally improves scores on the BLEU and

SARI metrics for the target language, but combining these approaches does not always result in better scores. Especially for the CEFR level A2. However, for CEFR levels A1 and B2 (which have fewer data available) these approaches are more effective in improving scores.

Moreover, they introduce the concept of copy labels. In the context of TS, a "copy" refers to a segment of text that is identical to the original text. This can occur when the TS process produces a simplified version that is the same as the original text, without any changes or modifications. Spring et al. (2021) noted higher numbers of direct copies for higher CEFR levels, because simplifications on these levels tend to be closer to the original text and therefore do not require many changes. By «using an explicit <copy> label instead of the CEFR level for all segments where source and target were identical», they were able to reduce the number of directly copies, because these labels help the model to distinguish between sentences that require further modification and sentences that can be accepted unchanged. However, the copy-labelled models did not consistently outperform their counterparts in terms of BLEU and SARI scores (Spring et al., 2021).

4.1.1.8 Lexical Simplification

Lexical Simplification (LS) is a subtask of TS and aims to make a text simpler by substituting difficult words with simpler ones, while preserving its meaning and grammaticality (Lee & Yeung, 2018). LS systems typically use a pipeline architecture (Paetzold & Specia, 2016). The pipeline begins with 1) Complex Word Identification (CWI) to find target words to be simplified, followed by a 2) Substitution Generation component to generate substitute candidates for these complex words. 3) Substitution Selection discards candidates that may distort the meaning of the text or affect its grammaticality, and 4) Substitution Ranking determines the best output by ranking the remaining candidates by simplicity. Figure 13 summarizes these pipeline steps.

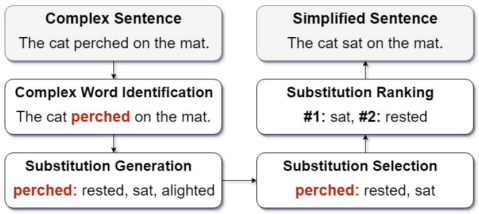

Figure 13: Lexical Simplification pipeline (Paetzold & Specia, 2017b)

Each step in the LS pipeline represents a separate NLP task, which will not be discussed in detail here. Paetzold and Specia (2017b) conducted a survey in which they evaluated the literature for each step in the typical LS pipeline and benchmarked existing approaches for these steps on publicly available datasets.

4.1.1.9 Personalized Lexical Simplification

Most LS systems first perform CWI to detect complicated target words, and then find and rank appropriate substitutions for them. According to Lee & Yeung (2018) most of the LS systems «assume one best substitution or one fixed ranked list of substitutions» and do not consider variations in vocabulary among users. To address «the expected heterogeneity among non-native speakers with different language backgrounds and proficiency levels» (Paetzold & Specia, 2016), Lee & Yeung (2018) argue for the use of personalized CWI to «tailor lexical simplification systems to the vocabulary proficiency of the user». To do so, they add personalization to both the CWI and Substitution Ranking task of the LS pipeline: detect for a user which words require simplification and reject substitution candidates that are still too difficult for the user.

To create personalized lexical simplification datasets, Lee & Yeung (2018) used the BenchLS dataset containing 929 instances of target words and their substitutions, annotated by English speakers. Based on a rating of 12,000 English words on a five-point scale and a subdivision of this word rating into complex and non-complex, a modified version of the BenchLS dataset was created for each participant in a two-step approach: 1) «to detect which words require simplification», and 2) «to reject substitution candidates that are still too difficult for the user». Lee and Yeung (2018) demonstrated that models that utilized personalized datasets outperformed user-independent models in terms of both the number of unnecessary simplifications and the number of complex words in the output. Maybe a similar approach can be applied creating personalized substitution lists per CEFR level based on word lists per level.

4.1.1.10 BERT-based Lexical Substitution

Previous research on lexical substitution typically obtains substitute candidates by identifying synonyms of the target word from lexical resources (such as BenchLS) and ranking the candidates based on the context in which they appear. These approaches have two limitations: (1) They may not consider good substitute candidates that are not synonyms of the target word in the lexical resources, and (2) They do not account for the influence of the substitution on the overall context of the sentence (Zhou et al., 2019).

To address these limitations, Zhou et al. (2019) propose an end-to-end BERT-based lexical substitution approach that can propose and validate substitute candidates without the use of annotated data or manually curated resources. Their approach first applies dropout to the target word's embedding, which partially masks the word and allows BERT to consider the target word's semantics and context equally when proposing substitute candidates. Next, candidates are validated based on their substitution's influence on the

overall contextualized representation of the sentence. Zhou et al. (2019) demonstrated that the approach performs well in both proposing and ranking substitute candidates.

Instead of applying the dropout mechanism to the complex word's embeddings for partially masking the word, Qiang et al. (2020) proposes to mask the complex word of the input sentence. This approach generates candidates of the complex word by «considering the whole sentence that is easier to hold cohesion and coherence of a sentence». While both approaches of Zhou et al. (2019) and Qiang et al. (2020) only focus on substitute generation, the new research of Qiang et al. (2021) includes CWI, substitute generations, and substitute ranking.

Qiang et al. (2021) treat CWI as a sequence labeling task, and fine-tuned BERT to predict the word complexity. For substitution generation, Qiang et al. (2021) used BERT's masked-language modeling and replaced the complex word in a sentence with the special symbol "[MASK]". BERT then provides the probability distribution corresponding to the masked word. Because BERT «is very likely to generate substitute candidates that are semantically different from the complex word, although it considers the context», they concatenated the original sentence *S* and the masked sentence *S'* as a sentence pair. Feeding the sentence pair (*S, S'*) to BERT results in predictions that «are related with both the complex word and the context» (Qiang et al., 2021). Finally, for substitute ranking they used multiple features.

4.1.1.11 Sentence Simplification with Deep Reinforcement Learning

Recent approaches often utilize insights from machine translation to learn how to simplify complex sentences by using monolingual corpora of both complex and simple sentences. Zhang & Lapata (2017) proposed a model that is based on a recurrent neural network with an encoder-decoder architecture. The encoder converts the input sentence into continuous representations which are then used by the decoder, another LSTM network, to generate the simplified output sentence. The model was trained using reinforcement learning to optimize a reward function that values simplicity, relevance, and fluency in the output. The model also includes a lexical simplification component to improve its performance. This component is learned explicitly and integrated with the reinforcement learning-based model.

Maybe it is possible to formulate a reinforcement learning task where the agent is not just rewarded by simplification, but rather by the difference between the target proficiency level and current sentence level. A model that detects the level of a sentence or individual sentence criteria (lexis, accuracy, complexity) would be needed. This combined with other metrics that ensures fluency and relevance would maybe result in a model

that is capable of learning the generation of adapted responses according to a language level.

4.1.1.12 Style Transfer

Another approach to transforming a sentence or phrase into more or less complex language is style transfer. The goal of style transfer is to automatically control the style attributes of a text while preserving the content. This goal coincides with the goal of TS, but without prescribing a particular type of processing. Thus, style transfer may also be used to modify a sentence or phrase into more complex language. Language is situational, and with style transfer, it may even be possible to not only match every generated utterance in the dialog system to a skill level, but perhaps introduce other styles as well. For example, English learners at a technical school have a different language style than business students. Defining style transfer requires distinguishing "style" and "content" which is typically done in one of two ways (Mou and Vechtomova 2020): The first method is based on linguistic definitions, with non-functional linguistic features classified as style (e.g., formality) and semantics classified as content. The second method is data-driven, where the invariance between two corpora (e.g., positive and negative reviews) is considered content, while the variance (e.g., sentiment) is considered style.

There has been a surge in research on text style transfer (TST) due to increasing demand for this technology and both traditional linguistic approaches and more recent neural network-based approaches have been developed (Jin et al., 2022). Traditional approaches often rely on term replacement and templates, but these often require domain-specific templates, hand-crafted phrase sets, and expression look-up tables (Jin et al., 2022). In recent years, several neural methods have been proposed for TST, often using standard Seq2Seq models when parallel data is available (Rao and Tetreault 2018). But obtaining parallel data for text style transfer can be challenging, and in some cases, it may not be possible (Jin et al., 2022).

For non-parallel data disentanglement (i.e., Variational Auto-Encoder and Generative Adversarial Networks) is a method that involves separating the content and style of a text (Jin et al., 2022). The goal of disentanglement is to manipulate the style of a text independently of its content, allowing the transfer of style from one text to another while preserving the content (Jin et al., 2022). Disentanglement-based models for text style transfer typically perform the following steps (Jin et al., 2022):

- Encode the text *x* with attribute *a* into a latent representation z (i.e., $x \rightarrow z$)
- Manipulate latent representation *z* to remove the source attribute (i.e., $z \rightarrow z'$)
- Decode into text *x'* with the target attribute *a'* (i.e., $z' \rightarrow x'$)

Overall, disentanglement aims to provide a flexible and controllable way to perform style transfer by allowing the separation and manipulation of content and style in a text.

For a successful style transfer, the output text must not only exhibit the correct target style, but it must also preserve the original semantics and maintain natural language fluency (Jin et al., 2022). Thus, BLEU scores, which are commonly used in text style transfer, can be problematic because they mainly evaluate content and may not accurately reflect transferred style strength or semantic preservation (Jin et al., 2022). Therefore, the commonly used evaluation criteria for style transfer include transferred style strength, semantic preservation and fluency (Jin et al., 2022).

4.1.2 Text Complexification (TC)

Assuming that the response of the dialog system is at a low complexity level and the learner's CEFR level is at a higher level, the text should not be simplified but, on the contrary, complicated according to the learner's skill level. The inverse of TS is to increase the verbosity of a phrase or sentence, and thus to increase the text complexity or to elaborate the text. TC aims to produce a more complex version of a source sentence adding clauses and phrases and increasing lexical complexity while preserving grammar and semantics. Overall, while there are some tools[19] [20] that promise TC, it is a less studied area compared to TS.

Nevertheless, it is assumed that many of the described approaches to TS can also be used for TC. Looking at the Seq2Seq approaches (see 4.1.1.2 and following) described, it might be possible to mirror the parallel corpus and thus learn how to generate a more complex sentence from a given simpler input sentence. Perhaps it is even possible to not only mirror a parallel corpus, but to merge the original and the mirrored corpus. This would maybe result in a model that is able to both increase and decrease the language level of an input sentence. However, it is strongly expected that this will only work when using a controllable approach (e.g., see 4.1.1.7) with source-side artificial tokens that indicate the target sentence level. Otherwise, it would not be clear what adaptation is required. Regarding a lexical simplification pipeline (see 4.1.1.8) the CWI task could be replaced by an easy word identification approach (using rule-based, word-frequency, etc.) and during the substitute selection step, instead of selecting the easiest candidate, the most complex (by word-frequency) sentence could be picked. While these are all just assumption, no actual paper was found, that supports any of these thoughts.

[19] https://www.csgenerator.com/

[20] https://teachandgo.com/complex-sentence-generators/

4.1.2.1 *Task Definition as Pipeline enabling the Use of other NLP Technologies*

Looking at the units for measuring spoken language, perhaps the task of TC (or adaption in general) could be solved using a pipeline architecture in which one component is responsible for lexical simplification and one for syntactic complexity, while another component ensures the accuracy of the response. Having this split, other related NLP tasks might be used.

For example, sentence fusion, also known as text fusion or text merging, is a task that involves combining multiple sentences or phrases into a single and coherent text (Geva et al., 2019). In contrast to text summarization, for sentence fusion it is important to preserve the meaning and structure of the original sentences. Merging two or more sentences would often result in connected main clauses or subordinate clauses, making a sentence more syntactical complex. Other NLP tasks that could become interesting are for instance word substitution (lexical complexity), phrase substitution (syntactical complexity), or sentence expansion (syntactical complexity).

4.1.2.2 *Controllable Text Generation using Transformer-based Pre-trained Models*

Controllable text generation is the process of generating text based on specific conditions or elements (Zhang et al., 2022). According to Zhang et al. (2022) and as shown in Figure 14 this is achieved by utilizing a system that consists of three components: an input, which includes a controlled condition (such as a positive sentiment) and a source text (I); a generative model (such as a pre-trained Language model) (P); and an output, which is the generated text that satisfies the input condition (O). For example, if the goal is to generate a sentence with a positive sentiment, the condition of "positive sentiment" and a prompt such as "I am always" would be input into a generative pre-trained Language model (PLM), resulting in an output sentence with the desired sentiment, such as "I am always happy to see you"

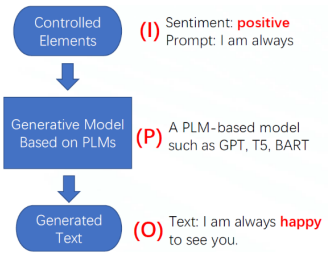

Figure 14: The IPO of controlled text generation (Zhang et al., 2022)

Maybe a similar approach can be applied using the target language level as controlled element and the source sentence as prompt. The goal of controllable text creation would then be to create a sentence that meets this level and remains as consistent as possible

in terms of content. Even though this approach is described as part of TC, it will probably be possible to do the same for general text adaptation.

Zhang et al. (2022) divide pre-trained Language models into three categories: Auto-Encoding (AE) Models, Auto-Regressive (AR) Models and Seq2Seq Models:

- AE models are created by altering the input text in some manner, such as obscuring certain words of a sentence and then attempting to rebuild the initial text (Zhang et al., 2022). Notable examples of this type include BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019).
- The primary function of AR models is to anticipate the next word by analyzing the text that has already been read (Zhang et al., 2022). The GPT (Radford et al., 2019; Brown et al., 2020) family is an example of a model that falls under this category.
- Seq2Seq models utilize both the encoder and decoder from the transformer for increased model adaptability (Zhang et al., 2022). Currently, the most well-known models of this category are T5 (Raffel et al., 2020) and mBART (Liu et al., 2020). In general, almost all pre-trained tasks used in AE and AR models can be adapted to Seq2Seq models (Zhang et al., 2022).

For controlled text generation using PLM, most techniques rely on using generative models such as AR and Seq2Seq models as a foundation and then directing them to produce the desired text (Zhang et al., 2022). But because PLM are «essentially still black-box models like other deep neural networks, making it difficult to interpret the generated text and lacking controllability», there has been a lot of interest in researching ways to effectively utilize PLMs in text generation while also maintaining control over the generative process (Zhang et al., 2022). Figure 15 summarizes approaches suggested by Zhang et al. (2022) for controlled text generation using PLMs. The methods divided into three categories would have to be further investigated. But if a suitable control mechanism is found, perhaps PLMs can be used for TC (or text adaption).

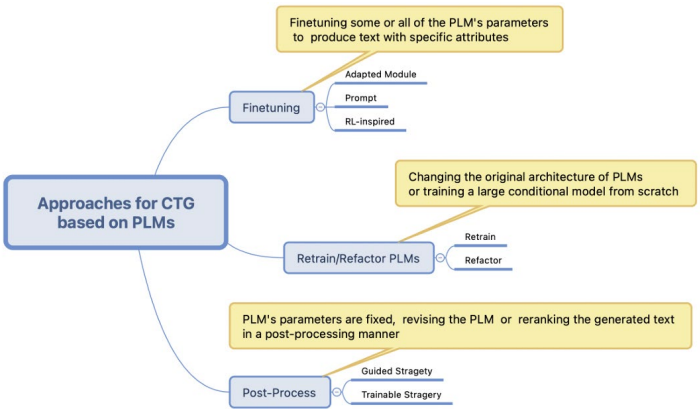

Figure 15: Overview of controlled text generation based on PLM (Zhang et al., 2022)

4.1.3 Automatic Evaluation

This chapter shows two metrics, BLEU and SARI, that can be used for automatic evaluation. Both metrics measures the similarity between a generated text and a simplification reference. The extent to which the same metrics can also be used for text complexification or text adaptation in general needs to be investigated. However, since a generated text is compared with a reference, it is assumed that the type of adaptation should not matter. Automatic evaluation can be used to compare and tune different models and approaches in this work or follow-up work. Ideally, human evaluation should also be taken into account to determine the quality of adaptions.

4.1.3.1 BLEU (Bilingual Evaluation Understudy)

BLEU (Papineni et al., 2002) score is a standard metric in machine translation and computes the n-gram overlap between the machine-translated text and a set of references. The BLEU score ranges from 0 to 1, with 1 indicating a perfect match between the machine-generated text and the reference text. A major problem with the BLEU metric in terms of TS is that copies of the original text are highly rewarded even though no simplification was done at all (Spring et al., 2021). Here is the general process to calculate the BLEU score (Papineni et al., 2002):

1. Break down the machine-generated text and reference text into n-grams, where *N* can be 1 (unigrams), 2 (bigrams), 3 (trigrams), or 4 (4-grams).
2. Count the number of matches between the machine-generated N-grams and the reference text n-grams for each n.
3. Calculate the precision (*p*) for each n-gram by dividing the number of matches by the number of machine-generated n-grams.
4. Calculate the geometric mean of the precision scores for all n-grams.
5. Apply a brevity penalty if the machine-generated text is shorter than the reference text with *r* length of reference- and *c* length of candidate translation

   $$\mathrm{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}$$

6. The final BLEU score is the geometric mean of the n-gram precisions multiplied by the brevity penalty.

   $$\mathrm{BLEU} = \mathrm{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

4.1.3.2 SARI (Self-referential Automatic Text Revision)

SARI (Xu et al., 2016) in contrast was introduced especially for TS and to punish excessive copying. It evaluates the output of a monolingual text-to-text generation system and can therefore compare system output to both references and the input sentence (Xu et al.,

2016). In contrast, machine translation metrics such as BLUE do not perform comparisons with (foreign) input sentences. Figure 16 show the regions that are differently treated by the SARI metric.

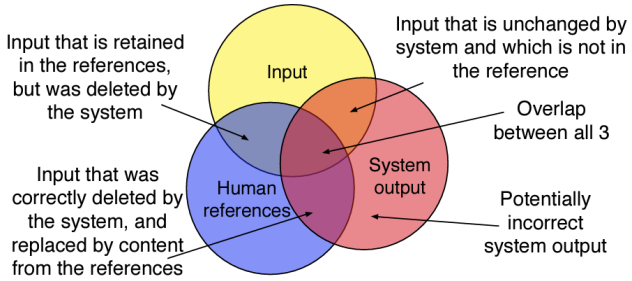

Figure 16: The regions that are treated differently in the SARI metric (Xu et al., 2016).

The SARI metric takes into account the input and rewards tokens in the generated text that are not present in the input but occur in one of the reference texts (add), as well as tokens in the input that are correctly preserved (keep) or removed (delete) in the generated text (Spring et al., 2021). The SARI metric «explicitly measures the goodness of words that are added, deleted and kept by the system» (Xu et al., 2016) and thus rewards simplification that are dissimilar from the input. SARI was shown to exhibit «reasonable correlation with human evaluation on the TS task» (Xu et al., 2016).

4.1.4 Datasets

The chapter introduces datasets that are used in text adaption tasks. Most often they are used for TS. This chapter is based on blog article by S. Ruderer from 6.12.2021[21]. These datasets could be used to train and evaluate the models in this work or follow-up work. In their survey about automatic TS, Al-Thanyyan and Azmi (2021) describe even more datasets including other languages.

4.1.4.1 Simple English Wikipedia

Simple English Wikipedia is a handwritten online encyclopaedia designed for English learners and contains simpler language than the main English Wikipedia. The popularity of using Wikipedia for simplification research is due to the availability of sentence alignments between main and simple English Wikipedia articles.

Zhu et al. (2010) created a parallel corpus **PWKP** with over 108'000 sentence pairs between simple and standard Wikipedia, including both one-to-one and one-to-many alignments (the latter of which represent instances of sentence splitting). Zhang and

[21] https://github.com/sebastianruder/NLP-progress/blob/master/english/simplification.md

Lapata (2017) released a standardized split of this dataset called **WikiSmall**[22], which includes 89'042 instances for training, 205 for development, and the original 100 instances for testing. The **WikiLarge**[22] dataset, created using additional alignments from other Wikipedia-based datasets, contains 296'000 instances and includes not just one-to-one alignments (Zhang & Lapata, 2017).

Based on the PWKP dataset, Xu et al. (2016) released a dataset called **Turk Corpus**[23] which consists of one-to-one alignments focused on paraphrasing transformations and multiple (8) simplification references per original sentence. The dataset includes 2'350 sentences, with 2'000 instances for tuning and 350 for testing. Alva-Manchego et al. (2020) created a dataset called **ASSET**[24] that is aligned with TurkCorpus and includes additionally manual references with multiple simplification operations like lexical paraphrasing, compression, and sentence splitting. Like TurkCorpus, the dataset contains both one-to-one and one-to-many alignments and has the same number of instances.

4.1.4.2 Newsela

The Newsela corpus, introduced by Xu et al. (2015), consists of 1'130 news articles that have been manually simplified by professional editors into four versions, with version 0 being the original and versions 1 to 4 being progressively simpler. These simplifications were created with the intention of targeting children of different grade levels. A manual evaluation of a subset of the data by Xu et al. (2015) found that the Newsela corpus has a greater presence and distribution of simplification transformations compared to the PWKP (see Chapter 4.1.4.1). Researchers can request the dataset[25], but they are not permitted to share splits of it, which hinders reproducibility and comparison among models.

Zhang and Lapata (2017) used sentence alignments between all versions of each article in the Newsela corpus but removed some pairs that were too similar. They ended up with a dataset consisting of 94'208 instances for training, 1'129 instances for development, and 1'076 instances for testing. The test set specifically contained only one-to-one alignments, with a single simplification reference per original sentence.

4.1.4.3 CEFR-ASAG Corpus

This CEFR-ASAG[26] corpus by Tack et al. (2017) includes various short writings composed by individuals who are not fluent in English. Each contributor was prompted to give a brief

[22] https://github.com/XingxingZhang/dress

[23] https://github.com/cocoxu/simplification/

[24] https://github.com/facebookresearch/asset

[25] https://newsela.com/data/

[26] https://github.com/anaistack/cefr-asag-corpus

response to an open-ended question that focused on the level of proficiency they were categorized into. Each question is labeled with a specific CEFR level. Additionally, 299 of the answers gathered were also labeled by a group of three CEFR-certified evaluators using the CEFR standard. Their labels, along with a label obtained through a majority vote, have been added to each of these texts.

4.1.4.4 Open Cambridge Learner Corpus (Uncoded)

The Open Cambridge Learner Corpus (Uncoded)[27] is a compilation of English text put together by Cambridge University Press and Cambridge English Language Assessment. The term "uncoded" means that it does not include any error tagging. It is a representative subset of the Cambridge Learner Corpus, which showcases the writing style of English language learners during exams. The corpus includes 2.9 million words from over 10,000 student responses from Cambridge English Language Assessment exams. It also includes data from a diverse range of first languages spoken by students from more than 60 countries.

4.2 Response Adaption Requirements

As shown in the previous analysis, there are different approaches to adapt chatbot responses, but all the approaches share the same goal of controlling the style attributes of a text while preserving the content. The extent of adapting these style attributes depends on the learner's skill level in this work. This chapter tries to formalize the task of response adaptation to a learner's language level by defining requirements.

4.2.1 Target Skill Levels

In the analysis (see 2.2) CEFR and Lehrplan21 were introduced. For the sake of simplicity, however, only the CEFR levels will be considered. Following the definition by the EDK of the levels of competence to be achieved, levels A1 to B2 appear to be particularly relevant. Based on the CALF-framework (see 2.3) the subsequent chapters try to formulize CEFR level-dependent complexity adaption. The determination of exact thresholds per criteria is not part of this work.

4.2.2 Type and Degree of Adaptation

The extent to which the system produce adapted language depends on the learners' CEFR-level. Since the text adaption should not only simplify complex responses but also make too simple responses more complex, this task should consider both TS and TC. By

[27] https://www.sketchengine.eu/cambridge-learner-corpus/#toggle-id-1

doing so, basic user should see complexity-reduced texts whereas proficient users should deal with complexity-increased responses. Further, the degree of adaptation to the response depends on the difference between the level of the answer and the skill level of the learner. The greater the difference, the more significant the adaptation should be. For example, if the chatbot's response is at level B2 or higher, but the learner's level is only A2, the adaptation must be more significant than if the learner has level B1. In contrast, minor skill discrepancies lead to little or no adaptation.

4.2.3 Complexity Adaption

Table 3 shows the complexity-related measurable criteria for increasing and reducing the complexity of chatbot responses. From an interactional point of view, complexity can be defined as the mean length of return. As the language proficiency matures, learner's mean length of utterances will increase. Likewise, the mean length of response (turn) should increase, as the user is able to proceed more information at once.

The grammatical complexity can be defined by the types of clauses used. Complexity-reduced texts should omit subordination, while complexity-increased texts can vary between simple, coordinated, and subordinated clauses. Thus, simple responses should only consist of coordinated clauses using `and`, `because`, `so`, and `but`.

Complexity adaption also affects the levels of formality. While the sentence "*would you be so kind as to tell me the color of your favorite animal?*" is lexically and grammatically relatively simple, considering the isolated words, the sentence as a chunk should be replaced by the less polite and informal version "*Could you please tell me…*" for basic users (L. Sauer, personal communication, November 28, 2022). Nevertheless, certain chunks of formulaic language can be assigned to a lower proficiency level, even if they have higher grammatical or syntactical complexity. These are called formulaic chunks.

Formulaic chunks are fixed phrases or collocations that are commonly used in a language. These phrases are often multi-word expressions that are used frequently in a specific context or situation. For example, "I'm sorry" is a formulaic chunk that is commonly used to apologize in English. Formulaic chunks are useful for language learners because they allow them to express themselves more fluently and accurately without having to stop and think about the individual words in the phrase. Learning and using formulaic chunks can also help learners to sound more natural and fluent when speaking or writing in the target language. Thus, simplified text should contain highly frequent formulaic chunks.

Aspect	Sub-category	Complexity-reduced	Complexity-increased
Complexity	Interactional	Short Mean Length of Turn	Longer Mean Length of Turn
	Grammatical	• No subordination. Just simple and coordinated clauses. • Coordinate clauses in spoken language: *and*, *because*, *so*, or *but* • Lower Level of Formality • More highly frequent Formulaic Chunks	• Use subordination. Vary between simple, coordinated and subordinated clauses. o Coordinate clauses: *and*, followed by *because*, *so*, or *but* o Conditional if-clauses, co-ordinate *or*-clauses, temporal clauses (*while*, *since*, *after*, *before*, etc.) • Higher Level of Formality

Table 3: Complexity-reduced and Complexity-increased "Syntactical Complexity"-Criteria as proposed by L. Sauer (personal communication, November 28, 2022)

4.2.4 Accuracy Adaptation

While the response from the dialog system should be error free and accurate, the adapted response may lack in accuracy due to processing. Thus, this point can be neglected, but by no means completely disregarded. The processed response must still have high accuracy, which should be ensured during adaptation. Since the computer's answers are usually grammatically correct, an additional feature in the future could be the generation of "livelier" and more realistic responses, including incomplete sentences, dysfluencies, and so on.

4.2.5 Lexical Adaption

Lexical adaption defines words and phrases used within a particular CEFR-level. Being familiar with the most frequently used words in any language will enhance comprehension while reading and listening, make communication simpler, and speed up the learning process. The CEFR does not have a specific vocabulary list or size for each level, but there are some estimates given by linguists and students[28]:

- A1 = 500 words
- A2 = 1,000 words (The top 1'000 words allows to understand about 80% of an average text)
- B1 = 2,000 words
- B2 = 4,000 words (The top 3'000 words allows to understand about 90% of an average text)
- C1 = 8,000 words (The top 5'000 words allows to understand about 95% of an average text)
- C2 = 10,000+ words (The top 10'000 words allows to understand about 99% of an average text)

The list also illustrates the reasoning behind why linguists advocate for the method of learning vocabulary in bands of frequency. While with 1,000 words already about 80% of

[28] https://vocabulary.one/en/important-facts#:~:text=Although%20The%20Common%20European%20Framework,A2%20%3D%201%2C000%20words

an average text can be understood, 10,000 words are required to understand 99% and have the vocabulary of a native speaker (see footnote[28]).

When speaking of a "word", the word family is meant, which includes the base word and its variants in inflection and derivation. For instance, active, actively, activities, and activity all belong to the same word family. Although CEFR has no specified vocabulary list per level, there are plenty of CEFR-based word lists available[29] [30]. Since the ChaLL is intended to strengthen chunk acquisition, not only isolated words must be considered, but also conversational phrases. The English Vocabulary Profiler [31] includes in addition to words also formulaic chunks. One or more of these word lists can be used to define the level dependent words.

Complexity-reduced text should be less lexical diverse and dense, while complexity increased responses should be more lexical diverse and dense. One way to formulize and measure the lexical diversity is using type-token-ratio (TTR) or the Guiraud's index. Unlike TTR, the index of Guiraud is independent of the length of the AS-unit and in this way, it leads to higher lexical richness for long texts than with simple TTR (Lindqvist et al., 2013).

Aspect	Sub-category	Complexity-reduced	Complexity-increased
Lexis	Diversity	• Low TTR / Guiraud's Index	• High TTR / Guiraud's Index
	Density	• Low proportion between lexical items and o the total number of words (Ure) o the total number of clauses (Halliday)	• High proportion between lexical items and o the total number of words (Ure) o the total number of clauses (Halliday)

Table 4: Complexity-reduced and Complexity-increased "Lexical"-Criteria as proposed by L. Sauer (personal communication, November 28, 2022)

Lexical density can be controlled using the proportion between lexical items and either the total number of words or the number of higher structural items like clauses. Lexical items need to be further specified but generally reflects the real content and includes nouns, verbs, adjectives and adverbs. Conversely, the remaining words would represent the functional words of a sentence.

4.2.6 Fluency Adaptation

Whilst response adaption was initially only thought of on written language, it makes sense to also consider adaptions on the synthesized output of the response. As shown in Table 5, speech fluency can be influenced primarily by speaking rate and pause length. While

[29] https://www.cambridgeenglish.org/images/149681-yle-flyers-word-list.pdf

[30] https://www.oxfordlearnersdictionaries.com/wordlists/oxford3000-5000

[31] http://www.englishprofile.org/wordlists/evp

complexity-reduced synthesized responses should have a low speech-rate and greater pauses, complexity-increased synthesized responses can have higher speech-rate and shorter pauses. It should be possible to parametrize these aspects when synthesizing.

Because the responses are synthesized by machine hesitation phenomena and dysfluency can be neglected. When synthesizing, supposedly only the accuracy must be considered. Once again, it could be an additional requirement to intentionally introduce errors in the form of incomplete sentences, dysfluencies etc.

Aspect	Sub-category	Complexity-reduced	Complexity-increased
Fluency	Temporal variables	• Low speech rate • Greater pause length • Short length of run	• Higher speech rate • Shorter pause length • Longer length of run
	Hesitation phenomena / Dysfluency		

Table 5: Complexity-reduced and Complexity-increased "Fluency"-Criteria as proposed by L. Sauer (personal communication, November 28, 2022)

4.2.7 Zone of proximal Development (ZPD)

When bringing the chatbots output in accordance with the skill level of the learner, the fundamental goal is to generate a response that can be processed by the learner. According to the Lehrplan21, basic and intermediary learners can only engage in a dialogue if the interlocutors are considerate and helpful. Nevertheless, text adaptations should not blindly follow the goal of making the text as simple as possible for a given language level. Since ChaLL is a learning platform, the goal is that students learn something. As simple as this statement sounds, it is complicated to develop a system that fosters learners without overburdening them. In between, there is a small Zone of Proximal Development (ZPD).

ZPD refers to the small window between what a learner is not yet able to solve independently and what the learner can achieve with guidance and encouragement from a partner (McLeod, 2012). A task-based (or even open-domain) dialog is within ZDP when it just beyond the learners' capabilities. So, the learning effect is highest when the learner is confronted with a dialog that is just out of his ability range.

The adapted responses should just be within the learner's capabilities. Following CALF, different dimensions of complexity criteria can be applied to adjust a response that is out of the learner's ability range. Lexis for example can be used to introduce new words and make them familiar to the learner via word repetition. Further, in terms of fluency, the speed can be adjusted to help the learner become accustomed to a faster rate of speech.

4.3 Response Adaptation Results

In this chapter, the resulting approach for response adaptation is presented. Due to the limited time after the preliminary study and the two-part focus of this work, only one concrete approach to response adaptation resulted. Before this approach is described, secondary work and other attempts are briefly described in the following chapter.

4.3.1 Secondary Work and other Attempts

Before starting with the main results on response adaptation, this chapter briefly summarizes secondary work and other experiments conducted during this work. The local installation of BlenderBot 2.0 can be counted as secondary work. BlenderBot was installed with the goal of generating text that can be used to test response adaption. Because the local installation of BlenderBot for Windows initially failed, the recommended installation process could not be followed exactly. An installation guide for Windows is attached to the Appendix B.

In addition, minor efforts have been made to classify texts according to a CEFR level. This includes an investigation and tests with services "CEFR Labelling and Assessment Services"[32] and "CEFR Readability Classification Service (EN) (0.3.0)"[33] of the European Language Grid (ELG). While the service approach (Breuker, 2022) looks promising and promises services for «for assessing the overall readability of a text, difficult words in the text and alternative words (suggestions) for these difficult words», unfortunately the service does not seem to be available at the moment. Especially since the service promises word suggestions, it might be worth pursuing this. If word suggestions are included, the service would probably be able to identify learner's CEFR level and adapt chatbot response at the same time. The second service, on the other hand, can be used to classify a text into up to 12 CEFR levels. Unfortunately, this only worked to a limited extent. Most likely there is a maximum number of calls, which was exhausted after a short time. Initially the idea was to use this service to test the generated answers, but due to this limitation this was discarded.

Other solutions that promise CEFR-classification are the CEFR-studio[34] or EnglishGrammar[35]. How the classification works would need to be studied for both.

[32] https://live.european-language-grid.eu/catalogue/project/5258

[33] https://live.european-language-grid.eu/catalogue/tool-service/17368/code%20samples/

[34] https://cefr-studio.edia.nl/

[35] https://englishgrammar.pro/test.php

4.3.2 CEFR Level-dependent Lexical Adaptation using Word Lists

The following LS solution combines two approaches previously described in the analysis to achieve CEFR-level dependent personalized Lexical simplification using word lists. For the task of CWI and the final substitute selection, the personalized word list approach from 4.1.1.9 is reconsidered and for the task of substitute generation, the BERT approach described in 4.1.1.10 serves as baseline. Further, this approach assumes that there is a word list per CEFR level, which defines what words should be known at a certain level. Chapter 4.2.5 shows that various institutions maintain such word lists. Although CEFR does not specify a universal list, these lists overlap to a considerable extent. This should legitimize the assumption of personalized word lists per CEFR level.

Depending on the provider, word lists are available in a structured format like JSON or only as text file (PDF). For the latter, a PDF parser script was created that generates a JSON word list per CEFR level. But this parser is not universal and for other distributors' word list other type of automatic extraction is required. For further processing, it is important that all word lists share the same format (JSON). Considering all CEFR-levels, the parsing resulted in six word lists. These lists are progressive, meaning that an A2 level student should be able to master A1 level words as well. Obviously, this means that the higher the level of the learner, the larger the vocabulary that should be known.

4.3.2.1 Personal Complex Word Identification

Knowing the words respectively the word families that correspond to a language level allows a statement about whether a word is too complicated for a learner with a certain CEFR level or not. Combining all the words from the word list with the same or a lower CEFR level results in a list of words that the learner should know. In contrast, words from word lists of higher CEFR levels correspond as too complicated words. For example, if the learner's skill level was previously identified as level B1, all words from A1, A2, and B1 should be familiar to the learner (known words). Words from B2, C1, C2, and words that are not in any of these lists are considered difficult (unknown words).

Applying this CWI approach on a given preprocessed (cleaned, lower cased, lemmatized and tokenized) chatbot response results in an assertion for each word in the AS unit whether it is known or unknown. This process is divided into two separate steps. In the first step, a word CEFR-level mask is created. The corresponding function takes an AS unit as input and returns a list of indices with the same length as the number of words in the AS unit. The parametrizable method can provide either an absolute or relative list of indexes as a result. While absolute indexes simply correspond to the index of the level (A1 = 0, A2= 1, B1 = 2, etc.), a relative index requires a reference level. By doing so, the indices

are calculated relatively to this reference level. Given a word level of A2 and a reference level of B1, the relative index would be 1 for example.

A mask with the relative values already contains the information about whether a word is classified as too complicated or not, simply by checking for values greater than 0. However, to have more possibilities when defining complex words, there is a second step, respectively a second method. This method takes both the token list and the level mask (`*token_level_mask*`) generated in the first step. Additionally, the following options can be used for additional configuration:

- exclude_words: Allows to exclude words by providing a list of word lemmas to be excluded. Assuming a task-based conversation about furniture for example, excluding certain words would allow confronting the learner with specific vocabulary (Furniture, Armchair, Cupboard etc.) that are within the ZPD.
- exclude_pos_tags: Allows words to be excluded by specifying a list of POS tags to exclude. This option can be used to focus on the relevant parts of the chatbot's responses. Regarding lexical simplification, the focus is particularly on nouns, adjectives and verbs. Other parts of speech can be excluded using this option.
- target_level: Allows to explicitly specify the minimum level for a token to be considered complex. Assuming the *token_level_mask* is relative, per default all positive values greater than 0 are considered as complex. But when using absolute word level mask, this option can be used to define the reference level. Additionally, this option can also be applied for ZPD, but in a rather imprecise way. By doing so, all words below or equals to the specified level (current-level + target-level) can be excluded from being identified as complicated. This works with both positive and negative targets.

Finally, this method returns a token level mask that provides for each token in a sentence the information about being complicated (1) or not (0).

The higher the difference between the overall sentence level and the learner's language level, the more often it happens that several complicated words follow each other. In the case of consecutive complicated words, sometimes it seems better to mask these words together. Instead of generating substitution candidates for each complex consecutive word, substitution candidates are generated only for the entire sequence of consecutive complex words. This has the effect that the token list and the token level mask become shorter.

Below is an example sentence generated using ChatGTP and the command: "can you formulate a sentence that correspond to CEFR level B2 about furniture". For the relative

token level mask, level A2 and A1 were set as reference. In case of A2, this means for example that all negative values correspond to levels below A2, all levels equal to 0 correspond to level A1, and all positive values are above level A2. The value 4 indicates a word that does not appear in any of the word lists (number of possible higher levels + 1).

Original Sentence (≈B2)	*"The contemporary sofa set with adjustable headrests and reclining options provides a high level of comfort and functionality, compared to the traditional, rigid couch."*
Token-Level-Mask (relative, reference A2)	[-1, 2, 4, 1, -1, 4, 4, -1, 4, 0, 0, -1, -1, 0, -1, 2, -1, 4, -1, -1, -1, 0, 4, 4]
Complex-Word-Mask	[0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1]
Consecutive Merge	[0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1]
Token-Level-Mask (relative, reference A1)	[0, 3, 5, 2, 0, 5, 5, 0, 5, 1, 1, 0, 0, 1, 0, 3, 0, 5, 0, 0, 0, 1, 5, 5]
Complex-Word-Mask	[0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1]
Consecutive Merge	[0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1]

4.3.2.2 Substitute Generation

The CWI step is followed by the substitution generation. For this purpose, the BERT-based lexical substitution approach proposed by Qiang et al. (2021) is reused. This means that BERT's masked language modeling is applied, to predict missing tokens in a sequence. Based on the previously defined complex word mask, (consecutive) complex words are iteratively masked with `[MASK]`. As suggested by Qiang et al. (2021) and to generate candidates that are not to semantically different from the complex word, the original sentence and the masked sentence are concatenated. Using the sentence pair (S, S') helps generate candidates that are both related with the context and the masked complex word. The number of substitute candidates can be parametrized.

Using the default number of 10 substitute candidates, the following example shows the candidates generated for the previous text example and the reference level A2. For this example, BERT's large, uncased model (`bert-large-uncased`) was used. This model is uncased, which means that is does not make a distinction between `English` and `english`.

A2	[('**contemporary**', ['contemporary', 'modern', 'traditional', 'conventional', 'flexible', 'standard', 'original', 'realistic', 'current', 'modernist']), ('**sofa**', ['sofa', 'couch', 'chair', 'lounge', 'cushion', 'seat', 'armchair', 'furniture', 'seating', 'bed']), ('**set**', ['set', 'sets', 'setting', 'style', 'bed', '##set', '##s', 'settings', 'suite', 'setup']), ('**adjustable**', ['adjustable', 'fixed', 'adjusted', 'adjustment', 'variable', 'extended', 'flexible', 'increased', 'comfortable', 'padded']), ('**headrest**', ['seat', 'head', 'back', 'chair', 'arm', 'cushion', 'neck', 'seating', 'shoulder', 'cushions']), ('**recline**', ['seat', 'back', 'seating', 'cushion', 'adjustable', 'lounge', 'rigid', 'flexibility', 'sleeper', 'swing']), ('**comfort**', ['comfort', 'comfortable', 'safety', 'security', 'support', 'luxury', 'functionality',

	'amenities', 'softness', 'stability']), ('**functionality**', ['functionality', 'function', 'flexibility', 'capability', 'functional', 'connectivity', 'performance', 'mobility', 'competence', '##tility']), ('**rigid**', ['rigid', 'flexible', 'stiff', 'modular', 'fixed', 'relaxed', 'firm', 'elastic', 'solid', 'soft']), ('**couch**', ['couch', 'sofa', 'seat', 'chair', 'cushion', 'lounge', 'furniture', 'seating', 'cushions', 'armchair'])]
A2 - consecutive merge	[('**contemporary sofa set**', ['couch', 'sofa', 'chair', 'seat', 'set', 'cushion', 'lounge', 'armchair', 'seating', 'chairs']), ('**adjustable headrest**', ['seat', 'adjustable', 'cushion', 'chair', 'arm', 'bench', 'sofa', 'back', 'pillow', 'swing']), ('**recline**', ['seat', 'back', 'seating', 'cushion', 'adjustable', 'lounge', 'rigid', 'flexibility', 'sleeper', 'swing']), ('**comfort**', ['comfort', 'comfortable', 'safety', 'security', 'support', 'luxury', 'functionality', 'amenities', 'softness', 'stability']), ('**functionality**', ['functionality', 'function', 'flexibility', 'capability', 'functional', 'connectivity', 'performance', 'mobility', 'competence', '##tility']), ('**rigid couch**', ['sofa', 'couch', 'seat', 'chair', 'cushion', 'lounge', 'style', 'armchair', 'cushions', 'seating'])]

Code related to substitute generation is based on A. Olivares medium article[36].

4.3.2.3 Substitute Selection

The final step of the proposed lexical simplification pipeline is to select the best substitute candidate. Like Lee & Yeung (2018), all proposed substitutes from step two are discarded if they are still too complicated for the learner. This is done by checking if the lemmas of the substitute candidates are included in the learner's list of known words. If they are included, they are considered as suitable candidate. For the moment, no additional substitute ranking is included. Accordingly, the candidate with the highest probability is selected from the remaining candidates. This candidate is equivalent to the first entry in the list of possible candidates.

In this two-step selection process, the algorithm tries to find the most appropriate candidate, considering only words that are known to the learner. While the number of generated substitutes might affect the outcome of the described algorithm, it is not guaranteed that a valid substitute candidate is found. Therefore, some complex words are not replaced and remain too complicated. This can also be seen in the following results of the CEFR level-dependent lexical simplification approach. For the same example sentence as before, these are the 4 adapted versions. In each case, the adapted text is displayed, as well as the adapted text with additional markings for substitutes (<s>) and still too complicated words (<c> and colored red).

Original	The contemporary sofa set with adjustable headrests and reclining options provides a high level of comfort and functionality, compared to the traditional, rigid couch
A2	the modern chair style with comfortable seat and seat option provide a high level of comfortable and functionality compare to the traditional soft couch the <s>modern</s> <s>chair</s> <s>style</s> with <s>comfortable</s> <s>seat</s> and <s>seat</s> option provide a high level of <s>comfortable</s> and <c>functionality</c> compare to the traditional <s>soft</s> couch
A2 - consecutive merge	the chair with seat and seat option provide a high level of comfortable and functionality compare to the traditional rigid couch the <s>chair</s> with <s>seat</s> and <s>seat</s> option provide a high level of <s>comfortable</s> and <c>functionality</c> compare to the traditional rigid couch

[36] https://medium.com/@armandj.olivares/how-to-use-bert-for-lexical-simplification-6edbf5a4d15e

A1	the modern chair style with adjustable head and back could present a high level of comfort and functionality compare to the modern rigid couch the <s>modern</s> <s>chair</s> <s>style</s> with <c>adjustable</c> <s>head</s> and <s>back</s> <s>could</s> <s>present</s> a high <c>level</c> of <c>comfort</c> and <c>functionality</c> compare to the <s>modern</s> <c>rigid</c> couch
A1 – consecutive merge	the chair with chair and present a high level of comfort and functionality compare to the traditional rigid couch the <s>chair</s> with <s>chair</s> and <s>present</s> a high <c>level</c> of <c>comfort</c> and <c>functionality</c> compare to the traditional rigid couch

4.3.2.4 Lexical Complexification

In addition to the described algorithm for complex word replacements, the approach was extended to increase lexical density. Like for CWI, the code also includes easy word identification, marking learner's known words as easy. For example, if the learner's skill level was previously identified as level B1, all words at CEFR levels A1 and A2 are marked as too easy (1), while all other words are not considered as words to be substituted (0). This additional feature to identify easy words, is also why the selection process is divided into two separate steps as described before. Further the parametrization helps to prevent unnecessary adaptations, that might especially occur in easy word identification. For example, many functional words are represented in the word lists of beginners. To prevent the system from masking these words, for example only nouns could be considered as valid targets using the described parameters.

The substitute candidate generation is done the same way as for complex words, but in the final selection, all candidates that are still too easy are rejected by comparing candidates and user's known word list.

4.4 Discussion and Future Work

Although there is literature on TS, the question of context-sensitive responses corresponding to a CEFR level has not been studied and posed a novel challenge. This work's contribution is a survey of related solutions, a collection of requirements for this component under consideration of L2 proficiency measuring criteria, and a concrete approach for personalized LS including implementation.

LS is typically approached by means of rule-based, statistical approaches using methods from machine translation, or a mix of both. For better control, interpretability, and to address specific target audiences, traditional LS solutions were extended with control mechanisms. With Seq2Seq models, for example, this controllability is achieved by using artificial tokens, such as the desired CEFR level of the target sentence. Whether the same hand-crafted rules from rule-based approaches or the same artificial tokens from data-driven approaches can also be used for text complexification, could not be answered with this survey. However, it is assumed that some LS systems can be reused for TC or even be extended to solve the whole adaptation task. For example, a parallel LS dataset with the original and simplified text could be reversed to learn how to generate a more complex text. By merging the original dataset and the mirrored dataset and by using artificial tokens, it might even be possible to learn text adaptations in both directions. Neither of these experiments could be performed within the scope of this work and consequently, some questions about text adaptation remains unanswered:

- How well can approaches from LS be used for text complexification? So far, only assumptions have been made on how to apply LS approaches for LC. Whether these assumptions are valid could not be verified. Likewise, the question remains whether text adaptation should be solved as a single task or split into simplification and complexification.
- How good does the adapted language need to be? Sources of errors such as hallucination, fluency errors, anaphora resolution and bad substitution have been mentioned in the survey. To what extent these are problematic has not yet been answered. With the goal for natural language interaction, the system may be less vulnerable to errors. Perhaps some type of errors could even appear human-like.
- How well should the text adaptation be controlled? It was shown that with various LS system, the form and extent of adaptation can be controlled. Whether these control mechanisms are sufficient for ChaLL needs further clarification.
- How good are the language resources? Datasets have been presented in this chapter. The parallel datasets are mostly translated from native speakers. It is not

clear whether a model that learns from these datasets meets the needs of learners.

- How to evaluate text adaptation systems? Metrics have been explored in this work, but not yet applied. They originate from the LS domain. Whether they are also suitable for TC is not clear.

In addition to the general technical questions about text adaptation, an attempt was made to formulate the chatbot response adaptation for ChaLL by means of requirements elicitation. For this purpose, the four criteria from CALF were considered. A simple (complex) answer is less (more) syntactical complex, less (more) fluent, and less (more) lexically dense, while still being accurate. For each category, pragmatic assumptions were made about what constitutes to complexity-reduced and to complexity-increased texts. Syntactic complexity and lexis are particularly crucial in response adaptation. Assuming that the chatbot's responses are synthesized accurately and correctly, the challenge is to maintain accuracy during adaptation. Fluency on the other side is considered very important and crucial. As it deals with speech rate and pause lengths, however, it seems to make more sense to consider adaptations to fluency as part of TTS.

The proposed approach for personalized lexical adaptation only focuses on the vocabulary. It assumes CEFR level-dependent word lists for complex (simple) word identification and substitute selection, while substitute candidates are generated using BERT's masked-language modelling. In this way, complex words are identified in the text, for which substitute candidates are generated and finally those are selected that are not still too complicated depending on the given user's CEFR level. This approach has been implemented but not systematically evaluated. Therefore, no conclusion about the quality of the adaptation can be made so far. To ensure accuracy, the substitute selection might have to be extended. Basically, only substitutes that are grammatically and semantically correct should be selected. Regarding BERT, no finetuning strategy has been used so far. By doing so, the substitute generation could be probably further improved.

But the personalized text adaption solution is configurable, interpretable and extensible. The approach can be configured by defining word lists, by parameterizing the CWI and by specifying the number of substitutes generated. In addition, personalization does not have to stop at CEFR-level but can perhaps be further personalized. For example, word lists could be kept per user, with words that the learner used (or heard) multiple times in previous dialogs. This would allow dynamic and user-sensitive lexical adaptation. When specifying the parameters for CWI, ZPD was considered. This includes the settings for word exclusion, POS tag exclusion and target level. For example, depending on a task (e.g., furniture & locations), specific words (table, chair, etc.) could be excluded from CWI.

This would allow learners to be intentionally faced with task-specific words. For even more sensitivity, substitute ranking could favor those substitute candidates closer to the current CEFR level in the future.

The personalized LS approach does not yet consider formulaic chunks. But as certain CEFR level word lists also include formulaic chunks per level, the chunks could first be searched and excluded from CWI. Nonetheless, it is assumed that the formulaic chunks typically compose of words that are anyway included in the corresponding CEFR level's word list. Thus, they are ignored anyway.

At the same time, it was shown that the same approach can be applied to make a text lexically more complex. Instead of identifying complex word, the word lists are used to find words that are below current learner's level. Then substitute candidates are similarly generated. Finally, those candidates are selected that are not still too easy by comparing the candidates and the known words.

Since the previous approach only deals with lexis, additional adaptation is required for syntactical adaptation. But no concrete results could be generated regarding syntactical complexity in this work. Some tests were performed using GTP3 and ChatGTP and interesting results were observed on some very specific prompts. However, these results will not be discussed further in this work. Nevertheless, such PLMs seem to offer interesting possibilities for the further course of the ChaLL project. First, they seem promising for text adaptation and ChatGTP even knows the CEFR levels: "Simplify the following sentence to CEFR grade A2". This would, secondly, probably allow to generate custom datasets. And third, PLM's may even be used to supplement automated evaluation for text adaptation: "Which of the simplified texts is the simplest?". Since the metrics described require hand-crafted references, this would be a handy addition.

5 Conclusion

Started without a clear goal, this research has yielded versatile and useful results for the ChaLL project. Many aspects of the work have emerged only in the course of the work and apart from the funding proposal, little was known about the research subject in the beginning. The work was therefore approached in a relatively unbiased manner but managed to gain valuable insight into the research subject and its subtasks and thus provide useful preliminary work for the further development of ChaLL. Nevertheless, many aspects require additional investigation and evaluation, as stated in the two separate discussions at the end of Chapters 3 and 4. The current chapter briefly summarizes the work and concludes with a critical appraisal and an outlook.

This work has started with an investigation of alternative solutions of L2 system and simultaneously answering the first research (RQ1) question. As oral interaction is often not adequately addressed inside the classroom and traditionally harder to practice outside the classroom, CALL has become a novel way to teach students to speak without time constraints and pressures of the traditional classroom. Exploring other solutions has shown the diversity and possibilities of CALL. While online games are not specifically designed for language learning and therefore do not follow the principles of the SLA, they can be useful for maintaining language skills or stimulating interest in learning a new language. Educational games, on the other hand, are designed for the use in classrooms, but still, they lack in effectiveness and teaching pragmatic or cultural knowledge. Finally, L2 apps are designed specifically for language learning and aim to teach language skills through structured lessons and exercises. Although L2 apps are effective in fostering explicit receptive reading, vocabulary and grammar in L2, the research revealed that many of the L2 solutions do not offer learners the opportunity to express themselves spontaneously and interactively, leaving potential for improvement. And this potential is what the ChaLL project aims to exploit. In addition, with Quazel a solution was found that already includes a large proportion of the idea of ChaLL.

The potential collaboration with Quazel also led to a change in the focus of this work. While the original idea of the work was to design and implement the system architecture, in the end only the former was done. For this purpose, requirements were first defined. Subsequently, the architecture was developed incrementally by incorporating additional features and requirements at each iteration. The final architecture is concluded in Chapter 3.3 and visualized in Figure 11. The result is a service pipeline architecture consisting of multiple services for different tasks, including an orchestrating central service and an observer service responsible for supervision, monitoring and feedback generation. With this approach, it was shown how centralized control and decentralized logic could look

like. Further, it was demonstrated how one or multiple observer service could gather evidence during sessions, used to generate targeted feedback. However, only the actual implementation will show whether the third research (RQ3RQ3:) question has been completely answered and if the proposed architecture is suitable. Nevertheless, the outcome of this work hopefully helps when starting with the implementation of ChaLL by showing interrelationships, difficulties and most importantly possibilities.

As preliminary work for the response adaptation task, the second research question (RQ2RQ2:) is relevant. This task requires, on the one hand, the definition of levels and, on the other hand, the definition of criteria that can be used to measure language according to these levels. For the definition of levels, the two references CEFR and Lehrplan21 were examined. Even though Lehrplan21 further subdivides the skill levels of CEFR, the less granular CEFR levels are sufficient in this work. While both the CEFR and Lehrplan21 formulates competencies as “Can Do” this definition is insufficient and inappropriate for the response adaptation task.

Therefore, criteria to measure L2 proficiency were defined in collaboration with L. Sauer. These criteria, divided into the categories of (syntactical) complexity, accuracy, lexis and fluency, can be used to both identify the complexity of a chatbot response and to frame the task of response adaptation. While the syntactic complexity and lexis criteria can directly be applied to the chatbot response, accuracy must be maintained throughout the adaptation process, and fluency (speech rate, pause length, etc.) must be considered during text synthesize. Based on these four criteria, pragmatic assumptions were made to describe what constitutes to complexity-reduced and to complexity-increased texts. In this way, the second research question is at least partially answered. But further research and the specification of exact thresholds per level and criterion are required for an exact differentiation between the levels.

The contribution of this work regarding the fourth research question (RQ4RQ4:) is a survey of related solutions, a collection of requirements, and a concrete approach for lexical adaptation including implementation. As summarized in the discussion in Section 4.4, it is assumed that solutions from LS can also be used for text adaptation in general, but no literature was found on this. While the implemented personalized LS approach, looks promising in terms of configurability, interpretability and extensibility, it neither solves the response adaptation task nor fully answers the fourth research question.

Chapters 3 and 4 conclude with a discussion and an outlook, thus answering the final research question (RQ5). Whether the proposed architecture will be viable will only become clear when the actual implementation begins. Similarly, the task of context-sensi-

tive response adaptations leaves some unanswered questions that need further investigation. Nevertheless, the results of this work provide a foundation for the initiation of the ChaLL project, facilitating its implementation. Further, a potential follow-up semester work could delve deeper into the area of context-sensitive response adaptation to provide more comprehensive understanding of the task. Finally, this work should be understood as a collection of ideas rather than an exact blueprint for ChaLL.

Bibliography

Alderson, J. C. (2007). The CEFR and the need for more research. *The Modern Language Journal*, *91*(4), 659–663.

Al-Thanyyan, S. S., & Azmi, A. M. (2021). Automated Text Simplification: A Survey. *ACM Comput. Surv.*, *54*(2). https://doi.org/10.1145/3442695

Alva-Manchego, F., Bingel, J., Paetzold, G., Scarton, C., & Specia, L. (2017). Learning how to simplify from explicit labeling of complex-simplified text pairs. *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 295–305.

Arteaga, D. L. (2000). Articulatory phonetics in the first-year Spanish classroom. *The modern language journal*, *84*(3), 339–354.

Baevski, A., Zhou, Y., Mohamed, A., & Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, *33*, 12449–12460.

Bajorek, J. (2017). L2 Pronunciation Tools: The Unrealized Potential of Prominent Computer-assisted Language Learning Software. *Issues and Trends in Educational Technology*, *5*, 60–87. https://doi.org/10.2458/azu_itet_v5i1_bajorek

Beatty, K. (2013). *Teaching & Researching: Computer-Assisted Language Learning* (0 Aufl.). Routledge. https://doi.org/10.4324/9781315833774

Bingel, J., & Søgaard, A. (2016). Text simplification as tree labeling. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 337–343.

Bott, S., Saggion, H., & Mille, S. (2012). Text simplification tools for spanish. *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, 1665–1671.

Breuker, M. (2022). CEFR Labelling and Assessment Services. In *European Language Grid: A Language Technology Platform for Multilingual Europe* (S. 277–282). Springer International Publishing Cham.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., & others. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, *33*, 1877–1901.

Buechel, L. L., & Lichtenauer, K. (2019). Grading and Gathering Evidence in Swiss Elementary and Lower Secondary School English Language Classrooms. In *The Routledge Handbook of Language Education Curriculum Design* (S. 222–237). Routledge.

Chandrasekar, R., Doran, C., & Bangalore, S. (1996). Motivations and methods for text simplification. *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*.

Chen, X., Zou, D., Xie, H. R., & Su, F. (2021). *Twenty-five years of computer-assisted language learning: A topic modeling analysis*.

Chik, A. (2013). Naturalistic CALL and Digital Gaming. *TESOL Quarterly*, *47*(4), 834–839. https://doi.org/10.1002/tesq.133

Conneau, A., Baevski, A., Collobert, R., Mohamed, A., & Auli, M. (2020). Unsupervised cross-lingual representation learning for speech recognition. *arXiv preprint arXiv:2006.13979*.

Council of Europe. (2001). *Common European Framework of Reference for Languages: Learning, Teaching, Assessment*. https://rm.coe.int/common-european-framework-of-reference-for-languages-learning-teaching/16809ea0d4

Daller, H., Van Hout, R., & Treffers-Daller, J. (2003). Lexical richness in the spontaneous speech of bilinguals. *Applied linguistics*, *24*(2), 197–222.

De Belder, J., & Moens, M.-F. (2010). Text simplification for children. *Prroceedings of the SIGIR workshop on accessible search systems*, 19–26.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dinan, E., Roller, S., Shuster, K., Fan, A., Auli, M., & Weston, J. (2018). Wizard of wikipedia: Knowledge-powered conversational agents. *arXiv preprint arXiv:1811.01241*.

Du, X. (2009). The affective filter in second language teaching. *Asian social science*, *5*(8), 162–165.

Ellis, R., & Barkhuizen, G. P. (2005). *Analysing learner language*. Oxford University Press.

Foster, P. (2000). Measuring spoken language: A unit for all reasons. *Applied Linguistics*, *21*(3), 354–375. https://doi.org/10.1093/applin/21.3.354

Geva, M., Malmi, E., Szpektor, I., & Berant, J. (2019). DiscoFuse: A large-scale dataset for discourse-based sentence fusion. *arXiv preprint arXiv:1902.10526*.

Glasmachers, T. (2017). Limits of end-to-end learning. *Asian conference on machine learning*, 17–32.

Godwin-Jones, R. (2014). Games in language learning: Opportunities and challenges. *Language Learning & Technology*, *18*(2), 9–19.

Goh, C. C. M., & Burns, A. (2012). *Teaching speaking: A holistic approach*. Cambridge University Press.

Golonka, E. M., Bowles, A. R., Frank, V. M., Richardson, D. L., & Freynik, S. (2014). Technologies for foreign language learning: A review of technology types and their effectiveness. *Computer assisted language learning*, *27*(1), 70–105.

Grimm, N., Meyer, M., & Volkmann, L. (2015). *Teaching English*. Narr Francke Attempto Verlag GmbH + Co. KG. https://elibrary.narr.digital/book/99.125005/9783823378310

Guiraud, P. (1954). *Les caractéristiques statistiques du vocabulaire*. Paris, France: Presses Universitaires de France.

Halliday, M. A. K. (1989). *Spoken and written language*. Oxford University Press, USA.

Hedge, T. (2011). *Teaching and learning in the language classroom* (1 publ., [Nachdr.]). Oxford Univ. Press.

Huang, H., Xu, H., Wang, X., & Silamu, W. (2015). Maximum F1-score discriminative training criterion for automatic mispronunciation detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *23*(4), 787–797.

Huang, M., Zhu, X., & Gao, J. (2020). Challenges in building intelligent open-domain dialog systems. *ACM Transactions on Information Systems (TOIS)*, *38*(3), 1–32.

Jabbari, N., & Eslami, Z. R. (2019). Second language learning in the context of massively multiplayer online games: A scoping review. *ReCALL*, *31*(1), 92–113. https://doi.org/10.1017/S0958344018000058

Jin, D., Jin, Z., Hu, Z., Vechtomova, O., & Mihalcea, R. (2022). Deep learning for text style transfer: A survey. *Computational Linguistics*, *48*(1), 155–205.

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., & others. (2017). Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, *5*, 339–351.

Klein, D., & Manning, C. D. (2003). Accurate unlexicalized parsing. *Proceedings of the 41st annual meeting of the association for computational linguistics*, 423–430.

Komeili, M., Shuster, K., & Weston, J. (2022). Internet-Augmented Dialogue Generation. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 8460–8478. https://doi.org/10.18653/v1/2022.acl-long.579

Kormos, J. (2011). Speech production and the Cognition Hypothesis. *Second language task complexity: Researching the Cognition Hypothesis of language learning and performance*, *2*, 39–60.

Lee, J. S., & Yeung, C. Y. (2018). Personalizing lexical simplification. *Proceedings of the 27th International Conference on Computational Linguistics*, 224–232.

Lindqvist, C., Gudmundson, A., & Bardel, C. (2013). *A new approach to measuring lexical sophistication in L2 oral production* (S. 109–126). EUROSLA - the European Second Language Association. http://urn.kb.se/resolve?urn=urn:nbn:se:su:diva-98174

Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., Lewis, M., & Zettlemoyer, L. (2020). Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, *8*, 726–742.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach* (arXiv:1907.11692). arXiv. http://arxiv.org/abs/1907.11692

Loewen, S., Isbell, D. R., & Sporn, Z. (2020). The effectiveness of app-based language instruction for developing receptive linguistic knowledge and oral communicative ability. *Foreign Language Annals*, *53*(2), 209–233.

Lord, G. (2016). Rosetta Stone for language learning: An exploratory study. *IALLT Journal of Language Learning Technologies*, *46*(1), 1–35.

Maddela, M., Alva-Manchego, F., & Xu, W. (2020). Controllable text simplification with explicit paraphrasing. *arXiv preprint arXiv:2010.11004*.

Martin, L., Sagot, B., de la Clergerie, E., & Bordes, A. (2019). Controllable sentence simplification. *arXiv preprint arXiv:1910.02677*.

Morin, R. (2007). A neglected aspect of the standards: Preparing foreign language Spanish teachers to teach pronunciation. *Foreign Language Annals*, *40*(2), 342–360.

Ng, H. T., Tetreault, J., Wu, S. M., Wu, Y., & Hadiwinoto, C. (Hrsg.). (2013). *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*. Association for Computational Linguistics. https://aclanthology.org/W13-3600

Ng, H. T., Wu, S. M., Briscoe, T., Hadiwinoto, C., Susanto, R. H., & Bryant, C. (2014). The CoNLL-2014 shared task on grammatical error correction. *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, 1–14.

Nishihara, D., Kajiwara, T., & Arase, Y. (2019). Controllable text simplification with lexical constraint loss. *Proceedings of the 57th annual meeting of the association for computational linguistics: Student research workshop*, 260–266.

Nisioi, S., Štajner, S., Ponzetto, S. P., & Dinu, L. P. (2017). Exploring neural text simplification models. *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 2: Short papers)*, 85–91.

Oroujlou, N., & Vahedi, M. (2011). Motivation, attitude, and language learning. *Procedia - Social and Behavioral Sciences*, *29*, 994–1000. https://doi.org/10.1016/j.sbspro.2011.11.333

Paetzold, G., Alva-Manchego, F., & Specia, L. (2017). Massalign: Alignment and annotation of comparable documents. *Proceedings of the IJCNLP 2017, System Demonstrations*, 1–4.

Paetzold, G. H., & Specia, L. (2017). A Survey on Lexical Simplification. *Journal of Artificial Intelligence Research*, *60*, 549–593. https://doi.org/10.1613/jair.5526

Paetzold, G., & Specia, L. (2016). Semeval 2016 task 11: Complex word identification. *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, 560–569.

Paetzold, G., & Specia, L. (2017). Lexical simplification with neural ranking. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 34–40.

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). Bleu: A method for automatic evaluation of machine translation. *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 311–318.

Pfenninger, S. E., & Lendl, J. (2017). Transitional woes: On the impact of L2 input continuity from primary to secondary school. *Studies in Second Language Learning and Teaching*, *7*(3), 443–469. https://doi.org/10.14746/ssllt.2017.7.3.5

Qiang, J., Li, Y., Zhu, Y., Yuan, Y., Shi, Y., & Wu, X. (2021). LSBert: Lexical Simplification Based on BERT. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *29*, 3064–3076. https://doi.org/10.1109/TASLP.2021.3111589

Qiang, J., Li, Y., Zhu, Y., Yuan, Y., & Wu, X. (2020). Lexical simplification with pretrained encoders. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(05), 8649–8656.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., & others. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, *1*(8), 9.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P. J., & others. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, *21*(140), 1–67.

Rothe, S., Narayan, S., & Severyn, A. (2020). Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics*, *8*, 264–280.

Scarton, C., & Specia, L. (2018). Learning simplifications for specific target audiences. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 712–718.

Shibano, T., Zhang, X., Li, M. T., Cho, H., Sullivan, P., & Abdul-Mageed, M. (2021). Speech Technology for Everyone: Automatic Speech Recognition for Non-Native English with Transfer Learning. *arXiv preprint arXiv:2110.00678*.

Siddharthan, A. (2014). A survey of research on text simplification. *ITL-International Journal of Applied Linguistics*, *165*(2), 259–298.

Spring, N., Rios, A., & Ebling, S. (2021). *Exploring German Multi-Level Text Simplification*.

Tack, A., François, T., Roekhaut, S., & Fairon, C. (2017). Human and Automated CEFR-based Grading of Short Answers. *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, 169–179. https://doi.org/10.18653/v1/W17-5018

Thomson, R. I., & Derwing, T. M. (2015). The effectiveness of L2 pronunciation instruction: A narrative review. *Applied Linguistics*, *36*(3), 326–344.

Ulasik, M. A., Hürlimann, M., Germann, F., Gedik, E., Benites, F., & Cieliebak, M. (2020). CEASR: a corpus for evaluating automatic speech recognition. *Proceedings of the 12th Language Resources and Evaluation Conference*, 6477–6485.

Ure, J. (1971). Lexical density and register differentiation. *Applications of linguistics*, *23*(7), 443–452.

Wilson, S. (2008). *Components of Cognitive Apprenticeship: Scaffolding*. Retrieved from.

Xu, W., Napoles, C., Pavlick, E., Chen, Q., & Callison-Burch, C. (2016). Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, *4*, 401–415.

Young, T., Xing, F., Pandelea, V., Ni, J., & Cambria, E. (2022). Fusing task-oriented and open-domain dialogues in conversational agents. *Proceedings of the AAAI Conference on Artificial Intelligence*, *36*(10), 11622–11629.

Zhang, H., Song, H., Li, S., Zhou, M., & Song, D. (2022). A survey of controllable text generation using transformer-based pre-trained language models. *arXiv preprint arXiv:2201.05337*.

Zhang, S., Dinan, E., Urbanek, J., Szlam, A., Kiela, D., & Weston, J. (2018). Personalizing dialogue agents: I have a dog, do you have pets too? *arXiv preprint arXiv:1801.07243*.

Zhang, X., & Lapata, M. (2017). Sentence Simplification with Deep Reinforcement Learning. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 584–594. https://doi.org/10.18653/v1/D17-1062

Zhang, Z., Takanobu, R., Zhu, Q., Huang, M., & Zhu, X. (2020). Recent advances and challenges in task-oriented dialog systems. *Science China Technological Sciences*, *63*(10), 2011–2027.

Zhao, G., Sonsaat, S., Silpachai, A., Lucic, I., Chukharev-Hudilainen, E., Levis, J., & Gutierrez-Osuna, R. (2018). L2-ARCTIC: A non-native English speech corpus. *INTERSPEECH*, 2783–2787.

Zhou, W., Ge, T., Xu, K., Wei, F., & Zhou, M. (2019). BERT-based lexical substitution. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3368–3373.

Appendix A: ChaLL Funding Proposal, Solution

The following text in this appendix is from the ChaLL research proposal and has been copied verbatim. It is intended to aid understanding of the ChaLL project and to shows the state of the art that is intended to be advance with the ChaLL-project:

We identified the following research areas that encompass our project: Speech-to-Text for language learners, error detection in the learners' utterances and generation of appropriate feedback, detecting the learners' skill levels, and adapting the chatbot's responses accordingly.

Speech-to-Text for language learners

The main bottleneck for developing a system like ChaLL is the ability to "understand", i.e. transcribe, learners' speech – including the errors they make – adequately using Speech-to-Text (STT) technology. As this forms the basis for all subsequent challenges, it is crucial that this step is as accurate as possible.

In general, STT systems are able to transcribe speech ever more accurately. The main driver is the application of large-scale pre-trained speech representation models based on transformer architectures. The most well-known representation is Wav2Vec [BAEV20], which is publicly available. It achieves state of the art performance when fine-tuned on a small set of audio recordings in a target language, e.g., a Word Error Rate (WER; the percentage of words transcribed incorrectly) of 2% using only 1 hour of English speech data [CONN21].

However, these evaluations use data from controlled samples of speech uttered by native speakers of English. In our setting, we will apply STT to spontaneous speech by non-native language learners who make various types of errors, e.g. incorrect morphology and syntax, mispronunciations, altered stress and prosody, and insertion of L1 vocabulary.

These characteristics lead to decreased STT performance. For spontaneous spoken language, WERs are typically higher than for prompted or read-aloud speech and range around 25 to 30% [ULAS20]. A recent work showed that current Wav2Vec based models perform worse on L2 speech than L1 data ([SHIB21], WER of 13% vs. 2%18). However, the 13% WER was achieved on the L2-ARCTIC dataset [ZHAO18], which consists of highly advanced university students (CEFR levels B2 to C1 in English). In [HUAN15] the authors show that a model trained on L1 data performs worse on L2 data than a model directly trained on L2 data; hence, there is a need to use targeted data, i.e. spontaneous speech by young L2 speakers.

For the part of the STT feasibility study, we want to evaluate how well existing STT systems work for young Swiss learners of English, and how much additional in-domain data is needed to achieve a sufficient recognition performance (WER) for subsequent tasks and interactions.

For this, we will leverage a pre-trained model and the targeted data which we will collect. Specifically, we will fine-tune the XLSR-1B model on different amounts of data of Swiss primary school children speaking English to project the learning curve (e.g. 1h, 5h, 10h, 20h, 50h, 100h). The performance could be further augmented by measures such as the following:

- including further native and non-native English STT corpora in the fine-tuning
- optimising model architectures and hyperparameters
- creating synthetic fine-tuning data by reading English corpora with a Swiss German TTS [DOGA21]

Error Detection in the Learners' Utterances and Generation of appropriate Feedback

An important functionality of a language learning system like ChaLL is to detect errors in the learners' utterances, categorize them, and to give appropriate feedback to help the learners advance.

There is a considerable amount of research for error detection in written language (see [NG14], [NG13]); however, additional difficulties arise when it comes to spoken content. As we have outlined in the previous section, STT transcriptions are not flawless, and error rates increase with the characteristics of the speech that we want to process in this project: spontaneous productions of non-native speakers. In addition, we expect that standard STT models have a tendency to recognize sequences that fit the model, thus partly correcting those errors which we want to detect. For example, an English STT system will match non-words or L1 insertions to its vocabulary and might thus transcribe a sentence such as "Today I was playing on the Wiese" into "Today I was playing on the PC".

To our knowledge, no spoken-language system exists that enables us to differentiate between these different sources of errors.

In written language, errors can be found and feedback delivered by detecting unexpected sequences [SCHN18a] or high surprisal [SCHN22]. Due to its statistical nature, error detection is not fully reliable in written, and even less so in spoken language. Additionally, STT systems aim at providing the most likely textual representation of speech input. By doing so, the system approximates its underlying language model, which typically corresponds to error-free training data stemming from large collections of written texts such

as Wikipedia, news articles or domain-specific corpora. In order for the system not to remediate actual errors, we will divert a representation of the respective speech input from the STT system prior to the determination of the most likely sequence of words. At that stage, we will make use of the n-best matches and their ranking to determine at which positions lexical errors might have occurred with high likelihood. N-best approaches have been explored for improving spoken language understanding for dialogue systems [GANE21, LI19, KHAN15], but not, to our knowledge, for the task of identifying errors in spoken production.

Other error types such as incorrect word order, non-idiomatic functional parts (e.g. prepositions) or transfer errors (e.g. false friends) can, on the other hand, be detected on the transcribed speech, as STT systems will not automatically correct them in most cases. Since all our judgments are based on probabilistic information, delivered feedback must be phrased carefully; suggestions or recasting the error candidates in context are more advisable than pointing out errors to the users.

Detecting the Learners' Skill Levels

To guide the conversations with a system like ChaLL into the „zone of proximal development", which is the optimal level of potential development, the system must adapt its language to the learner's level. An important preliminary step is thus to detect a learner's skill level.

While automatically predicting CEFR (Common European Framework of Reference for Languages20) skill levels on written texts (e.g., writing assignments) has been studied ([KERZ21], inter alia), detecting these levels in (transcribed) spoken language automatically remains an open challenge. Complementary to CEFR classifications, there are methods to gauge learning complexity of individual words and multi-word terms [ALFT18], [SHAR21] and readability metrics (e.g. Flesch-Kincaid, Dale Chall, ARI) that analyze word-per-sentence and syllables-per-word ratios and usage of (un)common words. Such approaches also offer an insight into a learner's language proficiency level.

Further, the question of how to produce context-sensitive responses that adhere to a specific CEFR level in a chatbot has to date not been studied and poses a novel challenge. A related task is text simplification of written texts. Approaches to text simplification apply methods from machine translation to translate an input sentence to a simplified version [ALTH21], and there are approaches that attempt to generate sentences at a specific CEFR level [SPRI21].

Dialog System to generate Responses

To implement and explore the above-mentioned capabilities, we will make use of existing chatbot technology. To ensure a high user motivation and retention, we are particularly interested in efforts to give state-of-the-art transformer-based chatbots a) a persona that is consistent across chat sessions [ZHAN18] and b) the ability to acquire knowledge on arbitrary topics during conversations online [DINA19]. A consistent persona, a memory of past conversations, and knowledge about a user's interest are key to ensuring user engagement and trust. BlenderBot 2.0 [KORN22a, KORN22b] is an example of such a chatbot with consistent persona, memory of past conversations, and the ability to acquire knowledge on a topic online. Since BlenderBot 2.0 is open-source, it provides a good basis to incorporate modules for error recognition and feedback generation, detecting the learners' language skill level, and adapting the chatbot's utterances accordingly.

Task-oriented dialogue systems [ZHAN20] are relevant regarding our planned focused mode where users solve structured tasks in a guided manner. Several datasets for training and evaluating conversational artificial intelligence exist that model situations such as booking a hotel, flight, or concert ticket [BUDZ18], but, to the best of our knowledge, none of them have been explored in the context of language learning and will thus have to be designed and implemented from scratch.

Text-to-Speech to synthesize Responses

Since chatbots like BlenderBot are text-based, our STT model will produce the needed texts from the user's spoken utterances, and a third-party solution for Text-to-Speech (e.g. Google Voice) will convert the chatbot's adapted textual responses into audio that can be played back to the learners, enabling a voice-based interaction. For optimal learning progress, we will use a text-to-speech system where we can adapt the speaking speed to the users' capabilities.

Appendix B: BlenderBot Installation Windows

1) https://parl.ai/docs/tutorial_quick.html#install
    a. git clone https://github.com/facebookresearch/ParlAI.git ~/ParlAI
2) Adjustments to ParlAI to make it run on Windows:
    a. https://github.com/facebookresearch/ParlAI/issues/4305#issuecomment-1013883793
    b. Otherwise Error `ModuleNotFoundError: No module named 'fcntl'`
    c. Also remove 'fcntl' from requirements.txt
3) Install C++ Build Tools
    a. https://learn.microsoft.com/en-us/answers/questions/136595/error-microsoft-visual-c-140-or-greater-is-require.html
    b. Dekstop Development with C++
4) Crate new Venv with Phyton 3.9
    a. pip install urllib==1.26.5
    b. pip install googleapis-common-protos==1.56.4
    c. pip install pypiwin32
        i. Otherwise, error: ImportError: parlai now requires iopath for some I/O operations. Please run `pip install iopath`
    d. cd ~/ParlAI; python setup.py develop
        i. Possibly repeat in case of error
5) Start Chatbot with Zoo Model
    a. parlai i -mf zoo:seeker/r2c2_blenderbot_400M/model -t blended_skill_talk
6) Or Start as Flask API
    a. parlai flask -mf zoo:seeker/r2c2_blenderbot_400M/model -t blended_skill_talk
    b. parlai flask -mf zoo:seeker/r2c2_blenderbot_400M/model --task task-master