DELFT UNIVERSITY OF TECHNOLOGY

MASTERS THESIS

Keymagine: Automatic Generation of Keyword Mnemonics with LLMs

Author:
Safouane el Hilali
Student number:
4914422

Thesis Supervisor:
Prof. Dr. M.M. SPECHT
Second Supervisor:
Erna Engelbrecht

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in the

Web Information Systems Group
Software Technology

June 26, 2024

TU Delft

Declaration of Authorship

I, Safouane el Hilali, declare that this thesis titled, "Keymagine: Automatic Generation of Keyword Mnemonics with LLMs" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: 

Date: 2024-06-26

DELFT UNIVERSITY OF TECHNOLOGY

Abstract

Electrical Engineering, Mathematics and Computer Science
Software Technology

Master of Science

Keymagine: Automatic Generation of Keyword Mnemonics with LLMs

by Safouane el Hilali

Memorizing vocabulary is a key part of second language acquisition; however, many people rely on rote memorization. Despite the proven effectiveness of the mnemonic keyword method for learning vocabulary, its usage remains limited because coming up with keywords can be time-consuming and creatively demanding. Previous solutions for automatically generating mnemonic keywords are inflexible and outdated, given the advancements in the field of Natural Language Processing driven by large language models (LLMs) in recent years. This study's research questions focus on how LLMs can be used to generate personalized mnemonic keywords and how these personalized mnemonics impact the learning experience and outcome compared to non-personalized approaches. By designing Keymagine, an LLM-powered system for keyword generation, we show that LLMs can effectively generate keywords through In-Context Learning and be personalized through user feedback. In an experimental evaluation, students (N = 22) used both Keymagine-generated and other automatically generated keywords to learn 36 German words. Results demonstrated a significantly higher perceived helpfulness of Keymagine-generated keywords and a significantly higher rate of recall.

Acknowledgements

All praise is due to God, who gave me the faculties and opportunities that allowed me to do my thesis without many problems, if any.

My sincerest gratitude goes to my thesis advisor Prof. Dr M. M. Specht for his guidance from when I was orientating until the end. I am also immensely thankful to my daily supervisor, E. Engelbrecht, for her dedicated support and insights from her relevant background. I am also grateful to Prof. L. C. Siebert for his contribution as a member of the graduation committee.

Special thanks to my peers in the Educational Technology research group for their input and feedback during the course of my thesis, and doubly so to C. Busropan for his creative input in giving my system the name Keymagine.

I am a thousandfold grateful to my family, and especially my parents, who always boundlessly supported me. Without them, neither the beginning nor the completion of this journey would have been possible.

Lastly, my gratitude to everyone who participated in my user study: ∗∗∗∗∗, ∗∗∗∗∗, ∗∗∗∗∗∗, ∗∗∗∗, ∗∗∗∗, ∗∗∗, ∗∗∗∗, ∗∗∗∗, ∗∗∗∗, ∗∗∗∗∗∗, ∗∗∗∗∗∗, ∗∗∗∗∗∗, ∗∗∗∗∗∗, ∗∗∗∗∗∗, ∗∗∗∗∗∗, ∗∗∗∗∗∗, ∗∗∗∗∗∗, ∗∗∗∗∗∗. They were all invaluable for the completion of this thesis.[1]

[1] Names redacted due to HREC regulations.

Contents

List of Figures

List of Tables

List of Abbreviations

L1	Native **L**anguage
L2	Second **L**anguage
CoT	**C**hain-**o**f-**T**hought prompting
ICL	**I**n-**C**ontext **L**earning
KWM	**K**ey**w**ord **M**ethod
LLM	**L**arge **L**anguage **M**odel
NLP	**N**atural **L**anguage **P**rocessing
SDXL	**S**table **D**iffusion **XL**

Chapter 1

Introduction

Vocabulary learning is an essential part of second language acquisition [1]. In European middle and high schools, pupils are expected to learn lists of vocabulary for foreign language classes such as English, French, or German [2]. Most pupils approach this task through rote repetition [3], [4]. Mnemonic techniques are a powerful way to learn vocabulary and have been shown to be more effective than rote rehearsal, yet they are not widely utilized [5], [6].

The biggest challenge in vocabulary learning is the difficulty in memorizing and recalling new words. Memorization involves encoding the word into memory, while recall requires retrieving the word from memory when needed. These processes and the learning outcome are heavily influenced by the type of vocabulary learning strategy or activity a learner uses [7]. Rote repetition, where the learner simply rehearses words repeatedly, often leads to shallow processing, making long-term retention difficult [8]. In contrast, mnemonic techniques have been shown to be highly effective in enhancing vocabulary retention and recall. These methods involve creating associations that force deeper cognitive processing and create stronger memory traces [9].

One such mnemonic technique is the keyword method. Pressley, Levin, and Delaney [10] explains that to use this method, a learner finds a word (the keyword) in their native (L1) language that sounds similar to the foreign (L2) word. They then construct a vivid mental image of an interaction between the L2 word and the keyword. The next time they see this word, they will be reminded of the keyword, sparking the mental image that reminds them of the L2 word's meaning. A more detailed explanation is given in Chapter 2. A great number of research studies have been carried out to investigate the keyword method's effectiveness, showing good results in word retention [9], [11], [12]. Yet, like most mnemonic-based learning strategies, it is rarely used by students [10], [13].

Two reasons have been given for the difference between its effectiveness and underutilization. First, coming up with good keywords can take a lot of creativity and time. Learners feel this time might be more effectively spent on rote rehearsal [14]. Japanese EFL learners reported enthusiasm for the keyword method's effectiveness, but realized that the time investment was too high. At times, they found the keyword method unsuitable for certain words and could not come up with any keyword, and would therefore rather spend their time on rote rehearsal [15]. Second, a lot of school-age children struggle to generate effective keywords, either due to the required cognitive load or simply because they lack the capability of generating effective keywords [16].

For this reason, this research project will provide a system for automatically generating mnemonic keywords for learners. To accomplish this, the system will be driven by a large language model to choose the best keywords based on pairs of vocabulary words.

1.1 Research Questions

The main research question of the thesis is:
How can generative AI be used to generate personalized keyword mnemonics for learning vocabulary?

To formulate an answer to this question, multiple sub-research questions have been formulated.

- **Sub-question 1: How can large language models be used in keyword mnemonic generation?**
  This question aims to find out how exactly LLMs can generate and choose effective keywords.
- **Sub-question 2: How can automatically generated keyword mnemonics be personalized to a learner?**
  This question aims to find out how each learner does not have to use the same keywords but can use keywords that are closer to their preferences.
- **Sub-question 3: How do personalized automated keyword mnemonics affect the learning outcome and experience compared to non-personalized automated keywords?**
  This question aims to test the effect of personalized automated keywords on the learning experience and outcome of its users, specifically the recall of vocabulary and the helpfulness.

1.2 Overview

This report is structured as follows: in Chapter 2, a theoretical background of vocabulary learning and the keyword method is provided. Additionally, related work on the automatic generation of keywords is discussed. Chapter 3 describes the solution designed to address sub-questions 1 and 2 regarding the challenge of automatically generating keywords. Chapter 4 outlines the research setup used to evaluate the system with human participants and presents the results of the evaluation. Chapter 5 discusses the results. Finally, a conclusion is offered in Chapter 6.

Appendix A displays screenshots from all screens that participants in the evaluation encountered. Appendix B shows the verbal and visual cues used in the evaluation for each keyword generated by *Transphoner*. Appendix C shows my thesis proposal.

Chapter 2

Background & Related Work

In this section, we will look at past research on the processes underlying the acquisition of second-language vocabulary, keyword mnemonic techniques, and steps that have been taken toward automating the process of generating keyword mnemonics.

To be able to aid learners well in their quest to learn foreign language vocabulary, it is essential to know what cognitive processes are involved when learning vocabulary, both with and without the help of mnemonic devices. A lot of psychological models of cognition can be used to understand and work with educational interventions. It is important to note that each just sheds light on part of the picture, and doesn't provide the entire blueprint.

2.1 Memory Processes

A great amount of vocabulary learning research overlaps with research on memory in general, as the basis of vocabulary learning is memorization. The most popular model of memory is the *multistore model* or *three-stage model* [17]. This theory posits that information to be memorized is processed in three stages: encoding, storage, and retrieval. Encoding is the process of converting a sensory input into a form that can be processed and stored in memory. When memorized, it is in the storage state. Finally, retrieval is the process of retrieving said information from memory.

These steps cannot be taken independently of each other. If information is not encoded well enough, it can be hard or impossible to store and retrieve. Likewise, bad storage impedes retrieval, and all stages must succeed for a person to 'remember'.

Encoding is especially important because it is the first step in the process of learning vocabulary [9]. The topic of encoding is vast and encompasses all the ways in which a word is committed to memory, the process that most people consider when they think of 'learning.'

Some theories of encoding that are relevant to the topic of Section 2.3 are mentioned in [9]: selective encoding, transformational encoding, and elaborative encoding. Selective encoding focuses on filtering out irrelevant information and emphasizing the important aspects that will aid in better recall. Transformational encoding involves changing the raw information into a more manageable and memorable format, mostly by using simplifications or summarizations to make the information easier to retain. Lastly, elaborative encoding enhances the memorability of information by adding additional context or connections, making it richer and more interconnected within the existing memory network.

Dual Coding Theory (DCT), proposed by Allan Paivio states that information is stored in memory in two systems: nonverbal and verbal [18]. The verbal system is specialized in processing and storing codes for words like "book," "Buch," "study" and "life", representing both concrete and abstract concepts, while the nonverbal

system is specialized in concrete concepts [19]. An abstract concept like "success" can only conjure imagery through associated concrete concepts like "trophy." While they are stored independently, they are interconnected.

2.2 Vocabulary Learning

Vocabulary acquisition is a form of memorization, so it proceeds in the aforementioned three stages. It is important to recognize that it involves more than memorization, since it also encompasses understanding the meaning, usage, and nuances of words in various contexts.

2.2.1 Vocabulary 'Acquisition?'

Jiang emphasizes the importance of defining what it means to have learned a word [20]. Many experimental studies measure vocabulary acquisition by testing subjects' ability to recognize, recall, or provide translations of foreign language words. Jiang argues that this only indicates whether a word is remembered, not truly acquired, which is much broader.

Nevertheless, this straightforward working definition can be applied for practical reasons but has to be accepted explicitly rather than implicitly. Thus, we can say that our evaluation given in Chapter 4 only measures the effectiveness of our system for the first stage of foreign vocabulary learning, namely, the association between the L2 word and an existing meaning in their first (L1) language [21].

2.3 The Keyword Method

According to its original design, the mnemonic keyword method (also called the keyword method from here on), is a two-phase process [22]. In the first phase, the subject finds a word in their L1 language that sounds like (part of) the L2 word. This is the keyword. In the second phase, the subject creates a (mental) image of the definition of the L2 word and its keyword interaction.

To give an example, *dormir* means sleep in French. A good keyword for this word would be *door*. With this keyword, we can create a mental picture of somebody sleeping soundly on top of a door lying flat. One could also create a meaningful sentence, like "You should close the *door* before you *sleep*" [10].

2.3.1 Underlying Mechanisms

The efficacy of this method can be explained by its engagement of both code systems in the framework of Dual Code Theory [18]. By creating vivid mental images, both the verbal and non-verbal are engaged and more pathways are created for retrieval. A meaningful visual image also helps to form a base for memory to store a new L2 word's meaning Shapiro and Waters, which is transformational encoding.

Elaborative encoding also plays a role since word pairs are augmented with new information, i.e. the keyword. This makes the items stand apart from each other even more by introducing unique cues Worthen and Hunt. In the case of the keyword method, the keyword links the L2 word to a unique identifier in the form of imagery.

2.4 Properties of Good Keywords

Raugh and Atkinson gave three criteria that keyword designers should adhere to [11]: first, the keyword sounds or looks as similar as possible to the foreign word or part of the foreign word. Second, The keyword should be easily representable in imagery. Abstract words can work if they are associated with symbology [24]. Third and lastly, the keywords used to learn a list on a particular day should be unique, so that no keyword is used to learn more than one word.

2.4.1 Phonetic Similarity

The link between the L2 word and the keyword by means of sound is called the "acoustic link" by Raugh and Atkinson [11]. For there to be a strong association between the two words, there does not have to be a phonetic similarity between the entirety of the two words, as the keyword may also sound like only part of the L2 word.

2.4.2 Orthographic Similarity

Phonetic similarity is not the only way by which the keyword and L2 word may be associated, as they can be orthographically similar as well [11], [25]. This means they may be similar in spelling, despite their pronunciation being dissimilar. As an example, take the word *"caballo"*, which is Spanish for *horse*. One can take *ball* as the keyword because of its orthographic similarity with caballo, even though the word is pronounced /kaˈbajo/ (ka-ba-yo). For word pairs between languages with similar scripts, this is another good criterion, since comparing written words is an effective form of learning [26].

2.4.3 Psychoactive Properties

Psychoactive properties of the keyword relate to the keyword's interaction with the subject's psyche. This has two components: the ease with which the keyword and target word can be visualized, and the relation of the keyword to the subject's experiences.

The ease with which a word evokes a sensory mental image is called imageability [27]. Ellis and Beaton found that a keyword's effectiveness is influenced in part by its imageability [28]. Atkinson and Raugh found that the imageability of the L1 word also plays a great role in the recall of words [22]. In a study where university students learned target words using the keyword method that were evenly split between highly imageable and lowly imageable words, it was found that the students were able to recall the highly imageable words more accurately than the lowly imageable words. This effect was seen in both immediate and delayed recall tasks [23].

The relation of the keyword to the subject's experiences concerns whether the word is familiar to the subject and how well they can recognize the keyword in the L2 word **campos2004drawing** , Campos, Amor, and González. This will become clearer in Section 2.5.4.

2.5 Mediating Factors

A number of factors have been studied in order to ascertain their influence on the retention rate of learners using the keyword method.

2.5.1 Short-Term and Long-Term Recall

One of the factors that has been investigated is the impact of the keyword method on short-term and long-term recall. Some papers that investigated the retention rate of subjects who used the keyword method found that the immediate recall of the keyword method group was higher than the control group. However, this advantage declined or even disappeared after one or more weeks [30], [31]. Some studies showed that subjects using rote learning had at most a slight decline over a week compared to subjects using the keyword method, although their mean recall rate was poorer in both short-term and long-term tests [29], [32].

2.5.2 Self-Made and Given Imagery

The keyword method works by the subject relating the keyword and target word through an image. Whether this image is imagined by the subject or provided to them makes a large difference in subjects' recall rate. Some studies have found that imposed imagery (visual cues) improves both immediate and delayed recall [10], [29], [30], [33]. It was also found to suppress the decline in recall rate by Thomas and Wang [32]. Others have found no difference or a negative effect on recall for subjects that were provided with imagery [34], [35]

2.5.3 Self-Made and given Keywords

The question of imposition applies not just to images, but to the keywords themselves too. The reasoning behind the opinion that keywords should be provided by the experimenter is that the participants can't reliably find the best keywords. On the other hand, the view that the keyword should be generated by the subject is based on the reasoning that the experimenter and subject may conflict in their ways of creating mental associations [29]. Compared to the question of imposed versus induced images, studies are mixed on whether imposed keywords are more effective than subject-generated keywords [10]. In a study comparing self-generated and imposed mnemonics for learning chemistry concepts, self-generated mnemonics boosted recall of the mnemonic more than received mnemonics [36]. However, it does not improve recall of the corresponding chemistry information. They acknowledge their learning material was made up of different kinds of facts, like lists and definitions, and different types of content are more conducive to creating effective mnemonics than vocabulary lists of word pairs.

A 2×2 study by Shapiro and Waters found that the difference in recall between subjects that were given keywords and those that had to make their own was insignificant [23]. However, there are still valid reasons for providing subjects with keywords, as discussed in Chapter 1.

2.5.4 Peer-Generated Keywords

Apart from subject-generated and experimenter-generated keywords, there is a combined approach, namely providing keywords that have been generated by the subject's peers (i.e. of the same social and educational background). Several studies in

this have been undertaken in this area [29], [37]. Campos et al. [29] found significantly better recall for highly vivid words for the group using peer-generated keywords than the group using subject-generated keywords. Another study [37] found significantly higher immediate recall for words of low vividity in the group where subjects used peer-generated keywords compared to the group using experimenter-generated keywords. Yet another paper found peer-generated keywords to outperform subject- and experimenter-generated keywords [38]. This demonstrates the importance of how keywords relate to the learner's psyche. In short, subjects profit from having keywords provided, especially when are psychologically a good "fit" for the subject.

2.6 Auto-generated keywords

There have been a few papers proposing systems for the automatic generation of mnemonic keywords. The most cited paper that tackles this technical challenge is "Transphoner: Automated mnemonic keyword generation" [26]. Using an algorithmic approach, it looks up the input word in several dictionaries to get its pronunciation, meaning, and other attributes like imageability. It searches for candidate keywords in the target language and uses an optimization algorithm to find the best-matching keyword sequence that maximizes phonetic, semantic, imageability, and orthographic similarity.

Özbal, Pighin, and Strapparava published another paper around the same time that used an algorithmic approach to generate keywords that were based on orthographic and phonetic similarity, as well as creative sentences that included the keyword and L1 word [39]. Unlike "Transphoner: Automated mnemonic keyword generation" [26], this process required a human to select the most suitable generated keywords and sentences for assessment.

Anonthanasap, Ketna, and Leelanupab created a system for automatically generating keywords for English-Japanese vocabulary pairs [40]. Its focus is on phonetic similarity, and it has an elaborate algorithm called Jemsoundex for finding similar-sounding words for mnemonic keywords, based on a modified Soundex algorithm [41].

A study by Anonthanasap, He, Takashima, *et al.* provides an interactive interface where learners can browse foreign language words and see phonetically similar keywords [42]. It only incorporates phonetic similarity to suggest keywords so it does not bring much innovation to the table there, but it showed that participants benefited from the suggestion of images more than static visualization.

SmartPhone [35] is the most recent study and the only one to integrate large language models in the pipeline. It uses them to generate verbal and visual cues for automatically generated keywords by Transphoner. In their user study where automatically generated verbal cues were compared with manually created ones, they found that automatically generated verbal cues alone did not improve learning over just the keyword, and visual cues were perceived as helpful but performance was mixed.

2.7 Generative AI in Education

Generative Artificial Intelligence is an inventive form of AI that does not just analyze existing data like classical AI tasks, such as classification and detection, but can

generate new content [43]. This field of research has gained immense popularity since 2021, with the arrival of ChatGPT [44].

2.7.1 Large Language Models

Large language models (LLMs) fall under the umbrella of language modeling. This is a field of research with the goal of modeling the likelihood of sequences of words, to predict the probabilities of future tokens or absent tokens [45]. Language modeling has evolved through four key phases: statistical language models (SLM), neural language models (NLM), pre-trained language models (PLM), and finally the current phase of LLMs.

With these phases, the field has shifted from statistical to neural models [45]. The focus is now on pre-trained language models (PLMs) that use the transformer architecture [46] and are trained on large-scale datasets of textual data, demonstrating strong capabilities in a wide range of natural language processing (NLP) tasks [45]. These models, when scaled up, not only enhance their overall capabilities but also develop new skills, such as in-context learning, which are not observed in smaller models like BERT. The term "large language models" refers to these vastly larger PLMs, which may contain up to hundreds of billions of parameters. Notable examples of such models include GPT-4, LLaMa, and Mistral [45].

Pretrained foundation models (PFMs), such as BERT and GPT-4 are trained on vast datasets, thereby forming the core for numerous applications [47]. Hence the term 'foundation' is used to describe these models, as they serve as the foundation for many different tasks across a diverse range of domains, including language, vision, and robotics. However, NLP is the domain that has been most profoundly impacted by the coming of these models [48]. Since the early PFMs BERT and ELMo were introduced, the field of NLP shifted to primarily using foundation models as the main research instrument.

Previously, different groups developed unique models for specific NLP tasks like parsing or translation, often using complex pipelines. Now, a single foundation model is typically adapted with a small amount of task-specific data to handle multiple tasks. This streamlined method is more effective and often beats the older, more complicated systems [48].

2.7.2 LLMs in Education

Both students and teachers have found applications for LLMs in education [44]. Students are using LLMs to answer questions, provide guidance, and correct errors. Teachers have found a use for it in automatic grading, generating questions, and what is of particular interest to us, material creation.

The great potential of LLMs to aid teachers in creating educational material has been explored to create material for language learning, like Koraishi as cited in [44], who used it to adapt material for an English as a Foreign Language class to different proficiency levels.

2.7.3 LLMs and Adaptive Learning

Personalized and adaptive learning has traditionally been limited to recommending different resources to students based on their existing knowledge. LLMs open up the possibility to personalize the educational experience in more innovative ways,

for example by providing educational material in the styles of different personas [49].

Existing work on personalized and adaptive learning is classified by Wang, Xu, Li, *et al.* in two categories: knowledge tracing and content personalization [44]. Knowledge tracing is assessing the knowledge of students so that their learning path can be most effectively carved out. Content personalization, on the other hand, focuses on selecting and presenting learning materials that best suit the individual student's interests and learning style.

LLMs have been applied to both categories. A recent example of LLM-powered knowledge tracing is KAR^{3}L [50], which is a student model that predicts how well a student can recall flashcards. It integrates BERT embeddings and retrieval techniques, thus taking into account not only a student's previous performance but also semantic information in the cards, which especially improves recall prediction for cards not previously studied. This approach enables KAR^{3}L to effectively introduce new flashcards that address gaps in a student's knowledge, moving beyond mere repetition of the studied material. This model significantly outperforms traditional student models by capturing semantic relations between flashcards.

In a study by Pesovski, Santos, Henriques, *et al.* [49], a tool was developed within a learning management system at a software engineering college that personalizes learning materials by generating them in three styles based on specified learning outcomes. It offered materials in the traditional professorial style and others which incorporate pop-culture elements, such as Batman and Wednesday Addams, to diversify the educational experience. The preliminary study involving 20 students showed that while the traditional style was predominantly used, the variety in presentation styles supported increased engagement and study time, especially among students who were initially less familiar with the topics. These findings suggest that LLMs can be effective in delivering personalized educational content that caters to various learning preferences and needs.

2.7.4 LLMs and Mnemonics

The potential of LLMs and text-to-image models for creating mnemonics has also been explored already. Wong and Wolf demonstrate how LLMs can generate embellished acronym mnemonics [51]. This type of mnemonic is used to remember lists of words by creating a more memorable sentence where each word starts with the first letter of the items to be recalled [9]. A commonly used example is "**D**eath **A**lways **B**rings **G**reat **A**cceptance," which helps remember the five stages of grief (Denial, Anger, Bargaining, Grief, Acceptance). Wong and Wolf's patent disclosure uses LLMs to generate this type of mnemonic and text-to-image models to create accompanying pictures as visual aids. The authors also included an element of personalization by letting users modify words of the generated mnemonic.

2.7.5 Text-to-Image Models

The history of text-to-image models started with AlignDRAW, published in 2015 as an early effort which produced unrealistic results [52]. Generative Adversarial Networks (GANs) emerged in 2016, which improved the quality but still faced issues like training stability and limited data handling. A real breakthrough came with the advent of diffusion models, which significantly advanced the state of text-to-image generation. Unlike their predecessors, diffusion models are characterized by their ability to handle detailed and complex image generation tasks with greater stability

during training [53]. They achieve this through a process that iteratively refines images by reversing a diffusion process. In essence, it starts from noise and adds structure step by step until a coherent image is formed that corresponds to a textual description [53].

Text-to-image diffusion models can generate vivid, detailed images from textual descriptions. What is especially relevant to the generation of visual cues for mnemonics is that the models can combine unrelated concepts in reasonable ways [54]. This is useful to us since keyword mnemonics rely on forming unique and memorable associations between disparate elements to enhance recall.

Chapter 3

System Design

The background in memory and mnemonics research described in Chapter 2 showed that peer-generated keyword mnemonics combine the benefits of self-generated and teacher-generated keywords, as they don't require a learner's time and creativity, but still suit the learner's consciousness. Section 2.6 further presented papers that proposed solutions to digitally generate keywords. To the best of my knowledge, the last one of these papers was published in 2017 [55], before the onset of LLMs. In the meantime, LLMs have been exceeding traditional approaches to many NLP problems, so there's a clear gap in the research where there has been no attempt to use LLMs for keyword mnemonic generation.

It's therefore up to us to bridge this gap. This section starts by outlining the design of the keyword generation system, which is called **Keymagine**, as it was implemented and used in the user evaluation described in Chapter 4. It also describes features of the system that are not tested or implemented. Building Keymagine answers the first two sub-questions posed in Chapter 1: "How can large language models be used in keyword mnemonic generation?" and "How can automatically generated keyword mnemonics be personalized to a learner?"

3.1 Requirements Analysis

Based on the features of good vocabulary-learning and mnemonic tools mentioned in the last chapter, a list can be made of features that this system should and should not have. We use the MoSCoW method [56] to separate these features into those that are necessary (must have), those that are important but not critical (should have), those that are nice to have and could be implemented in the future (could have), and those that are not required at all (won't have).

Must Have

Requirement 1 is the first and most crucial component that the system must have. Unlike previous approaches [26], [39], [40], this system is (one of) the first to use large language models for keyword generation, rather than deterministic algorithms. Needless to say, without this requirement, there would be no system.

1. The system must generate keywords by prompting an LLM.

Requirement 2 states that the generated keywords must be in English. There are three reasons for this. First, the word list from [28] which is used in the evaluation contains English-German word pairs, which makes it logical to generate English keywords. Second, the participants recruited for the evaluation all understand English. Third, LLMs are only as good as the data they're trained on. Most LLMs are

trained on datasets where the vast majority of the text is in English, which can result in reduced accuracy when executing tasks in other languages [57]. Therefore generating keywords in English is a fundamental requirement for this system.

2. The system must generate keywords in English.

Requirement 3 states that the system should generate a keyword close to the L2 word in either sound or spelling. It is for good reason that this is the first criterion of good keywords given by [11]. As seen in Section 2.3, this is of utmost importance for the keyword method to be used effectively, since it has the greatest effect on encoding and retrieval of the word pair.

3. The system must generate keywords with high phonetic or orthographic similarity to the L2 word.

Requirement 4 is necessary as the goal of our system is to aid the user in picking the best keyword for their needs. It would be ideal if our system consistently generated the optimal keyword for the learner, but the preference of the learner for a particular keyword can vary considerably. Furthermore, in the process of deciding between the proposed keywords, the learner may get inspired and come up with their own keyword which they prefer most, necessitating the option to enter their self-generated keyword.

4. The system must propose generated keywords to the user but also allow them to input their own keyword.

Ranking keywords as stated in requirement 5 is also a must as we are trying to minimize the cognitive load on the user when choosing keywords. The generated keywords will be ranked by their similarity in sound and/or spelling, but they will also be evaluated based on their imageability, as we've seen in Section 2.4.3 that this is one of the highest mediating factors influencing the effectiveness of a keyword in recall.

5. The system's proposed keywords must be ranked by imageability and phonetic or orthographic similarity

In order to learn a list of word pairs, the system must be able to take a file with a list of L1 and L2 word pairs and be able to present them to the user one at a time. This is especially relevant for the user evaluation

6. The system must take a list of pairs of English and L2 words and present them sequentially to the user.

Finally, the user interface must be accessible through a web browser and be as simple as possible, in order to minimize distractions during the learning process.

7. The system must be accessible through a simple web interface.

Should Have

The system should include the generation of verbal and visual cues, even though these are not essential to its functioning. While research is mixed on the effectiveness of providing visual cues to learners, the majority of studies in Section 2.5.2 have demonstrated positive outcomes. Therefore, it is our opinion that the system should include this requirement. With the advent of generative AI, this is the first time in history that these cues can be generated at scale with minimal human energy needed, so it is worth taking the opportunity to put it to the test.

8. The system should generate verbal and visual cues along with keywords.

Could Have

It's possible for the system to generate and rank keywords for a user based on the words they already know. This is a form of knowledge tracing, as the user's performance is tracked and used to modify their future learning material. This is an idea for future iterations of the system and is not implemented here.

9. The system could generate keywords based on the user's previously learned words.

Content personalization can take many forms. Some ideas include generating verbal cues related to the learner's media preferences or generating visual cues in the learner's preferred style, such as painting, cartoon, or photorealistic. However, this introduces many new variables that need to be tested, for which time and resources were not available. Content personalization will therefore not be included in the current iteration of the system, but is an interesting avenue to explore.

10. The system could generate keywords or cues based on the user's persona.

Won't have

The semantic similarity between two words is their similarity in meaning. Raugh and Atkinson [11] described a modified version of their keyword method in which keywords are chosen based on phonetic and semantic similarity. For example, the keyword *"curt"* is used for the German word *"kurz"*, which means short. However, there are two problems with this method method. The first problem is that this approach significantly constrains the potential keyword pool, since filtering by both meaning and sound naturally reduces the number of options. The second reason is that this restriction is unnecessary, as there is no evidence that high semantic similarity positively affects the use of the keyword mnemonic. Keywords can be completely unrelated, as long as they are well encoded. In fact, unrelated keywords with bizarre imagery have been shown to be at least as effective [9] or more effective [23], [58] than common imagery. Transphoner takes semantic similarity into the equation, but does not explain this decision [26]. We therefore conclude that semantic similarity is wholly unnecessary to keyword generation.

11. The system will not take into account semantic similarity between the L2 word and keyword.

3.2 High-level Overview

This section provides a high-level overview of the technology used to create Keymagine.

3.2.1 Technology

The entire stack is written in Python. It was chosen for its extensive community support and large ecosystem of libraries and frameworks, making it possible to use it to create both an LLM program and a web application.

DSPy To construct the language model program for our project, we selected DSPy as the framework [59]. Although several frameworks are available for managing language models, DSPy stood out because it represents a new paradigm in programming language models. Unlike traditional frameworks like LangChain, which primarily focus on prompt engineering, DSPy introduces a more systematic approach, separating the flow of the program from the prompts, saving time from "prompt engineering". It also allows for algorithmically optimizing LLM programs through built-in optimizers.

DSPy programs involve a series of calls to language models (LMs) organized into DSPy modules. Each module includes three key internal parameters: the (local) language model's weights, the instructions provided, and the input and output examples. DSPy can optimize these three parameters with multi-stage algorithms with specialized optimization techniques for refining instructions and creating examples. Because they systematically explore more options and directly optimize against specified metrics, DSPy compilers can produce high-quality prompts and examples [60].

These prompts can be very different from what a person would have considered. Battle and Gollapudi [61] optimized several LLMs for a mathematical reasoning task. The highest-scoring optimized prompt for Llama2-70B was styled as the logbook of a captain from Star Trek, asking the system to "plot a course through this turbulence and locate the source of the anomaly."

Open Source LLMs We opted to use only open-source LLMs in our system for three reasons. The first reason is scientific reproducibility. Proprietary models can only be used through an external party's API, which can be changed or withheld at any time. This poses a risk for future researchers seeking to duplicate results. The second reason is that the initial conceptions of Keymagine's pipeline included personal information, such as age and hobbies, being used to prompt the LLM for personalized keywords by including user info in the prompt. Sending personal information to external APIs poses privacy risks. Using open-source LLMs allows this information to stay in a controlled environment where all sensitive data remains local. The third reason is that open-source LLMs are very capable and approach proprietary models in certain tasks [62]. Since the prompts used in Keymagine are not particularly complex, open-source LLMs fare well, and have lower operational costs than most proprietary models, making them a viable choice. Another reassurance that using open-source LLMs is not a bad idea is that Keymagine could be further optimized automatically by DSPy.

The LLM chosen to run this entire stack is Meta's Llama3-70B, which was released on the 18th of April, 2024 [63]. This decision was influenced by several factors. I had confidence in Llama 3 since Llama 2 had widespread success in local language model enthusiasts online, and the new version quickly gained positive feedback and high ratings from the community. Other language models were also tested. A strong contender was Mixtral-8x7B, which was one of the other top-performing open-source models, and whose generated keywords were sometimes different but overall on par with those of Llama 3. Other models, like Mistral-7B and Llama3-7B, were also tested; however, they failed to generate keywords with the desired level of diversity and quality.

Many new LLM releases come with two versions, the base model and an 'instruct' model. The base model simply generates text by predicting the next word

in a sequence, as an LLM should. Many users want the model to follow their instructions in a helpful manner instead, so 'Instruct' models are released, which are fine-tuned versions of the LLM, specifically optimized to follow user instructions and perform tasks like generating code or chatting with the user [64]. Since the tasks the LLM has to perform in this system don't require following detailed instructions, the base model was chosen.

Tech stack The web application used for user evaluation is built with a Flask backend, using the HTMX and Alpinejs libraries for sending AJAX requests and minor user interactions. Flask facilitated development because it was easy to learn and get started with. HTMX and Alpinejs are two javascript libraries that provide a simple way to add some Javascript interactivity to the user interface. User data is saved in MongoDB, which was chosen for its flexibility, as it didn't constrain the author to a fixed SQL schema or the need to migrate databases if the schema changed. Finally, the entire stack runs on Docker, so that the entire stack can be started with a single command and potential problems with dependencies are minimized.

3.3 Pipeline

Overall, the keyword generation pipeline works as follows. The pipeline of Keymagine has four inputs: the foreign language, foreign word, native language, and native word. Initially, the foreign language and L2 word are processed by a module that generates a list of ten candidate keywords. Each of these keywords is then evaluated by another module to get their imageability. The final ranking of keywords is determined by multiplying their frequency of occurrence by their imageability scores. This process results in a ranked list of keywords, which is then returned and proposed to the user.

There are two other LLM-powered modules. One of them takes a keyword and native word, and generates a verbal cue. The other module takes the verbal cue and generates a prompt for a text-to-image model to generate a visual cue.

This pipeline is illustrated in the flowchart displayed in Fig. 3.1.

3.4 Best Keyword Selection

This subsection details how the candidate keywords are generated, and how they are ranked according to two factors: frequency and imageability.

3.5 Prompting for Keywords

Generating candidate keywords is done using a DSPy signature with the task instruction "Generate an English word that looks similar to the foreign word." The signature takes `language` and `foreign_word` and outputs `similar_word`. The full signature class is displayed in Listing 1.

This signature is used in a DSPy module whereby the LLM is called with those exact instructions, as shown in Listing 2. Upon every call, the LLM was configured to generate 10 completions at once for the prompt. This is a cost- and time-effective way to get multiple completions, because the prompt only has to be processed once. Hence only the output tokens are multiplied by ten. Since it only generates a single keyword before halting, the multiplied computational cost is minimal.

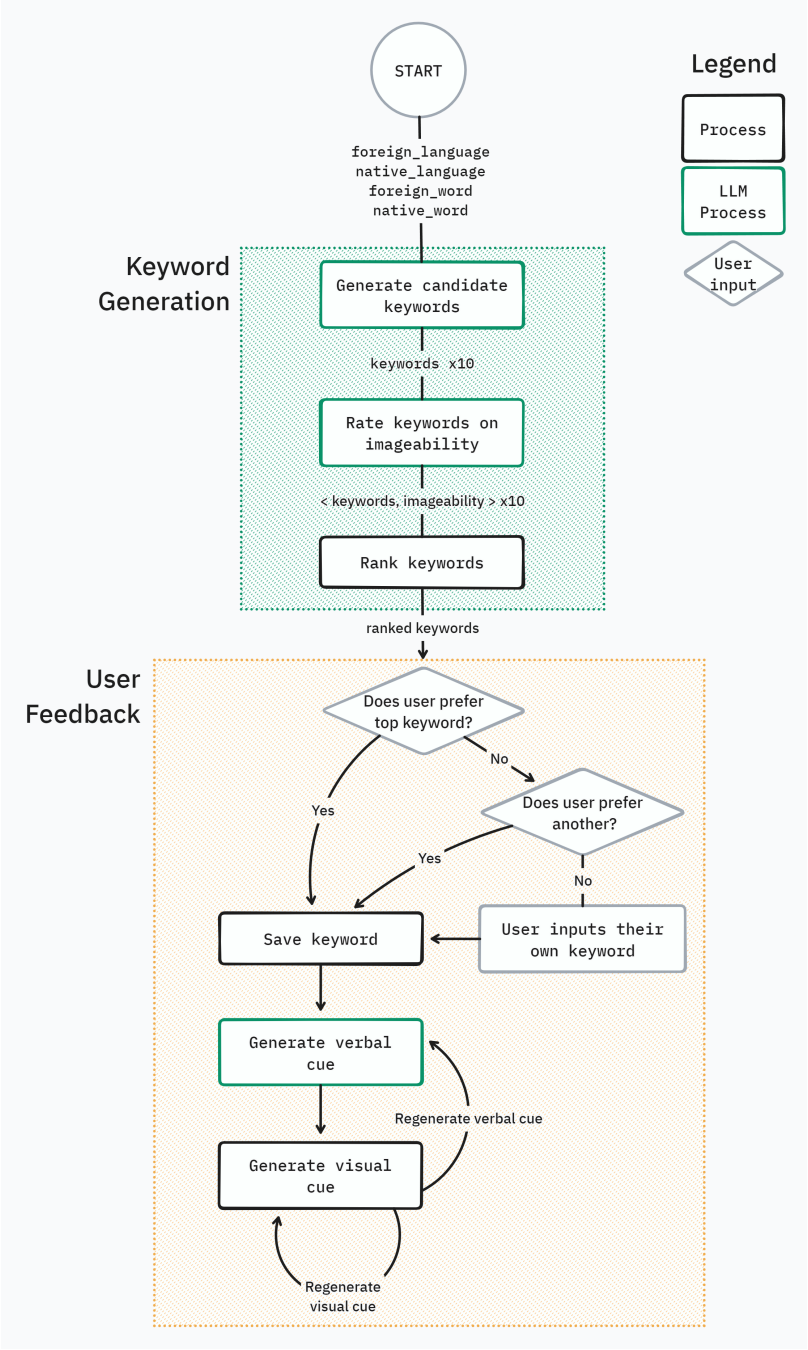


FIGURE 3.1: Flowchart of the keyword generation pipeline.

`n=10` was chosen because a lower number of return sequences than 10 resulted in outputs that lacked sufficient diversity, while a higher number led to excessive diversity. While it was initially expected that increasing `n` would maintain the same probability for each keyword to be generated, this actually resulted in a broader range of unique keywords, each appearing only once or twice. If `n` was too small, the amount of unique candidates being generated was logically too constrained. `n=10` was chosen as a balanced trade-off.

The LLM was not called by just giving the prompt and input values. To guide the LLM to follow the instructions correctly, we used in-context learning. In-context learning is a technique where the model is given examples within the prompt as

```python
class SimilarOrthography(dspy.Signature):
    """Generate an English word that looks similar to the foreign
    ↪ word."""
    language = dspy.InputField()
    foreign_word = dspy.InputField()
    similar_word = dspy.OutputField(desc="An English word with similar
    ↪ spelling to foreign_word. Don't just translate it.")
```

LISTING 1: The DSPy signature used to instruct the LLM.

```python
class SimilarWordModule(dspy.Module):
    def __init__(self):
        super().__init__()
        self.generate_word = dspy.Predict(SimilarOrthography)

    def forward(self, language, foreign_word) -> list[str]:
        return self.generate_word(language=language,
        ↪ foreign_word=foreign_word, config=dict(n=10,
        ↪ stop="\n")).completions.similar_word
```

LISTING 2: The DSPy module that generates the keywords.

demonstrations to learn and adapt its response to [65]. It is a strong alternative to fine-tuning, as it allows the LLM to understand the task without requiring extensive retraining on task-specific data, while still achieving remarkable results [66].

DSPy's `LabeledFewShot` optimizer provides an easy way to use this technique. One simply specifies the desired maximum amount of examples to use and provides it with a set of labeled examples. Listing 3 shows how this was implemented in the program, with 1 example for illustration.

Thirty-three examples were used to instantiate to compile the program. Some of them were made by the author (e.g. French: *raconter*, English: *to tell*, keyword: *raccoon*). Others are examples from previous papers (e.g. Japanese: *arashi*, English: *storm*, keyword: *airship*) [22], [40]. The bulk came from two books by Gruneberg: *German by Association* [68] and *French by Association* [67]. They are part of the *Linkword* method, which is a series of books, computer programs, and language courses founded by Dr. Gruneberg, which essentially teaches vocabulary through the KWM, with hundreds of vocabulary items and keywords per book. The examples from these books were selected semi-randomly by the author, ensuring they were simple, unambiguous, and did not use archaic language.

3.6 Rating Imageability

Since we established in Section 2.4.3 that the imageability of a keyword is positively correlated with its effectiveness in the KWM, it is clear keyword candidates should be ranked on the imageability.

One way to do this is through psycholinguistic databases that contain imageability entries for words. An example is the MRC Psycholinguistic Database, which contains 9,240 entries in its imageability database on a scale from 100 to 700 [69].

```
class CompiledCandidatesGenerator():
    def __init__(self):
        trainset = [
            # ...
            Example(language="french", foreign_word="poisson",
            ↪  similar_word="poison").with_inputs("language",
            ↪  "foreign_word")
            # ...
        ]
        lfs_optimizer = LabeledFewShot(k=60)
        self.similarword_lfs =
        ↪  lfs_optimizer.compile(SimilarWordModule(),
        ↪  trainset=trainset)
```

LISTING 3: How in-context-learning is set up in DSPy.

The problem with these is they are too fine-grained and only have imageability entries for a small portion of words in their databases. For our purposes, it's enough to know whether a word has high or low imageability.

Thus we use an LLM-based module that takes a single word and categorizes it as either high imageability, moderate imageability, low imageability, or symbolizable. The module uses the same `LabeledFewShot` DSPy optimizer as mentioned previously, with five examples from each category. The example words for low imageability were selected from the first quartile of words from the MRC Psycholinguistic Database [69]. Words with moderate imageability were taken from the interquartile range, while words with high imageability were taken from the third quartile. The examples of symbolizable words (love, justice, democracy, horror, and peace) were devised by the author.

Each example in the module includes not only the input word and its corresponding imageability category but also a reasoning step to improve the reliability of the categorization. Eliciting intermediary reasoning steps from LLMs before the final answer is a technique known as Chain-of-Thought reasoning (CoT) [70]. CoT prompting has been shown to greatly enhance the performance of LLMs on a variety of reasoning tasks, including mathematical reasoning, commonsense reasoning, and complex problem-solving.

By including reasoning steps, the LLM can better 'understand' the rationale behind the categorization, which results in more consistent classifications. In this case, instead of simply labeling a word as low imageability, the example would also include an explanation, such as "This word is categorized as low imageability because it is an abstract concept." This method helps the LLM internalize the criteria for each category, resulting in more robust and reliable outputs.

3.7 Verbal Cue Generation

The verbal cue module takes the native word and keyword, and outputs a sentence instructing the learner to imagine a specific scenario. We use twelve examples that were taken from the material used in the experiment described by Ellis and Beaton [28]. Table 3.1 shows the full list of few-shot examples used for the evaluation.

English word	Keyword	Verbal Cue
plate	fortune-teller	Imagine a fortune-teller with a pile of silver plates
kitchen	cook	Imagine your kitchen and a cook in it
to rent	meat	Imagine you rent meat to friends in your room
to pay	sailor	Imagine sailors pay for hot rum
cliff	clip	Imagine nail-clippers on a cliff
flag	fan	Imagine a flag on a fan
to call	roof	Imagine you call a friend to put a new roof on a cottage
to dig	crab	Imagine crabs dig holes in the sand
scissors	shear	Imagine shears besides a pair of scissors
lawn	raisin	Imagine your lawn covered in raisins
to push	store	Imagine you push stores in a cupboard
to paint	striking	Imagine strikers paint slogans on walls

TABLE 3.1: English Words, Keywords and Verbal Cues Used for Few-Shot Examples

3.8 Visual Cue Generation

To generate visual cues, there is a module that takes a prompt and returns an image from a text-to-image AI model, and an LLM-based module to create the prompt for the aforementioned module.

The latter module takes the verbal cue generated by the verbal cue module, and simplifies it by taking out superfluous words, leaving only the principal subjects in the prompt. To illustrate, it would take 'Imagine your lawn covered in raisins' and return 'lawn covered in raisins', or take 'Imagine sailors pay for hot rum' and return 'sailors paying for rum'.

This simplification is necessary because the text-to-image model used in the experiment is the open-source diffusion model *Stable Diffusion XL (SDXL)* [71], specifically SDXL-1.0, which succeeds *Stable Diffusion (SD)* [72]. Both models are developed by Stability AI, and are open-source. SDXL does not always generate the desired output, therefore the simplification reduces the chance of generating unrelated content. Table 3.2 presents the complete list of few-shot examples used for the evaluation.

The image is displayed under the verbal cue with a width and height of 300 pixels. It was made small on purpose because using the visual cue was optional, and a larger image would be too distracting to ignore. Furthermore, we did not want participants to notice the details of visual cues, since their purpose is to give the participant an idea of what to imagine, not to admire the details. AI-generated images are generally non-suspect at first glance, but incoherent in the details, so we tried to prevent this from distracting the participants.

Since participants might prefer other verbal and visual cues than the proposed ones, two buttons are displayed at the bottom of the page, which lets participants re-generate either the visual cue or both the verbal and the visual cue.

For the same open-science reasons we have opted for open-source LLMs, we chose to use open-source text-to-image diffusion models. That leaves out popular but proprietary models like DALL-E [73], Imagen [74], and Midjourney [75]. This left us with the family of Stable Diffusion models as the best option. Due to SDXL's increase in model parameters and improved architecture, it is a direct upgrade over Stable Diffusion 1. x models [71].

We used the API endpoint provided by prodia.com to generate images with the available SDXL 1.0 base model, `sd_xl_base_1.0.safetensors [be9edd61]`. The

Verbal Cue	Text-to-Image Prompt
Imagine a fortune-teller with a pile of silver plates	fortune-teller with pile of silver plates
Imagine your kitchen and a cook in it	cook in kitchen
Imagine you rent meat to friends in your room	renting meat to friends in room
Imagine sailors pay for hot rum	sailors paying for rum
Imagine nail-clippers on a cliff	nail-clippers on a cliff
Imagine a flag on a fan	flag on a fan
Imagine you call a friend to put a new roof on a cottage	calling friend on top of a cottage roof
Imagine crabs dig holes in the sand	crab digging hole in sand
Imagine shears besides a pair of scissors	shears besides scissors
Imagine your lawn covered in raisins	lawn covered in raisins
Imagine you push stores in a cupboard	push stores in a cupboard
Imagine strikers paint slogans on walls	strikers painting slogans on wall

TABLE 3.2: Verbal Cues and Text-to-Image Prompts Used for Few-Shot Examples

classifier-free guidance scale hyperparameter was set to 10, which makes the model follow the prompt more closely than the default value of 7. Negative prompts were added to prevent the generation of inappropriate imagery for reasons that will be explained in Section 5.4.

3.9 Post-Evaluation

This section describes a part of the system design that did not make it into the system used for the human evaluation because it required data gathered from the user evaluation. This section is placed at the end of the current chapter rather than the discussion, because it relates to the topic of system design. However, it is recommended to read Chapter 4 and then return to this chapter so that no context is missing.

3.9.1 Automatic Optimization

We have limited ourselves to open-source models, which are often smaller and less powerful than proprietary models, such as those hosted by Anthropic and OpenAI. DSPy is particularly effective with small and open-source models due to its capability of *optimizing* language programs' performance [59]. The DSPy compiler simulates versions of the language programs and improves them based on a given metric.

This process allows DSPy to optimize performance without relying heavily on *"prompt-engineering"*, the manual wrangling of prompts to get desired results. This is more resource-intensive and less scalable. With DSPy compilers, automatically one can find better prompts, examples, and even finetune weights for local language models.

The generation of keywords in Keymagine can be compared to the process of information retrieval, where the input, a foreign word, serves as a query that prompts the system to return a list of keywords, analogous to a list of documents returned by a search engine. This allows for traditional information retrieval metrics, such as precision, recall, and F1-score to be applied as metrics to evaluate the relevance of

the keywords retrieved relative to the input query. By quantifying how well the retrieved keywords match the expectations set by the input query, we can objectively assess and improve the prompt and examples used in Keymagine.

Since keyword preferences are so subjective, we lacked a proper metric to be able to fully benefit from this DSPy feature. However, after the human evaluation was finished, we had a list of human-preferred keywords for every word pair in the word list. This list of keywords can be used as relevant keywords for automatic optimization. This section can be seen as a small experiment interlude.

3.9.1.1 Method

We retrieved this list by taking all the keywords used in the human evaluation that were either generated by Keymagine or the participant. Some keywords were removed manually as they were not likely to be similar to anyone but the participant who selected them, like the keyword "Ethiopia" for the German "Fahne". For each keyword, the number of times it was selected was counted. A sample of this data is presented in Table 3.3. The *keywords* are shown as tuples of (w, w_f), where w is a keyword and w_f is the number of times it has been selected by a participant.

No.	German word	Keywords
1	Sperre	(spear, 8), (pear, 1), (spare, 1), (sphere, 1)
2	Hose	(hose, 7), (house, 2), (horse, 1), (hoes, 1)
⋮	⋮	⋮
35	Sagen	(sage, 3), (saying, 2), (saigon, 1), (sagging, 1), (sacking, 1)
36	Reißen	(rice, 5), (raisin, 1), (rise, 1), (rising, 1), (rain, 1), (reisen, 1)

TABLE 3.3: A sample of the training data used for DSPy optimization.

For the metric used to optimize the keyword generation module, an improvised form of weighted precision was chosen. Precision measures how many of the generated keywords are relevant, i.e. previously chosen by participants. Weights are the keywords' frequencies in the training data, so that keywords that were chosen more times are more relevant. However, no formula could be found that works when an item can be in the retrieved items multiple times, so the sum of every generated w's w_f was used as a keyword's true weight, which was divided over the amount of keywords in the list. Metrics incorporating recall were not chosen since they measure what portion of total relevant keywords were generated, and it's not ideal to return all relevant keywords. Rather it would be better if a more relevant keyword is returned more times.

Four optimizers were tested on the model Mixtral-8x7B: LabeledFewShot, BootstrapFewShotWithRandomSearch, COPRO, and BootstrapFewShotWithOptuna. LabeledFewShot is the same teleprompter used for every module explained earlier. It simply takes k random samples from the set it's given. This one was initialized with $k = 16$. BootstrapFewShotWithRandomSearch performs a random search over few-shot examples to find an optimal set of examples to include in the prompt. BootstrapFewShotWithOptuna also tries to find an optimal set of examples, but uses the Optuna hyperparameter optimizer [76]. COPRO does not select or generate examples but generates and tests different variations of the initial signature.

Optimizer	Weighted Precision
No Optimization	108.16
LabeledFewShot	136.09
BootstrapFewShotWithRandomSearch	136.47
BootstrapFewShotWithOptuna	115.77
COPRO	108.16

TABLE 3.4: Results of the optimization experiment

3.9.1.2 Results

The results after optimizing Mixtral-8x7B with different optimizers at the default settings are displayed in Table 3.4. Compared to the LabeledFewShot optimizer with 4 examples, it can be seen that the other optimizers do not fare much better.

COPRO and the raw module without any optimization rank lowest. The instruction used by COPRO was the same as the initial prompt, meaning it was not able to find an instruction that performed better. These optimizations or lack thereof do not supply any few-shot examples when prompting the LLM, which shows the importance of demonstrations.

LabeledFewShot and BootstrapFewShotWithRandomSearch had about the same and highest score, with BootstrapFewShotWithOptuna below it. Inspecting the compiled modules revealed the latter method resulted in a compilation with only one example, while the former two both use four examples.

3.9.1.3 Discussion

The results show that the amount of examples given to the LLM is the primary factor influencing performance. Interestingly the carefully selected examples by BootstrapFewShotWithRandomSearch performed only slightly better than the randomly chosen examples by the LabeledFewShot optimizer.

A possible explanation for this observation is the nature of the outputs, which are limited to single words. As a result, individual examples can not greatly differ in the amount of guidance the LLM is given, the same way a well-thought-out example in a logical reasoning task can do.

Chapter 4

User Evaluation

To answer the third sub-research question: "How do personalized automated keyword mnemonics affect the learning outcome and experience compared to non-personalized automated keywords?", a user study was conducted using the web application described in the previous chapter. This chapter describes the methodology and results of this study.

The objective of this study is to assess the relative effectiveness of Keymagine to the current best solution to generate mnemonic keywords. Transphoner [26] was determined to be the best one. Released in 2014, it is still the most sophisticated solution developed for this purpose in 2024, as it can generate keywords for multiple languages and is fully open source. Other solutions only work for one language [40], or have been proposed but not published for public use [39]. In addition, a recent paper [35] builds on top of Transphoner, which affirms our choice. Transphoner represents the older algorithmic paradigm of NLP, whereas Keymagine is a newer LLM-based approach. It should be noted that this experiment does not measure the validity of the keyword method itself nor does it compare the effectiveness of using visual or verbal cues.

4.1 Methodology

4.1.1 Participants

The 22 participants in this experiment were all students at the Delft University of Technology, mostly consisting of male computer science students. This allowed for efficient participant recruitment within a reasonable timeframe and resource availability. While this sample may not fully represent the broader population interested in learning foreign languages, we believe that the sample size and composition are sufficient to assess the relative effectiveness of our system compared to Transphoner.

Participants were selected to ensure none had extensive experience in German, which was necessary to maintain the integrity of the learning assessment. Since some of the words they would encounter have cognates in the Dutch language, participants were also required to have no higher than A2 level proficiency in Dutch, as self-reported.

4.1.1.1 Ethics

All participants signed an HREC-approved informed consent form before partaking in the experiment, granting us permission to store their anonymized data, and granting them the ability to withdraw their data for up to 2 weeks after the experiment.

Data was stored on the author's personal laptop. The participants' contact information and anonymous user IDs were stored in an encrypted `xls` file on the author's personal laptop.

4.1.2 Materials

The experiments were conducted with one participant at a time in empty rooms on the TU Delft campus. No other people were present other than the author and the participant. The experiment was carried out on the author's laptop. Due to privacy regulations regarding storing participant data on servers, they could not perform the experiment on their own devices by logging into the web app online.

The pipeline described in Chapter 3 describes the workflow for one single pair of words. Around this workflow, the web application that the participants used in the experiment contains pages to log in, read the instructions for the experiment, navigate through the individual word pairs, and take tests. The procedure is described in the following sections, and screenshots can be found in Appendix A.

In the web application, participants learned a total of 36 German words. The word list is the same as that used in the experiment by Ellis and Beaton [28] and subsequently used in the papers of Savva, Chang, Manning, *et al.*, Lee and Lan [26], [35].

4.1.2.1 Variables

The independent variable was the source of the keyword mnemonics. Every participant learned exactly half of the words with the keywords generated by Keymagine or themselves, which we refer to as the Personalized condition. The other half was learned with Transphoner-generated keywords, which we call the Transphoner condition.

As for dependent variables, we sought to measure the learning outcome and learning experience, which are objective and subjective variables. Since they are not quantifiable, we operationalized the learning outcome by measuring cued recall of the English words given the German word. Participants were tested on all learned German words by being presented with the full list of German words and a text input next to each word to fill in the corresponding English word. This list was presented in the same order the words were presented in, to avoid any recency biases.

The learning experience was operationalized by having participants rate each keyword on a 5-point Likert scale, indicating how helpful it was in helping them recall the words. Only helpfulness is measured, because this provided us with many data points to compare the two conditions, without overloading the participants with questionnaires. Participants were furthermore asked through open-ended feedback to provide their thoughts on the process of selecting their preferred keyword.

4.1.3 Procedure

4.1.3.1 Research Design

The system was tested by means of a within-subjects experimental design. While learning the German words, each participant is presented with keywords generated by Keymagine for half of the words, and keywords generated by Transphoner [26] for the other half.

The allocation of the conditions was as follows: for half of the participants, Keymagine keywords were assigned to the odd-numbered words, and Transphoner keywords were assigned to the even-numbered words. For the other half of the participants, this assignment was reversed, with Keymagine keywords assigned to the even-numbered words and Transphoner keywords assigned to the odd-numbered words.

We have two hypotheses. The first null hypothesis is that the helpfulness ratings of keywords generated by Keymagine are not significantly different from those generated by Transphoner. The alternative hypothesis is that the helpfulness ratings of Keymagine-generated keywords are significantly higher than those of the control method.

It is expected that higher recall rates are observed with the LLM-generated and custom participant-made keywords. Therefore, the second null hypothesis is that there is no significant difference in recall rates between participants using LLM-generated or custom participant-made keywords and those using non-personalized keywords. The alternative hypothesis is that participants using LLM-generated or custom participant-made keywords will have significantly higher recall rates compared to those using non-personalized keywords.

4.1.3.2 Data Collection

At the start of the experiment, the participants received an introduction to the theory of the KWM and how to proceed through the experiment. They were given an example of the keyword method applied to the French word *poisson*, meaning fish, where the keyword was poison. A screenshot of the interface for a single word pair was shown using the French word. Fig. A.1 in Appendix A shows the full instruction page as the participants saw it. They were told to make use of the keyword to learn the foreign words, but they were allowed to ignore the verbal or visual cues if they did not want to use them.

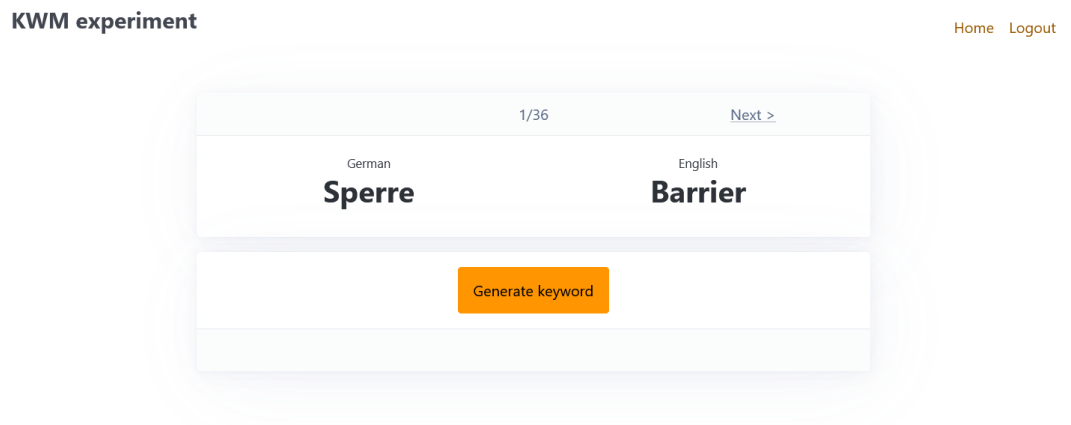


FIGURE 4.1: Screenshot of a "word pair" page at its start.

To learn a vocabulary pair in the Personalized condition, the user saw the foreign word and its translation side by side at the top of the page, as shown in Fig. 4.1. At the bottom of the page was a button that starts the generation process with a mouse click. The web page displayed a loading spinner and waited for the highest-ranked generated keyword. The user could choose to either use this keyword or not. If not, the system displayed a bullet selection list with the other keywords and an option

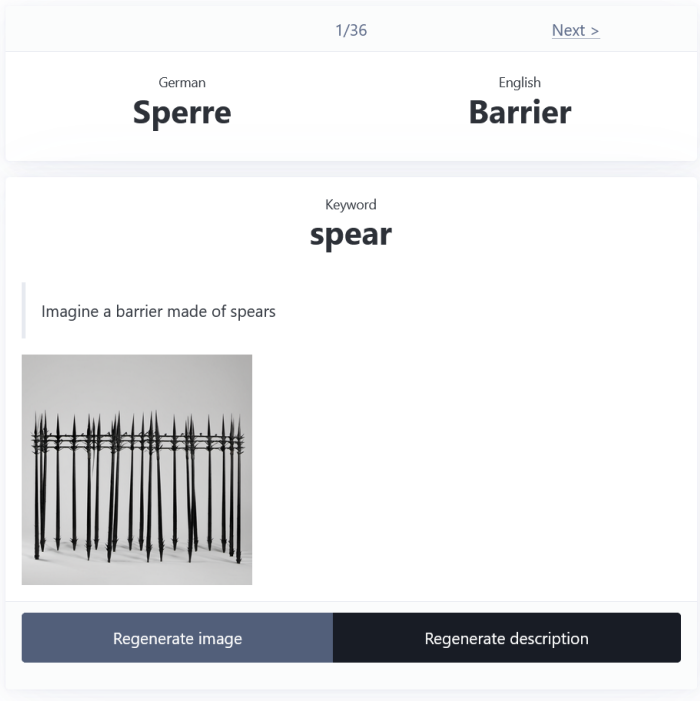


FIGURE 4.2: Screenshot of the final state of a "word pair" page.

to enter their own keyword. After the user selected their preferred keyword, the system generated and displayed a verbal and visual cue for the user. This can be seen in Fig. 4.2 In the Transphoner condition, the keywords and cues were pre-generated for each vocabulary pair. Therefore the system immediately displayed the keyword, verbal cue, and visual cue.

The participants studied the words in batches of 12 pairs. After each batch of 12, they were tested on their recall of the English words given the German words, to force them to practice the keyword method. The instruction they were given was *"Write the English translation of the following words. Use the keyword method to remember the words. This is just for reinforcing your memory. You will not get feedback. If you don't know a word, write "idk" or "-"."*

After completing the final batch of 12 pairs, participants took a five-minute break, during which they engaged in casual conversation with the experimenter or played the relaxing driving game *slowroads.io*. Following the break, participants completed the recall test. The final test can be seen in Fig. 4.3. Following the final test, they were redirected to the last web page where they were asked *"How helpful did you find the keywords? Please rate them from 1 (not helpful at all) to 5 (extremely helpful)."*. After rating each word, they were asked to answer two open questions. The first one attempts to gauge their opinion on the two conditions, without them knowing what the two conditions were: *"Did you notice any difference between when you could choose your keyword, and the ones where you couldn't? If so, what difference?"*. The second was open-ended feedback: *"Do you have any other thoughts, feedback or ideas? Anything goes."*

The keywords used in the Transphoner condition were directly sourced from the paper by Savva, Chang, Manning, *et al.* [26]. However, this paper did not include verbal and visual cues for each keyword. The supplemental material did include

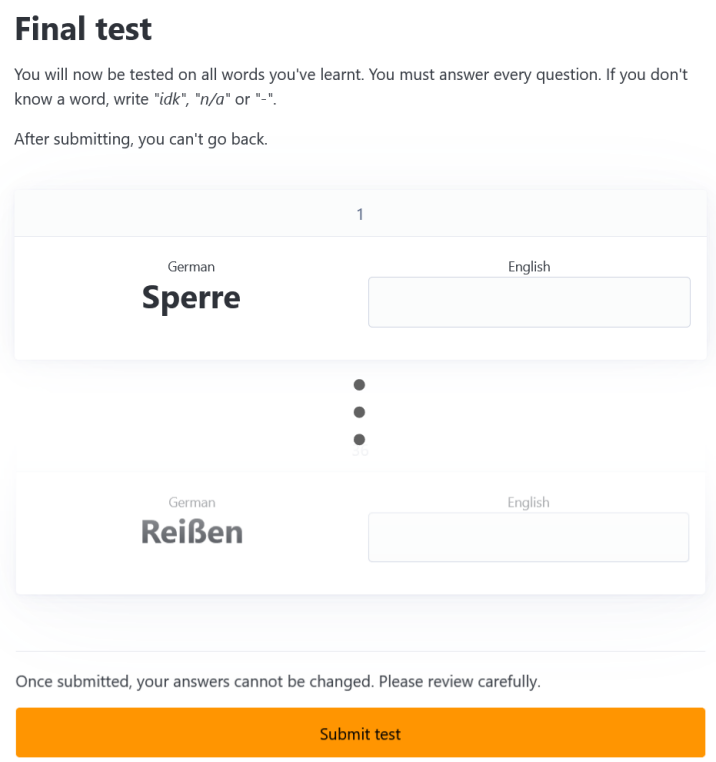


FIGURE 4.3: Screenshot of the final test page.

verbal cues, but this was If we were to present Transphoner-generated keywords to participants without verbal or visual cues, a confounding factor would be introduced to the experiment.

To prevent this, verbal and visual cues were generated for each word pair by the author. This was done by using the verbal cue module to generate a verbal cue. If the author determined that the generated scenario lacked coherence, it was regenerated up to three times. Visual cues were generated using the previously generated verbal cue. When the image did not feature both elements or when it was too disorganized to the point of being distracting, it was regenerated up to five times. These cues were saved and used in the experiments.

To ensure the quality of the data collection, participants were provided with concise instructions and procedures to follow. A quiet room was also made available for the participants to perform the experiment in. The author was also present in the same room to be available for questions. Since this can make the participant feel pressured, the author sat opposite the participant and did not unsolicitedly interfere with the experiment.

The answers were graded manually to account for typographical errors and synonyms. These were considered fully correct since they demonstrated that the participants had recalled the definition of the word. Both bare infinitive and present participle forms of the correct verb were also accepted as answers. In case a participant was unable to recall the English word or a synonym, they were permitted to provide a description. This was communicated to the participants at the outset of the study.

Given that English is not the native language of the majority of the participants, they were permitted to use the internet to look up the meaning of English words or

keywords during the learning process. Of course, they were not permitted to do so during any of the tests.

4.1.4 Analytic Strategy

To analyze the words, we performed a one-tailed t-test [77] to see if recall differs significantly between words learned with different methods. To analyze the helpfulness ratings, we performed the same test. The participants' open-ended comments were qualitatively analyzed in order to extract common opinions on Keymagine.

4.2 Results

In this section, we will go over the answers to all the subquestions we posed. We start with the first two and then move on to the third, which is answered by the results of the human evaluation study.

4.2.1 System Design Results

Our first research question was "How can large language models be used in keyword mnemonic generation." To this end, we designed the system described in Chapter 3 and found that LLMs don't require any finetuning to generate mnemonic keywords. Providing the LLM with a simple instruction and example inputs and outputs is sufficient to prompt the LLM for a keyword. Multiple completions can be generated at once to get a broader range of keyword candidates.

The second research question was "How can automatically generated keyword mnemonics be personalized to a learner?" This can be accomplished in at least three ways: knowledge tracing, where previous answers are used to model the learner's knowledge state, so that keywords and cues can be adapted to the person's current knowledge; content personalization, where generated material is customized to the learner's preferences and goals; and lastly, the implemented method, which is to incorporate human feedback into the learning process.

4.2.2 Recall Scores

The answer to the third sub-question "How do personalized automated keyword mnemonics affect the learning outcome compared to non-personalized automated keywords?" will be answered by the results gathered from the human evaluation. The human evaluation has two main results. The recall rates of the German words' meanings and the ratings the participants gave their used keywords.

Table 4.1 shows the graded results for each experimental condition. The correct column is for the fully correct answers. Incorrect answers are fully incorrect. Partial and Keyword are subsets of correct and incorrect answers respectively.

In the personalized category, the row labeled *User* represents word pairs where the participant provided their own keywords. The row labeled *Keymagine* corresponds to instances where participants selected a keyword generated by Keymagine.

The keyword column is a special reoccurring case of incorrect answers where the participant filled in the keyword instead of the English meaning when prompted. For example, one participant submitted *meet* as the translation of *mieten (to rent)*. The keyword used in this scenario was *meet*, and the verbal cue *"Imagine you rent a room to meet friends."* Another example for the word *Rasen (lawn)* were the answers

Category	Correct	Partial	Incorrect	Keyword	Total	Correct %
Transphoner	194	7	202	18	396	49.0
Personalized	276	7	120	14	396	69.7
Keymagine	221	5	95	9	316	69.9
User	42	0	16	3	60	72.4
Total	470	14	322	32	792	59.34

TABLE 4.1: Summary of Correct, Incorrect, Keyword, and Partial Grades for Different Categories

"rise" and *"lift"* when the keyword was *rise* and the verbal cue was *"Imagine your lawn has risen."*, or the answer *"to cover"* when the keyword and verbal cue were *"rasin"* and *"Imagine a lawn covered in raisins"*.

The partial category is counted as correct, even though they were not judged to be fully correct by the author. This category was assigned to answers that did not feature the exact English word, nor a synonym or form thereof, but still showed that the participant remembered the meaning of the word. One pair of words that had many partially correct answers submitted was *Dohle (jackdaw)*. The partially correct answers submitted were *"dove"*, *"jackhammer"*, *"dolly bird"*, *"jackwidow"* and *"jackblack bird"*. The reason jackhammer and jackwidow were counted as partially correct even though they are a type of tool and spider, is because the participants informed the author that they did not remember the name of the bird, which shows they did in fact know it was a type of bird.

Fig. 4.4 shows the mean percentage correct for each word in the word list, for both the Transphoner and personalized categories. The personalized category resulted in higher recall for most words.

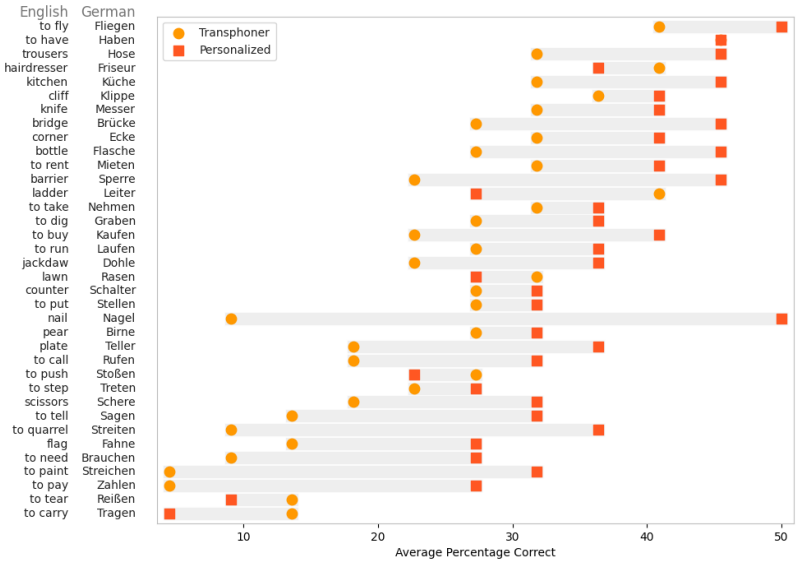


FIGURE 4.4: Comparison of mean percentage correct per word pair for Transphoner and Personalized categories.

We should note that user-generated and Keymagine-generated keywords should add up to the amount of personalized keywords, but it doesn't. The reason is that some participants did not generate a keyword for some words. One participant mentioned that he forgot to do it but used a self-generated keyword. For the other

participants who forgot to choose a keyword, it is not clear whether they used the keyword method or learned that pair without it. Hence, some results fall in the personalized category, but not in User or Keymagine. For the same reason, the percentage of fully correct answers in the personalized category is lower than the two subcategories.

Participants in the personalized category achieved a higher proportion of correct answers (M = 69.7%, SD = 46.0%) compared to those in the Transphoner-generated category (M = 49.0%, SD = 50.0%). A one-tailed t-test confirmed that the difference between the two means is statistically significant, $t(790) = 6.115$, $p < .001$. This suggests that the personalized condition has a significantly higher recall rate compared to Transphoner, strongly rejecting the null hypothesis of no difference.

4.2.3 Helpfulness Ratings

Category	Number of Ratings	Helpfulness	Standard Deviation
Transphoner	396	2.64	2.31
Personalized	374	3.49	1.96
User	58	3.81	1.84
Keymagine	316	3.42	1.95
1st suggestion	216	3.47	1.85
2nd suggestion	29	4.0	1.45
3rd suggestion	24	3.29	2.12
4th suggestion	26	2.69	2.44
5th suggestion	3	3.67	1.56
6th suggestion	8	3.25	1.44
7th suggestion	8	3.38	1.73
8th suggestion	2	2.0	1.0

TABLE 4.2: Helpfulness ratings for Transphoner-generated and personalized keywords

Table 4.2 shows the mean helpfulness ratings from the 1 to 5 Likert scale for the different methods keywords were generated by. The mean helpfulness was lowest for the Transphoner-generated keywords with 2.64. Following that are the keywords generated by Keymagine with a score of 3.42. Not unexpectedly, the keywords the participants thought of themselves were rated the highest on average, with a mean of 3.81. Fig. 4.5 shows the distribution of ratings given to keywords from both conditions. It is notable that keywords in the Transphoner condition received more than twice as many of the lowest rating.

Personalized keywords received higher helpfulness ratings (M = 3.49, SD = 1.956) compared to Transphoner-generated keywords (M = 2.64, SD = 2.312). The difference was statistically significant, $t(758) = 7.345$, $p < .001$, indicating a highly significant difference in perceived helpfulness between personalized and Transphoner-generated keywords. This suggests LLMs can effectively generate keywords for learners to use with the keyword method.

4.2.4 Participant feedback

The participants' responses to the question *"Did you notice any difference between when you could choose your keyword, and the ones where you couldn't? If so, what difference?"* were very similar. Only one participant reported not noticing a difference. Most

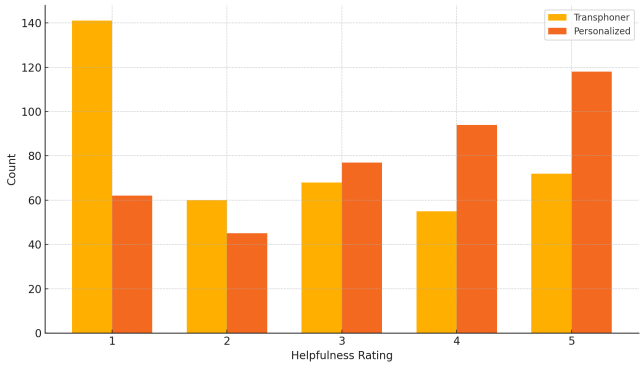


FIGURE 4.5: Distribution of Likert scale ratings for Transphoner and Personalized categories.

participants answered that they found it easier to remember words when they could choose their own keywords. Personalized keywords often create stronger and more memorable associations. Some examples are *"when i choose my own keywords, i could link the German word more to my own memory"*, *"The ones I chose I can relate to them better."*, *"I noticed I could remember the keyword faster and more accurately when choosing it myself."*, and *"I believe that when I choose my own words I memorize them better."*

Some participants found the Transphoner-generated keywords difficult due to them being unfamiliar or unintuitive. One participant said *"Sometimes I didn't know the word, so it didn't help me to memorize it."* This was especially the case for the keywords *"hora"* and *"kappa"*, as some participants had questions about it during the experiment. One participant noted that *"for the ones where I couldn't choose, sometimes I didn't understand the keyword (ho[r]a, kappa), which in the end didn't contribute to helping me memorize, since now I had to learn an additional word.""*

Chapter 5

Discussion

This chapter will describe some implications of the user evaluation's results. Then, the limitations of the study are described. Finally, some directions for future work are proposed.

5.1 Key Findings

We sought to evaluate the ability of LLMs to generate effective keyword mnemonics. We also looked at the impact of personalization on keyword generation. These variables were operationalized by measuring helpfulness and recall respectively, for two conditions: one where they could choose LLM-generated keywords, and one where they had to use keywords generated by Transphoner.

In the previous chapter, we saw that recall scores and helpfulness ratings greatly differed between Transphoner-generated and personalized conditions. The conducted t-tests suggested this difference was highly significant. We thus reject the null hypotheses, those being that the differences in helpfulness ratings and recall rates between the two conditions don't differ.

Upon reviewing the keywords generated by Transphoner, it was quite predictable they would not be ranked highly on average in terms of helpfulness. First of all, some of the keywords are quite obscure, such as the word *"hora"* which refers to an Eastern European and Israeli dance and was used as a keyword for the German Hose. Another example is "kappa," a Greek letter and mythological creature from Japanese folklore, which was the keyword used for *"Küche"*. It seems unlikely that the majority of native English speakers would be familiar with these words.

Secondly, the keyword *"sherry"* was assigned to two words in the list: *"Sperre" (barrier)* and *"Schere" (scissors)*. One of the three criteria for selecting keywords proposed y Raugh and Atkinson [11] was that each keyword used to learn a vocabulary list should be unique, in order to prevent learners from confusing two pairs of words. It is curious that Transphoner generates "sherry" for both words, despite the existence of superior keywords for *"Sperre"* in terms of orthographic and phonetic distance, such as *spear* or *spare*. It must therefore be the other factors that Transphoner takes into account, namely semantic distance and imageability, which made sherry the top choice. It can be concluded that excluding semantic distance as a factor in keyword generation with Keymagine was the right decision.

5.2 Keyword Generation

This section presents some findings regarding the keyword generation process that were observed during the study and after analyzing the results.

5.2.1 Translation Outputs

During the experiment, it was observed that in some instances the keyword generation module returned the translation of the German word instead of a keyword. This happened especially for the words *"Küche"* and *"Brücke"*. A suspicion arose that umlauts make it mess up for some reason. Tests with the words *"Mädchen"*, *"Hören"*, and *"Grüße"* confirm this suspicion. Feeding *"Mädchen"* into the keyword generation module resulted in four out of the ten completions being *"my daughter"*, while *"Grüße"* resulted in *"greetings"*.

Two ways were conceived to mitigate this problem. The first is to add another input field to the keyword generation module's inputs (`language, foreign_word`) so that we get (`language, foreign_word, translation`). Combined with an increased repetition penalty parameter, this will make the LLM less likely to repeat the translation in the similar_word output field.

The second way is to use DSPy Assertions [78]. This is a feature of DSPy whereby statements can be put into modules to check the output for a certain condition. When the condition fails, the module backtracks and tries again, this time inserting an instruction into the prompt to try and guide the LLM to the correct output. For example:

```
dspy.Suggest(not translation in output, "The output should not be a
↪  translation of the input word")
```

The program retries up to a maximum allowed amount of times. Upon reaching the limit, it either halts the execution or continues, depending on whether a hard or soft assertion is used. It should be noted that this solution to the problem is very dependent on the LLM chosen to run the program, since base models don't follow instructions as reliably as instruction-tuned models.

5.2.2 Output Variability

DSPy automatically caches LLM requests, so that identical inputs are not generated again. The cache was only flushed when the Docker image that the program ran on was restarted, which means there were groups of participants for whom the proposed keywords were identical. Because we observed the participants interacting with the system for the first word in the list, we could see the participants liked some of the Keymagine-proposed words, while hesitating with others.

The generation can thus differ between generations with the same model, but differ even more between different models. Table 5.1 shows the output of five-shot learning outputs between two LLMs of the same caliber: Llama3-70B and Mixtral-8x7B. The outputs are given in tuples (w, w_f) where w is a word, and w_f is the frequency with which it was generated.

Since the models were called with the same temperature parameter, which controls the randomness of the LLM's output, we can infer that different LLMs use different temperature scales. Therefore, temperature tuning should be conducted individually for each model, rather than being applied interchangeably across models. The results of the optimization experiment in Section 3.9 did not demonstrate significant improvements in addressing this variability. Therefore, ensuring the quality of generated outputs remains a big challenge.

	Meta-Llama-3-70B	Mixtral-8x7B-Instruct-v0.1
Friseur	(freezer, 4), (fries, 1), (fur, 1), (fry your, 1), (fresher, 1), (freeze, 1), (fertilizer, 1)	(fries, 6), (freeze, 1), (friesure, 1), (fresco, 1), (fresher, 1)
Pinda	(panda, 2), (pinata, 1), (pindal, 1), (pindar, 1), (peanuts, 1), (peanut, 1), (pinto, 1), (peter, 1), (pinda, 1)	(panda, 9), (pinterest, 1)

TABLE 5.1: Generated keywords for "Friseur" and "Pinda" by Meta-Llama-3-70B and Mixtral-8x7B-Instruct-v0.1 models.

5.3 Verbal Cue Generation

Some participants remarked that the verbal and visual cue generations did not align with the meaning of the keyword they had in mind. An example is the keyword *"fan"* for the German word *"Fahne"*, which means flag. The verbal and visual cue modules produced cues where fan was interpreted as a device for directing air currents, while the participant had in mind a fan of a football club waving a flag.

Not only does this issue apply to synonyms. Different interpretations of a single meaning are also possible. Take the keyword *"racing"*. This could refer to horse races, car races, etc. This is something that the user could have a strong preference for.

The problem of synonyms does not just apply to the keywords, but to L1 words in the Keymagine word list as well, such as *"rufen"*, which means to call, as in calling out or shouting. verbal cue generator interpreted 'call' as calling someone on the phone, however. This had no impact on our study, but if Keymagine is used to seriously study L2 vocabulary, it is crucial that the correct word is used.

Since verbal and visual cues could only be regenerated by pressing a button, participants had no option to correct the LLM to match their intended meaning. This forced them to either put mental effort into changing their mental preconceptions or ignore the generated cue. Users should therefore be able to give feedback to the verbal cue generation module regarding which synonym or form of the word they want to see.

5.4 Visual Cue generation

The visual cues turned out to be a major help to some participants. Two participants reported that the generated images helped them "a lot" in remembering the meaning of the words. Some participants also commented on how the strangeness of the images helped them remember: *"the images being funny helped me"*, *"For several examples, the description and image combination were so unintuitive that they ended up cementing themselves through their sheer absurdity"*. This corresponds with other research which has found that the more exorbitant the image, the more effective the mnemonic is [23].

5.4.1 Prompt Adherence

At the same time, there were criticisms of the generated visual cues. One participant commented that the images generated for their personalized keywords were "incoherent", while another found them "confusing". This is due to Stable Diffusion XL's limited ability to follow prompts accurately. While SDXL was chosen for its improved prompt adherence as written in Section 3.8, it is still lacking if you're seeking foolproof images.

For the generated image to aid the learner in using the keyword, it is obviously necessary for it to feature both the keyword and the meaning of the German word interacting as described in the verbal cue. However there were many instances during the experiments when SDXL would only include one of the two requested objects, or it would include both objects but not have them interact the same way the verbal cue described. This issue was especially prevalent when the English word and keyword were *"jackdaw" (German: Dole)* and *"dollar"*, or when they were *"trousers" (German: Hose)* and *"hose"*.

This resulted in some cases in participants regenerating images several times before realizing the images would not get better and they were better off creating an interactive image with their minds. Since each generation took between ten and twenty seconds, this is a lot of cumulative time wasted. The following paragraphs give some potential solutions.

Faster inference The benefit of faster inference is that less time is wasted generating a good image. Learners would be able to quickly iterate through generated images until they see a satisfactory result. Leaving out the option of acquiring faster hardware, leaves us with reducing computational cost. A simple way to do this is to generate images at a lower resolution. This would work for earlier Stable Diffusion models, but SDXL, which generates images in 1024×1024 resolution, is conditioned on image sizes. As a result, lower resolution images come out much more deformed [71]. Another way is to reduce the number of inference steps. This is a direct trade-off between speed and quality, so it should be tuned manually.

Many models have come out that make it possible to generate high-quality images in as few as one step, like SDXL Turbo [79], SDXL Lightning [80], and Hyper-SDXL [81]. These models allow generation speeds of fractions of a second, which would allow learners to quickly regenerate and select their preferred image. Since they are based on SDXL, they do not have inherently better quality or prompt adherence.

Better prompt adherence For the best prompt adherence, DALL-E 3 is one of the most accessible and state-of-the-art text-to-image models. Fig. 5.2 and Fig. 5.1 show how much DALL-E 3's generations stand out from the rest. They clearly feature both elements of the prompt, showing both of them prominently with the right interaction between them as the prompt specifies.

An upcoming family of open-source text-to-image models is the Pixart family of models [82]. Pixart-Sigma uses 0.6 billion parameters, while SDXL uses 2.6 billion. Despite the small size, it has superior prompt adherence due in part to its text encoder and training method. The study also demonstrates that Pixart-Sigma rivals proprietary models in quality. As can be seen in Fig. 5.2 though, jackdaws did not feature in its training data. Fortunately, this is not a limitation of the model, but of the small cost spent training. Stable Diffusion 3 [83] is currently the latest model in

(A) SDXL (B) Hyper-SDXL (C) SD 3 (D) DALL-E 3 (E) Pixart-Σ

FIGURE 5.1: Images generated with different image diffusion models given the prompt "Garden hose and trousers"

(A) SDXL (B) Hyper-SDXL (C) SD 3 (D) DALL-E 3 (E) Pixart-Σ

FIGURE 5.2: Images generated with different image diffusion models given the prompt "Jackdaw holding a dollar in its beak"

the family and demonstrates better prompt adherence than SDXL. With these fast developments, it is expected that performance will improve in the future when newer models based on the same open-source training method are released.

5.4.2 Open-Source Models & Safety

It is important to devote some time to the safety aspect of image generation. Since SDXL is an open-source model, absolutely no restrictions are placed on what one may generate with it. This gives a great advantage in our case as we're therefore able to generate whatever the learner prefers. For example, the German word *"Birne" (English: pear)* made one participant think of *Bernie Sanders*, so he was given the visual cue shown in Fig. 5.3. Another keyword chosen by some participants for the word *"Messer" (English: knife)* was Messi. It was therefore fortunate for them that the visual cues could be generated without a problem. DALL-E 3 however cannot or will not generate these images, as it "violates the content policy."

This freedom is also a double-edged sword. In the pilot study, an incident occurred at the 26th pair. The German word was *"Flasche"*, meaning bottle. The test subject chose *flash* as the keyword. Unlike many previous test runs, Keymagine didn't generate a verbal cue featuring a lightning flash or flash of light in a bottle, but something along the lines of imagining flashing a champagne bottle. This tripped the visual cue module up a lot, since it understood flashing in this context to be the act of exposing one's intimate parts, and presented an image of a woman doing exactly that to the test subject, to his complete shock.

In further experiments, this issue was prevented by adding a negative prompt `"((NSFW)), sexual content, cleavage, explicit, lewd, skin"` to every generation, which serves as an 'opposing' prompt, telling SDXL what not to put in the image. After implementing this change, it consistently avoided generating any not-safe-for-work (NSFW) or inappropriate content.

It is crucial that such images are not presented to learners, especially when considering their potential use by underage language learners. Not only would that

FIGURE 5.3: Visual cue generated for the verbal cue *"Imagine bernie sanders eats a pear."*

FIGURE 5.4: Visual cue generated for the verbal cue *"Imagine Messi with a knife."*

be highly unethical, but exposing minors to that type of inappropriate material or neglecting to prevent such exposure is also illegal.

5.5 Limitations

The evaluation of Keymagine has some limitations that must be acknowledged. Firstly, the sampling strategy employed resulted in a participant pool that mainly consisted of young adult males with higher education backgrounds. The participants in this study were primarily engaged in learning programming languages such as Rust and C++, rather than natural languages. This demographic bias is different from this study's initial motivation in Chapter 1, which focused on high school students and frequent language learners.

The fact that English was not the native language of any participant played a bigger role than anticipated. The previous studies used participants from anglophone countries [26], [28], so they undoubtedly had no issues with the English meaning of the words. In contrast, the participants in this study were international university students from all over the world, all studying in English but with varying levels of fluency. Some English words from the word list like *"to quarrel"* and *"flagon"* were unfamiliar to them. This was mitigated by allowing participants to look up the meaning of unfamiliar words.

The order of words during the final test was kept in the same order as they were shown during the learning process. The idea behind this was that it would remove recency bias as a factor during the recall test, but in hindsight, randomizing the words' order would have had the same effect without introducing new complications. One participant noted that the chunks of 12 being in the same order helped him remember the words, but they tried to discard this and only focus on the keyword.

We also noticed that the keyword for the fifth word in the list, *"Ecke"* was *"echo"*, for all participants except one. This keyword is not obvious enough for the LLM to generate it ten out of ten times. The reason it did was because it was left in the few-shot examples by accident. We also found *"Brücke"* and *"Rufen"* were used in the examples given to the keyword generation module. Since one should not mix

the training and test set, this was a big oversight, and might have given Keymagine an unfair edge.

5.6 Future Work

For future work, a few directions can be given. To begin with, further evaluations of Keymagine should at least build upon the solutions given to the issues noticed by the participants in their experiments, like the safety of generated words, the output of translations instead of keywords, and handling synonyms of words.

Since some participants mentioned how they would like to use keywords in their native language which is not English, extending the scope of Keymagine to include non-English keywords. This can provide insights and issues into different preferences that people with different native languages might have with automatically generated keywords.

This system should also be evaluated with broader demographics. The study that was conducted with adult university students was informative and had good results, but they don't accurately represent the spectrum of language learners. Since I emphasize in Chapter 1 how much vocabulary students in European high schools have to study, future research can look at how those pupils at different levels of learning benefit from Keymagine. This could help determine how well younger students with different cognitive abilities can use the system and benefit from it. Such studies could make the system more relevant for educational purposes at a broader scale.

Finally, it is worthwhile to research deeper personalization options for the keywords and cues. Knowledge tracing in the form of tracking learners' studied words and basing keywords on their knowledge state has the potential to help them a lot. For instance, if a learner is studying a compound word and already knows one component, they would only need a keyword for the part they are unfamiliar with.

Content personalization is also interesting to explore. Generating cues related to a learner's interests may positively influence their motivation and engagement. Personalizing keywords and cues based on cultural differences is also worthwhile. For example, the concept of a 'kitchen' might evoke different images in different cultures, which could influence the effectiveness of a mnemonic. Similarly, symbols for the concept of 'religion' differ across cultures, so a visual cue of a church likely doesn't resonate with Muslims or Buddhists.

Chapter 6

Conclusion

This thesis sought to find out how generative AI can be used to generate personalized keyword mnemonics for learning vocabulary. The three sub-questions posed were:

1. How can large language models be used in keyword mnemonic generation?
2. How can automatically generated keyword mnemonics be personalized to a learner?
3. How do personalized automated keyword mnemonics affect the learning outcome and experience compared to non-personalized automated keywords?

The first two sub-questions were answered by building an LLM-powered pipeline and interface named Keymagine. The first sub-question is answered by having built the keyword generation modules, which use in-context learning with multiple completions and imageability ratings to generate and rank keywords.

The third sub-question was addressed through human evaluation, which demonstrated that personalized mnemonics yield higher helpfulness ratings and improved recall rates compared to their non-personalized counterparts. This suggests that incorporating user feedback into the keyword generation process both enhances the learner engagement and learning outcome.

Keymagine shows how generative AI can be used in a simple way to generate personalized keyword mnemonics and outperform the previous state-of-the-art solution objectively and subjectively, which promises great gains for more sophisticated future research in the area of LLMs and mnemonic learning.

Future research should test an updated iteration of Keymagine with improvements based on the user study. The research could extend the application across multiple languages and a more diverse and representative group of language learners. More research should also go into improving the reliability of the keyword generation and deeper personalization of the keywords, verbal cues, and visual cues in the areas of knowledge tracing and content personalization.

A conceptual framework of the thesis is presented in Fig. 6.1

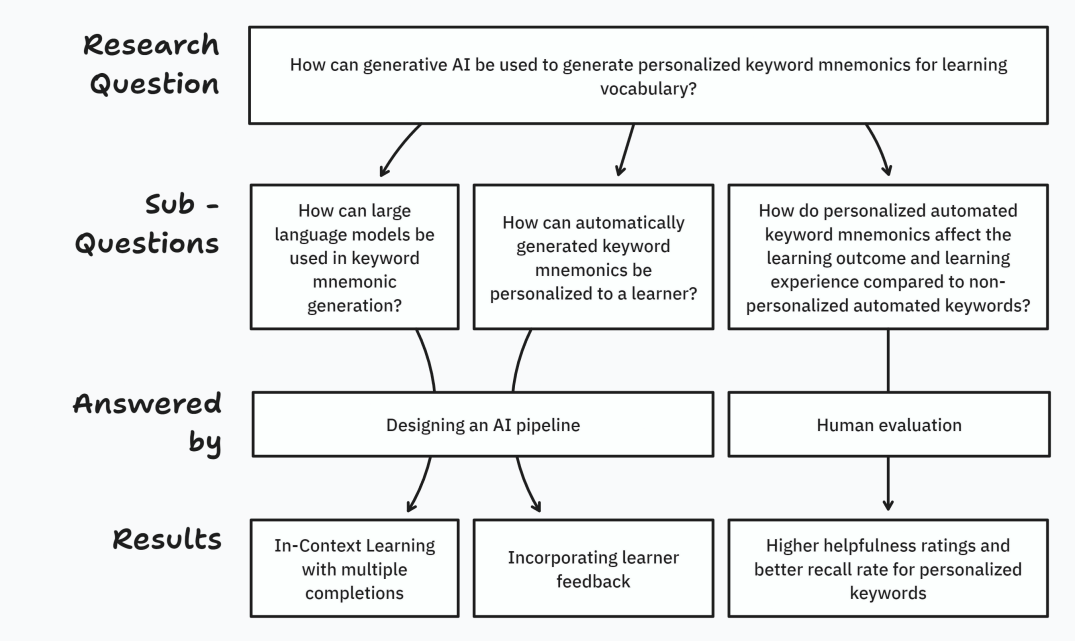


FIGURE 6.1: Conceptual Framework of this Thesis

Appendix A

Interface

This Appendix shows the key screens that participants went through as they did the experiment.

Instructions

Read this before starting. You can always return here later.

The Keyword Method

In this experiment, you will be using the *keyword method* to learn words in a foreign language. The keyword method is a technique used for learning new words by associating them with a *keyword* in your native language.

As an example, consider the following pair of words:

- French: *poisson*
- English: *fish*

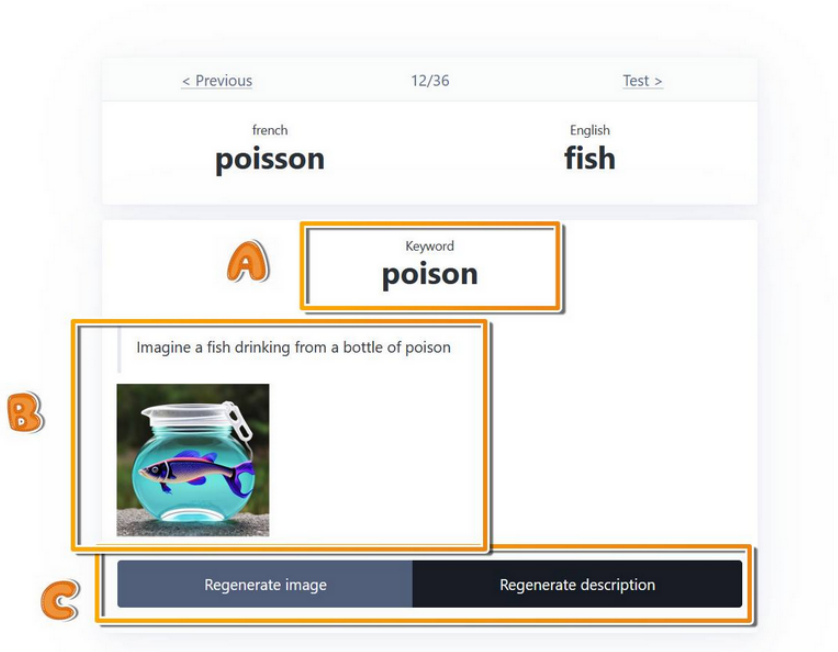
The French word *poisson* looks a lot like the English word *poison*. You can link the two words together with a vivid mental image, for example, by imagining a fish drinking a bottle of poison.

Next time you see the word *poisson*, you will remember the keyword *poison*, which reminds you of the image, and thus the translation *fish*.

This Experiment

You will be presented with 36 German words and their translations, one at a time. For each pair of words, a keyword is generated. For some of the pairs, if you don't think the association between the German word and the keyword is strong enough, you can choose from other options, or input your own. For the others, you should use the given keyword.

After the keyword has been set, the interface will look like this. You can see the keyword (A), a description and picture (B), and the option to get a new description or picture (C).



It is important for you to close your eyes for a few seconds, and try to strongly imagine a visual scene connecting the given keyword with the English meaning. The description and pictures are there to help you, but you may ignore them and create your own mental image.

After every set of 12 words, you will have a small test, to reinforce the previously learnt words.

At the end, you'll have a short break and then a final test.

If you have any questions, please ask Safouane.

When you're ready, click here to start.

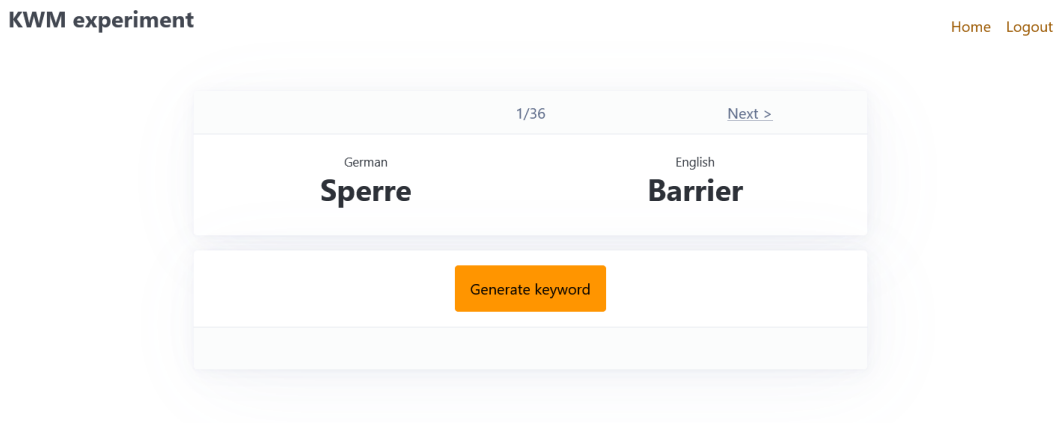


FIGURE A.2: Screenshot of a "word pair" page at its start.

FIGURE A.3: Screenshot of a "word pair" page after a keyword has been proposed.

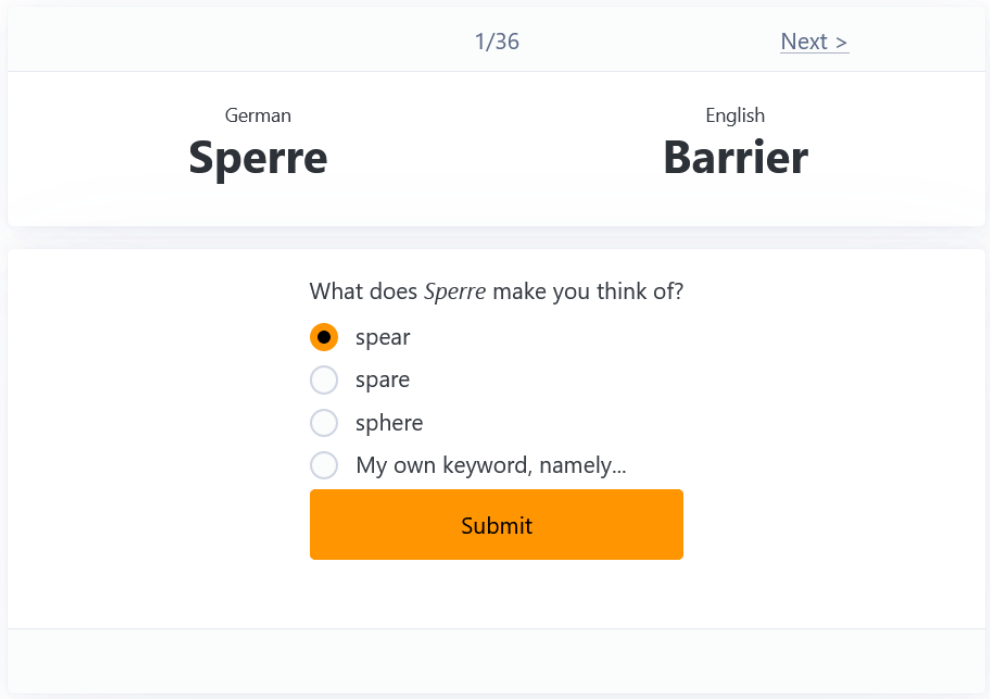


FIGURE A.4: Screenshot of a "word pair" page after disagreeing to a proposed keyword.

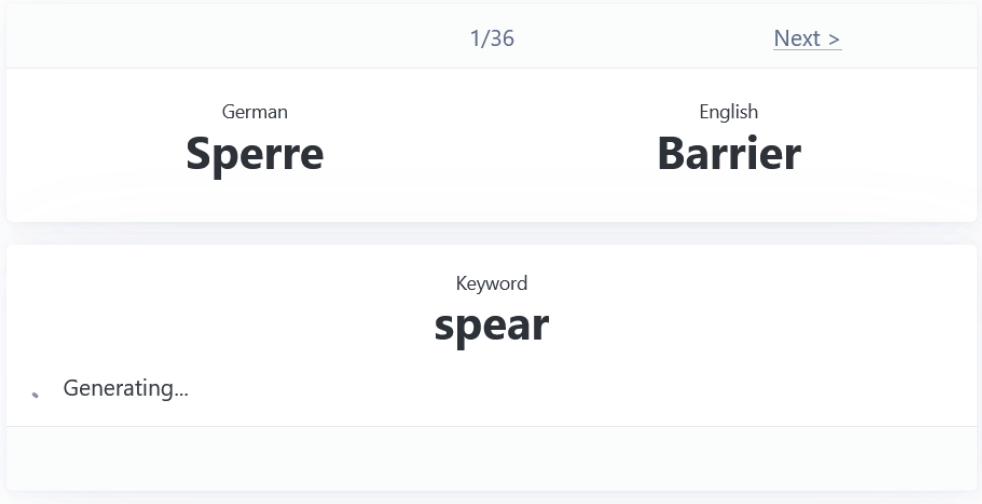


FIGURE A.5: Screenshot of a "word pair" page after choosing a keyword.

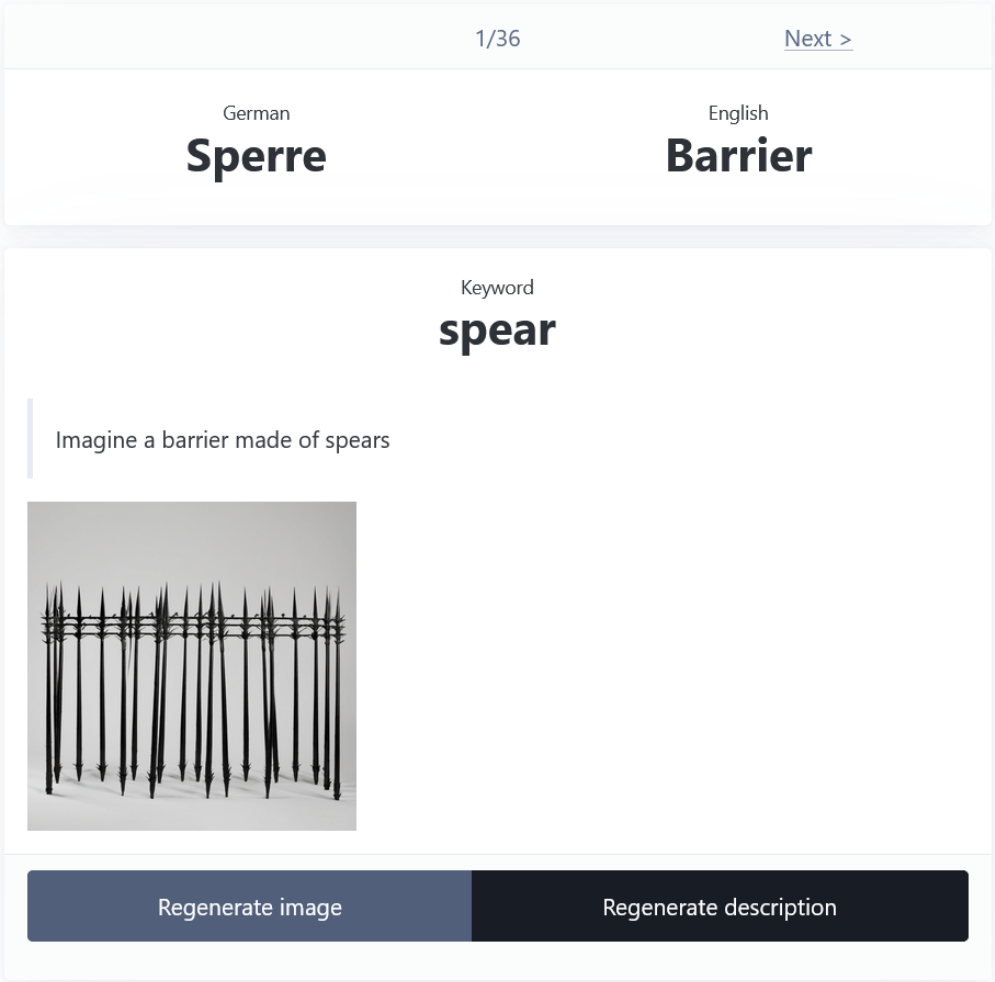


FIGURE A.6: Screenshot of the final state of a "word pair" page.

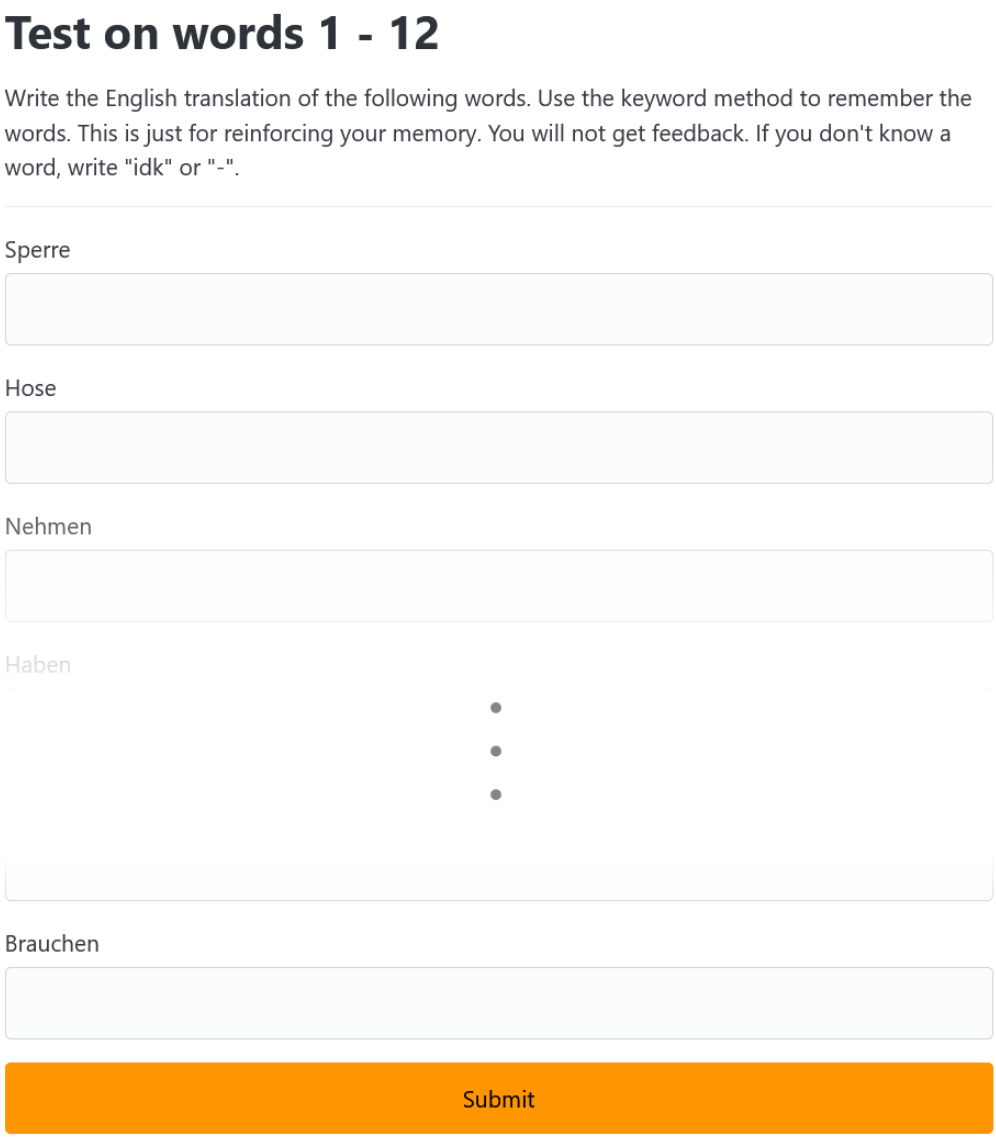


FIGURE A.7: Screenshot of the intermediary test page.

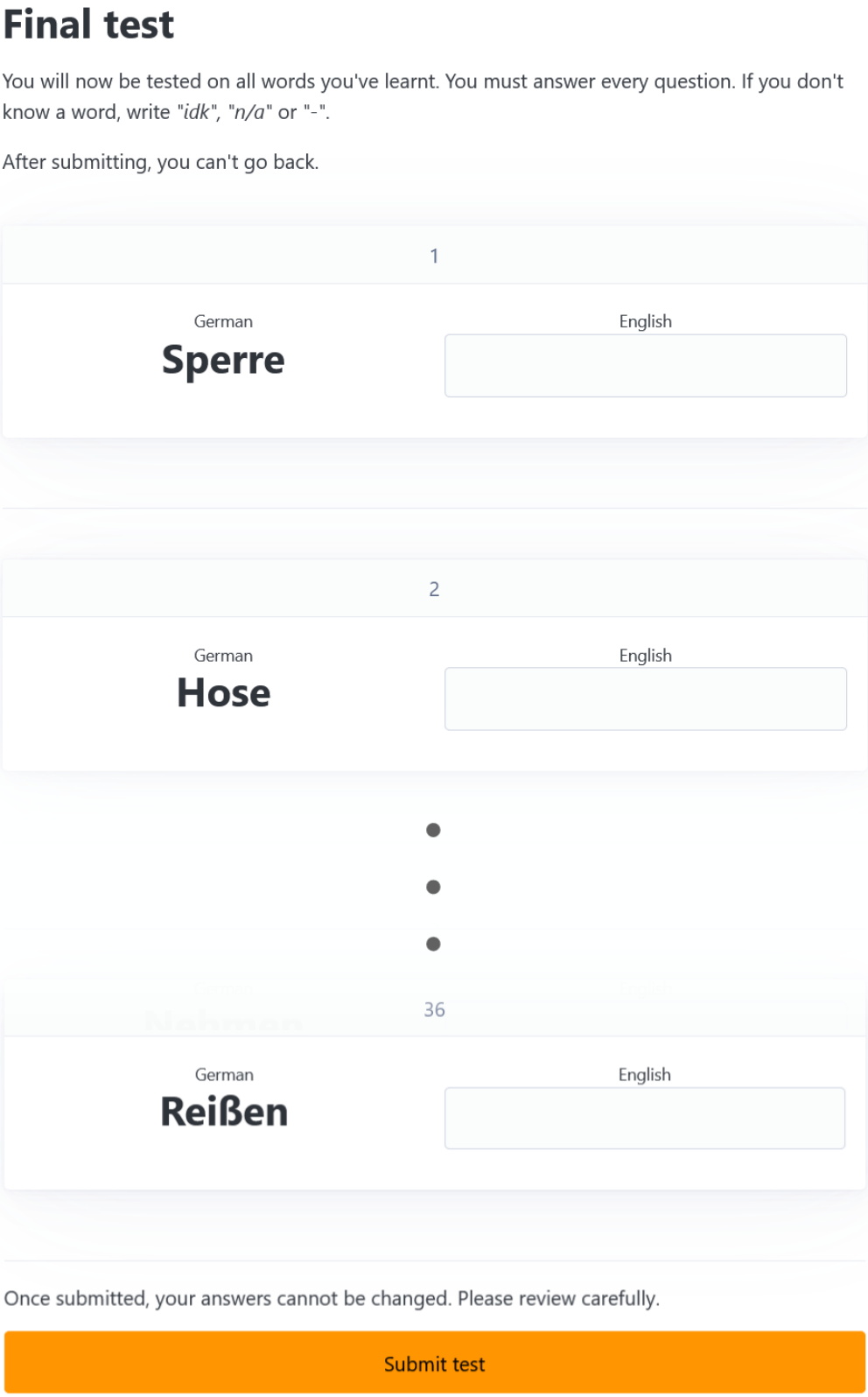


FIGURE A.8: Screenshot of the final test page.

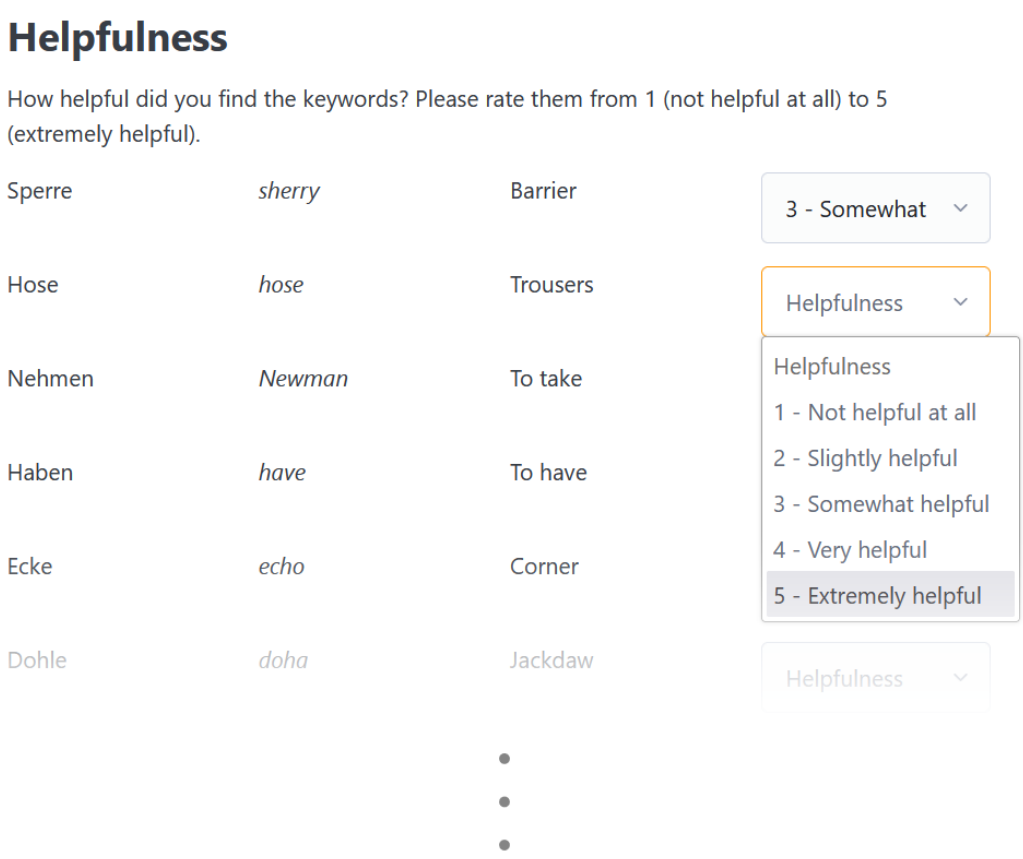
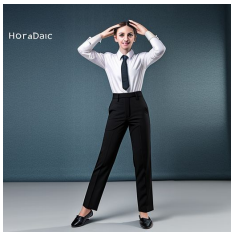


FIGURE A.9: Screenshot of the review page.

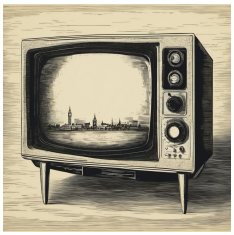
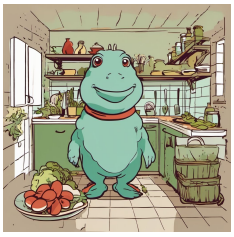
Appendix B

Transphoner Keywords

This appendix shows the keywords, verbal cues, and visual cues used for the Transphoner condition.

No.	English	German	Keyword	Verbal Cue	Visual Cue
1	barrier	Sperre	sherry	Imagine a barrier of glasses of sherry	
2	trousers	Hose	hora	Imagine a hora dance with trousers	
3	to take	Nehmen	Newman	Imagine taking Newman to the cinema	
4	to have	Haben	heaven	Imagine having heaven in your backyard	
5	corner	Ecke	echo	Imagine an echo in a corner	

No.	English	German	Keyword	Verbal Cue	Visual Cue
6	jackdaw	Dohle	dole	Imagine a jackdaw selling dole	
7	to buy	Kaufen	colon	Imagine a colon in a shop	
8	to fly	Fliegen	flagon	Imagine a flagon flying through the air	
9	ladder	Leiter	lighter	Imagine a lighter on a ladder	
10	hairdresser	Friseur	frizzy	Imagine a hairdresser with frizzy hair	
11	to put	Stellen	stellar	Imagine putting stars in a jar	
12	to need	Brauchen	broken	Imagine you need a broken plate	

No.	English	German	Keyword	Verbal Cue	Visual Cue
13	plate	Teller	telly	Imagine a plate on your telly	
14	kitchen	Küche	kappa	Imagine a kitchen with a kappa	
15	to rent	Mieten	meter	Imagine your meter is rented	
16	to pay	Zahlen	fallen	Imagine a fallen tree that you pay to have removed	
17	cliff	Klippe	clipper	Imagine a clipper on a cliff	
18	flag	Fahne	fauna	Imagine a flag for fauna	
19	to call	Rufen	Reuben	Imagine you call Reuben on a phone	

No.	English	German	Keyword	Verbal Cue	Visual Cue
20	to dig	Graben	grabber	Imagine you dig grabbers in the garden	
21	scissors	Schere	sherry	Imagine a bottle of sherry on a shelf with scissors	
22	lawn	Rasen	risen	Imagine your lawn has risen	
23	to push	Stoßen	stolen	Imagine you push stolen goods in a bag	
24	to paint	Streichen	stricken	Imagine stricken houses painted blue	
25	counter	Schalter	shelter	Imagine a counter as a shelter in a store	
26	bottle	Flasche	flashy	Imagine flashy bottles on a shelf	

No.	English	German	Keyword	Verbal Cue	Visual Cue
27	to quarrel	Streiten	triton	Imagine tritons quarrel in the sea	
28	to run	Laufen	loafer	Imagine loafers run in a race	
29	bridge	Brücke	bracken	Imagine a bridge in bracken	
30	knife	Messer	messy	Imagine a messy knife	
31	to step	Treten	treason	Imagine you step in treason	
32	to carry	Tragen	taken	Imagine you take a picture when you carry a camera	
33	nail	Nagel	novel	Imagine a nail in a novel	

No.	English	German	Keyword	Verbal Cue	Visual Cue
34	pear	Birne	bin	Imagine a bin full of pears	
35	to tell	Sagen	wagon	Imagine a wagon that tells you stories	
36	to tear	Reißen	ripen	Imagine ripe fruit that tears when you touch it	

Appendix C

Thesis Proposal

1 Introduction and Problem

Learning vocabulary is key to foreign language acquisition and something that children worldwide have to do. Mnemonics are a powerful tool for memorizing vocabulary more easily [1]. Keyword mnemonics are one type of mnemonics where a keyword is found that sounds like the unfamiliar word (e.g., “rufen” (to call) sounds like “roofing”). Those two are connected with a memorable imaginary scene (imagine you **call** your friend to help install a new *roof*). Seeing “rufen” will remind you of “roofing”, which will remind you of *call*.
This method has been time-tested, but is not widely used. In one study, students report that using the keyword method is too time-consuming: “I cannot think of a mental image or mnemonics for the target vocabulary. I’m bad at making them by myself. Also, imagery or mnemonics are not suitable for all the words. I’d rather spend my time on writing or vocalizing the target words.” [2]
There are few papers on automatically generating keywords, albeit of high quality systems [3]–[5], so there is a lot of room for innovation. Previous solutions on automatic keyword mnemonic generation use deterministic ways of assigning keywords to vocabulary. My approach uses LLMs to provide the learner with more creative, context-sensitive and imaginable information, by giving them keywords that are closer to them.

2 Thesis Objective

The objective of this thesis is to provide learners of (European) languages with the best possible keyword mnemonics to accelerate their retention of foreign vocabulary. This system has two components.
The first component is the generation of the best keyword. Several factors should be taken into account to give each learner the best keyword:

- *Phonetic similarity* is the similarity between the sound of the keyword and the foreign word to learn. Let’s take the example of DE:Küche (EN:Kitchen) and the keyword cook. Upon hearing Küche, the learner will be reminded of a *cook* in their *kitchen*.
- *Orthographic similarity* is the similarity in spelling between the foreign word and the mnemonic keyword. It could be the case that the foreign word and keyword do not sound alike in the slightest, but look familiar on paper. Take the foreign word FR:Raconter (rah-kon-tay, EN:to tell) and the keyword raccoon. Even though they don’t sound alike, it’s a suitable keyword due to its orthographic similarity.
- *Semantic similarity* is the similarity in meaning between the keyword and foreign word. Some studies [3], [6] take this into account when generating keywords, as it would facilitate “the forming of associations between foreign words and their native language translations” [6]. Since no sources were given to substantiate inclusion of this factor, it may not be relevant.
- *Imageability*: This is defined as the ease with which the word can be imagined, which is important in order to benefit the most from the keyword mnemonic technique. The imageability of a word is highly correlated with the average Age of Acquisition, for which datasets are available [7].
- *Etymology*: It is not necessary to generate elaborate mnemonics and keywords if the foreign and native words are already orthographically or phonetically similar due to being etymologically derived from each other or from a third word. E.g. EN:table and NL:tafel.
- *Previously known vocabulary*: Connected to etymology is the possibility of referring the learner to words in their native language(s). For example, if the learner wants to learn that DE:Kommode means dresser, and they are fluent in Spanish as well, then it will be enough to remind them that dresser is cómoda in Spanish. The system can also take into account vocabulary the learner has stored in the system already. For example, if the learner has already learnt beurre, they only require a mnemonic for arrachide when they want to learn FR:beurre d’arachide (peanut butter).

The second step of the keyword mnemonic technique is to encode a meaningful interaction between the keyword and the definition of the foreign word [8]. If we take the last example of Kommode and take “Komodo dragon” as the keyword, we might associate the two by imagining opening a dresser and finding a Komodo dragon sleeping inside. Thanks to new generative AI tech, automated generation of this second step is opened up for us.

- *Verbal cues* can be automatically generated by large language models, given the keyword and the foreign word's definition. For example, given "flashy" and "bottle" (DE:Flasche), output "Imagine a flashy bottle that stands out from the rest" [4].
- *Visual cues* can be automatically generated by feeding the verbal cue to text-to-image models, giving a reference point for learners with weaker visualization skill.
- *Personal interests*: During setup, the learner can input their hobbies and (pop-)cultural interests, e.g. sports and favorite film and book franchises, so that the verbal cues can incorporate scenes and characters that are close to the learner's heart.

This might be too big of a scope, so the focus can go to both or either of these two components.

2.1 Research Questions

The main research question that this thesis project tries to answer is:
How can generative AI be used to generate creative and personalized keyword mnemonics for learning vocabulary of European languages
To answer this question the following sub questions will be answered:

1. Can an LLM be fine-tuned for the task of keyword mnemonic generation?
2. How can personal interests and known vocabulary be encoded in LLMs in order to include them with each prompt?
3. What new types of cues can be made at scale using generative AI?

The first question is aimed at finding out whether an LLM can take into account all the factors mentioned that go into generating a keyword (phonetic similarity, imageability, etc.). Previous solutions [3], [5], [6] use deterministic solutions for this. What I want to find out with this subquestion is whether an LLM can take care of the whole process in order to generate keywords that are just as relevant as those in the previously mentioned papers, through e.g. chain-of-thought reasoning. [9]
The second question is about how the LLM can take the learner's personal interests and previously learned words into account. Given the persons's media interests, how do we create a pool of data that the LLM can take knowledge about the media from? This consists of (1) Getting the information from a knowledge bank like Wikipedia, Wikidata etc. and (2) storing the data in a form the LLM understands. Training an LLM for each person's interest would be too computationally expensive. Giving the entirety of the text with each prompt is expensive as well.
The third question tries to go beyond what has been done and try to find new ways of encoding keywords apart from imageable verbal cues and visual cues. Perhaps rhymes to remember sequences of words, or other memory systems.

2.2 Contributions, Engineering and Education

Below are the contributions that this thesis project hopes to have accomplished when it is finished. Programmatic contributions:

1. A system where a learner can input relevant personal information and lists of vocabulary to learn, for which keyword mnemonic cues are generated.

This study has in mind the middle- and high-school population of the Netherlands, who all have to learn lists of vocabulary in English, French, and German among other languages. Many people use spaced-repetition systems, which I believe can be enhanced with this programmatic contribution.
Research contributions:

1. Replace algorithmic ways of keyword generation with a single LLM-based method.
2. Find out how to incorporate data (interests, other vocab) that would be too large to include in the prompt.

3 State of the Art

The most cited paper that introduces automatic generation of mnemonic keywords is „Transphoner: Automated mnemonic keyword generation" [3]. Using an algorithmic approach, it looks up the input word in dictionaries to get its pronunciation, meaning, and other attributes like imageability. It searches for candidate keywords

in the target language and uses an optimization algorithm to find the best matching keyword sequence that maximizes phonetic, semantic, imageability, and orthographic similarity.
Jemsoundex [6] is a system for automatically generating keywords for English-Japanese vocabulary pairs. Its focus is on phonetic similarity, and it has an elaborate algorithm for finding similar-sounding words for mnemonic keywords, based on a modified Soundex algorithm [10].
A study by Anonthanasap, He, Takashima u. a. provides an interactive interface where learners can browse foreign language words and see phonetically similar keywords [11]. It only incorporates phonetic similarity to suggest keywords so it does not bring much innovation to the table there, but it showed that participants benefited from suggestion of images more than static visualization.
SmartPhone [4] is the most recent study and the only one to use LLMs. It uses them to generate verbal and visual cues for automatically generated keywords by Transphoner. In the user study where automatically generated verbal cues were compared with manually created ones, they found that automatically generated verbal cues alone did not improve learning over just the keyword, and visual cues were perceived as helpful but performance was mixed.
They give important directions for future research though, which I have incorporated in the thesis objectives: automatically generating keywords rather than using transphoner-generated keywords, personalized cues, and using other features of the word to generate cues other than pronunciation.

4 Method

User evaluation seems to be the only option to evaluate such a system, but working with high schoolers is a pain regarding HREC.
I'm planning to communicate with one of the language courses given at TU Delft (Italian, Spanish and Dutch) to set up an experiment in which the students use a Spaced-Repetition System (SRS) with inbuilt automatic keyword generation. This SRS will either be made by me as a basic web app, or be taken from one of the open source solutions (e.g. Anki [12]), with the automatic keyword generation as a plugin.
Another option is to release this plugin or SRS online and collect learning analytics, if the user opts-in. This would have lower quality research data, but has the potential to get data on many more people than in a classroom setting.
Anonthanasap, Ketna und Leelanupab evaluate their generated keywords from an information retrieval perspective [6]. Generated keywords were judged by Japanese language teachers for relevance, and their algorithm was then scored on common information retrieval metrics like precision, recall and NDCG. According to Dr. Derek Lomas this is a good way to go about it (private correspondence): "I think getting expert review and human learning data will be key. It won't be that hard to do."

5 Milestones and Expected Results

below are the milestones and deliverables for the project. At each date is specified what would be required to be presented.

1. Literature review (Nov-December)
   Deliverable: Literature review chapter of the report.
2. Keyword generation portion (January)
3. Cue generation portion (February)
4. User study? (March-April)
5. Final report and presentation

References

[1] A. L. Putnam, „Mnemonics in education: Current research and applications.," *Translational Issues in Psychological Science*, Jg. 1, Nr. 2, S. 130, 2015.

[2] A. Mizumoto und O. Takeuchi, „Examining the effectiveness of explicit instruction of vocabulary learning strategies with Japanese EFL university students," *Language Teaching Research*, Jg. 13, Nr. 4, S. 425–449, 2009.

[3] M. Savva, A. X. Chang, C. D. Manning und P. Hanrahan, „Transphoner: Automated mnemonic keyword generation,“ in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2014, S. 3725–3734.

[4] J. Lee und A. Lan, „SmartPhone: Exploring Keyword Mnemonic with Auto-generated Verbal and Visual Cues,“ in *International Conference on Artificial Intelligence in Education*, Springer, 2023, S. 16–27.

[5] T. Leelanupab und O. Anonthanasap, „Learning and immediate retention of Japanese vocabulary using generated mnemonic keywords,“ in *2017 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, IEEE, 2017, S. 315–320.

[6] O. Anonthanasap, M. Ketna und T. Leelanupab, „Automated English mnemonic keyword suggestion for learning Japanese vocabulary,“ in *2015 7th International Conference on Information Technology and Electrical Engineering (ICITEE)*, IEEE, 2015, S. 638–643.

[7] H. Bird, S. Franklin und D. Howard, „Age of acquisition and imageability ratings for a large set of words, including verbs and function words,“ *Behavior Research Methods, Instruments, & Computers*, Jg. 33, Nr. 1, S. 73–79, 2001.

[8] M. Pressley, J. R. Levin und H. D. Delaney, „The mnemonic keyword method,“ *Review of Educational Research*, Jg. 52, Nr. 1, S. 61–91, 1982.

[9] J. Wei, X. Wang, D. Schuurmans u. a., „Chain-of-thought prompting elicits reasoning in large language models,“ *Advances in Neural Information Processing Systems*, Jg. 35, S. 24 824–24 837, 2022.

[10] J. Celko, *Joe Celko's SQL for smarties: advanced SQL programming*. Elsevier, 2010.

[11] O. Anonthanasap, C. He, K. Takashima, T. Leelanupab und Y. Kitamura, „Mnemonic-based interactive interface for second-language vocabulary learning,“ *Proceedings of the Human Interface Society, HIS*, Jg. 14, 2014.

[12] *Anki*. Adresse: https://apps.ankiweb.net/.

Bibliography

[1] M. H. Ko, "Glossing and second language vocabulary learning," *Tesol Quarterly*, vol. 46, no. 1, pp. 56–79, 2012.

[2] Eurostat, *Foreign language learning statistics*, https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Foreign_language_learning_statistics, [Accessed: 31-05-2024], 2023.

[3] J. M. O'Malley, A. U. Chamot, G. Stewner-Manzanares, L. Kupper, and R. P. Russo, "Learning strategies used by beginning and intermediate esl students," *Language learning*, vol. 35, no. 1, pp. 21–46, 1985.

[4] B. Dóczi, "Comparing the vocabulary learning strategies of high school and university students: A pilot study," *WoPaLP*, vol. 5, pp. 138–158, 2011.

[5] A. L. Putnam, "Mnemonics in education: Current research and applications.," *Translational Issues in Psychological Science*, vol. 1, no. 2, p. 130, 2015.

[6] R. van de Lint and M. Bosman, "Mnemonics versus cramming. learning can be effective, efficient and fun. a systematic review studying memorization techniques in education," *CNS Spectrums*, vol. 24, no. 1, pp. 212–212, 69.

[7] S. Webb, A. Yanagisawa, and T. Uchihara, "How effective are intentional vocabulary-learning activities? a meta-analysis," *The Modern Language Journal*, vol. 104, no. 4, pp. 715–738, 2020.

[8] A. D. Baddeley, *Human memory: Theory and practice*. psychology press, 1997.

[9] J. B. Worthen and R. R. Hunt, *Mnemonology: Mnemonics for the 21st century*. Psychology Press, 2011.

[10] M. Pressley, J. R. Levin, and H. D. Delaney, "The mnemonic keyword method," *Review of Educational Research*, vol. 52, no. 1, pp. 61–91, 1982.

[11] M. R. Raugh and R. C. Atkinson, "A mnemonic method for learning a second-language vocabulary.," *Journal of Educational Psychology*, vol. 67, no. 1, p. 1, 1975, Publisher: American Psychological Association. [Online]. Available: https://psycnet.apa.org/journals/edu/67/1/1/ (visited on Jan. 16, 2024).

[12] M. Wyra, M. J. Lawson, and N. Hungi, "The mnemonic keyword method: The effects of bidirectional retrieval training and of ability to image on foreign language vocabulary recall," *Learning and instruction*, vol. 17, no. 3, pp. 360–371, 2007.

[13] Y. Chen, "Phonetic matching, semanticized phonetic matching and phono-semantic matching as techniques in keyword selection," *English Language and Literature Studies*, vol. 7, no. 1, 2017.

[14] J. Dunlosky, K. A. Rawson, E. J. Marsh, M. J. Nathan, and D. T. Willingham, "Improving students' learning with effective learning techniques: Promising directions from cognitive and educational psychology," *Psychological Science in the Public interest*, vol. 14, no. 1, pp. 4–58, 2013.

[15] A. Mizumoto and O. Takeuchi, "Examining the effectiveness of explicit instruction of vocabulary learning strategies with japanese efl university students," *Language Teaching Research*, vol. 13, no. 4, pp. 425–449, 2009.

[16] A. Campos, Á. Amor, and M. Á. González, "Presentation of keywords by means of interactive drawings," *The Spanish journal of psychology*, vol. 5, no. 2, pp. 102–109, 2002.

[17] M. Cardwell, *Dictionary of psychology*. Routledge, 2014.

[18] A. Paivio, "Dual coding theory: Retrospect and current status.," *Canadian Journal of Psychology/Revue canadienne de psychologie*, vol. 45, no. 3, p. 255, 1991.

[19] A. M. Borghi, F. Binkofski, C. Castelfranchi, F. Cimatti, C. Scorolli, and L. Tummolini, "The challenge of abstract concepts.," *Psychological Bulletin*, vol. 143, no. 3, p. 263, 2017.

[20] N. Jiang, "Lexical representation and development in a second language," *Applied linguistics*, vol. 21, no. 1, pp. 47–77, 2000.

[21] R. M. Easterbrook, "The process of vocabulary learning: Vocabulary learning strategies and beliefs about language and language learning," Ph.D. dissertation, University of Canberra, 2013.

[22] R. C. Atkinson and M. R. Raugh, "An application of the mnemonic keyword method to the acquisition of a russian vocabulary.," *Journal of experimental psychology: Human learning and memory*, vol. 1, no. 2, p. 126, 1975.

[23] A. M. Shapiro and D. L. Waters, "An investigation of the cognitive processes underlying the keyword method of foreign vocabulary learning," en, *Language Teaching Research*, vol. 9, no. 2, pp. 129–146, Apr. 2005, Publisher: SAGE Publications, ISSN: 1362-1688. DOI: 10.1191/1362168805lr151oa. [Online]. Available: https://doi.org/10.1191/1362168805lr151oa (visited on Jan. 16, 2024).

[24] T. Oku, "Reading comprehension–a keyword approach to adult efl learning–," *Mimasaka Women's Junior College*, vol. 47, pp. 21–31, 2002.

[25] J. H. Hulstijn, "1 q mnemonic methods in foreign language vocabulary learning," *Second language vocabulary acquisition: A rationale for pedagogy*, p. 203, 1997.

[26] M. Savva, A. X. Chang, C. D. Manning, and P. Hanrahan, "Transphoner: Automated mnemonic keyword generation," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2014, pp. 3725–3734.

[27] A. Paivio, J. C. Yuille, and S. A. Madigan, "Concreteness, imagery, and meaningfulness values for 925 nouns.," *Journal of experimental psychology*, vol. 76, no. 1p2, p. 1, 1968.

[28] N. C. Ellis and A. Beaton, "Psycholinguistic determinants of foreign language vocabulary learning," *Language learning*, vol. 43, no. 4, pp. 559–617, 1993.

[29] A. Campos, A. Amor, and M. A. González, "The importance of the keyword-generation method in keyword mnemonics," *Experimental psychology*, vol. 51, no. 2, pp. 125–131, 2004.

[30] C. E. Ott, D. C. Butler, R. S. Blake, and J. P. Ball, "The effect of interactive-image elaboration on the acquisition of foreign language vocabulary," *Language Learning*, vol. 23, no. 2, pp. 197–206, 1973.

[31] V. Siriganjanavong, "The mnemonic keyword method: Effects on the vocabulary acquisition and retention.," *English Language Teaching*, vol. 6, no. 10, pp. 1–10, 2013.

[32] M. H. Thomas and A. Y. Wang, "Learning by the keyword mnemonic: Looking for long-term benefits.," *Journal of Experimental Psychology: Applied*, vol. 2, no. 4, p. 330, 1996.

[33] Z. Ying, "Mental elaboration assist foreign-language vocabulary acquisition: Interactive picture with keywords," *US-China Foreign Language*, vol. 12, no. 12, 2014.

[34] M. Cancino, J. Silva, and F. Gatica, "The Role of Visual Cues in the Keyword Method: Assessing Variations of the Mnemonic Approach in L2 Vocabulary Learning," en, *MEXTESOL Journal*, vol. 45, no. 1, 2021, ERIC Number: EJ1289124. [Online]. Available: https://eric.ed.gov/?id=EJ1289124 (visited on Jan. 18, 2024).

[35] J. Lee and A. Lan, "Smartphone: Exploring keyword mnemonic with auto-generated verbal and visual cues," in *International Conference on Artificial Intelligence in Education*, Springer, 2023, pp. 16–27.

[36] J. G. Tullis and J. Qiu, "Generating mnemonics boosts recall of chemistry information.," *Journal of Experimental Psychology: Applied*, vol. 28, no. 1, p. 71, 2022.

[37] A. Campos, A. Amor, and M. A. González, "Drawing-assisted strategies in keyword mnemonics.," *Studia Psychologica*, 2004.

[38] M. González and Á. Amor, "Different strategies for keyword generation," *J. Ment. Lmagery*, vol. 28, pp. 51–58, 2004.

[39] G. Özbal, D. Pighin, and C. Strapparava, "Automation and evaluation of the keyword method for second language learning," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2014, pp. 352–357. [Online]. Available: https://aclanthology.org/P14-2058.pdf (visited on Jan. 22, 2024).

[40] O. Anonthanasap, M. Ketna, and T. Leelanupab, "Automated english mnemonic keyword suggestion for learning japanese vocabulary," in *2015 7th International Conference on Information Technology and Electrical Engineering (ICITEE)*, IEEE, 2015, pp. 638–643.

[41] J. Celko, *Joe Celko's SQL for smarties: advanced SQL programming*. Elsevier, 2010.

[42] O. Anonthanasap, C. He, K. Takashima, T. Leelanupab, and Y. Kitamura, "Mnemonic-based interactive interface for second-language vocabulary learning," *Proceedings of the Human Interface Society, HIS*, vol. 14, 2014.

[43] R. Gozalo-Brizuela and E. C. Garrido-Merchán, *A survey of Generative AI Applications*, arXiv:2306.02781 [cs], Jun. 2023. [Online]. Available: http://arxiv.org/abs/2306.02781 (visited on May 5, 2024).

[44] S. Wang, T. Xu, H. Li, *et al.*, *Large Language Models for Education: A Survey and Outlook*, arXiv:2403.18105 [cs], Apr. 2024. [Online]. Available: http://arxiv.org/abs/2403.18105 (visited on May 5, 2024).

[45] W. X. Zhao, K. Zhou, J. Li, *et al.*, *A survey of large language models*, 2023. arXiv: 2303.18223 [cs.CL].

[46] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[47] C. Zhou, Q. Li, C. Li, *et al.*, *A comprehensive survey on pretrained foundation models: A history from bert to chatgpt*, 2023. arXiv: 2302.09419 [cs.AI].

[48] R. Bommasani, D. A. Hudson, E. Adeli, *et al.*, "On the opportunities and risks of foundation models," *arXiv preprint arXiv:2108.07258*, 2021.

[49] I. Pesovski, R. Santos, R. Henriques, and V. Trajkovik, "Generative AI for Customizable Learning Experiences," *Sustainability*, vol. 16, no. 7, p. 3034, 2024, Publisher: MDPI. [Online]. Available: https://www.mdpi.com/2071-1050/16/7/3034 (visited on May 9, 2024).

[50] M. Shu, N. Balepur, S. Feng, and J. Boyd-Graber, *KARL: Knowledge-Aware Retrieval and Representations aid Retention and Learning in Students*, arXiv:2402.12291 [cs], Feb. 2024. [Online]. Available: http://arxiv.org/abs/2402.12291 (visited on May 8, 2024).

[51] H. Wong and E. Wolf, "Large language model (llm) generated personalized mnemonics," 2024.

[52] C. Zhang, C. Zhang, M. Zhang, and I. S. Kweon, *Text-to-image diffusion models in generative ai: A survey*, 2023. arXiv: 2303.07909 [cs.CV].

[53] L. Yang, Z. Zhang, Y. Song, *et al.*, "Diffusion models: A comprehensive survey of methods and applications," *ACM Computing Surveys*, vol. 56, no. 4, pp. 1–39, 2023.

[54] H.-K. Ko, G. Park, H. Jeon, J. Jo, J. Kim, and J. Seo, "Large-scale text-to-image generation models for visual artists' creative works," in *Proceedings of the 28th international conference on intelligent user interfaces*, 2023, pp. 919–933.

[55] T. Leelanupab and O. Anonthanasap, "Learning and immediate retention of japanese vocabulary using generated mnemonic keywords," in *2017 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, IEEE, 2017, pp. 315–320.

[56] D. Clegg and R. Barker, *Case method fast-track: a RAD approach*. Addison-Wesley Longman Publishing Co., Inc., 1994.

[57] X. Zhang, S. Li, B. Hauer, N. Shi, and G. Kondrak, "Don't trust chatgpt when your question is not in english: A study of multilingual abilities and types of llms," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 7915–7927.

[58] H. S. Mahdi and M. A. I. Gubeily, "The effect of using bizarre images as mnemonics to enhance vocabulary learning," *Journal of Social Studies*, vol. 24, no. 1, pp. 113–135, 2018.

[59] O. Khattab, A. Singhvi, P. Maheshwari, *et al.*, "Dspy: Compiling declarative language model calls into self-improving pipelines," *arXiv preprint arXiv:2310.03714*, 2023.

[60] D. Contributors, *Dspy optimizers and compilers*, Accessed: 2024-06-12, 2024. [Online]. Available: https://dspy-docs.vercel.app/docs/building-blocks/optimizers.

[61] R. Battle and T. Gollapudi, "The unreasonable effectiveness of eccentric automatic prompts," *arXiv preprint arXiv:2402.10949*, 2024.

[62] M. T. R. Laskar, X.-Y. Fu, C. Chen, and S. B. Tn, "Building real-world meeting summarization systems using large language models: A practical perspective," *arXiv preprint arXiv:2310.19233*, 2023.

[63] I. Meta Platforms, *Meta llama 3 70b*, Accessed: 2024-06-12, 2024. [Online]. Available: https://huggingface.co/meta-llama/Meta-Llama-3-70B.

[64] S. Zhang, L. Dong, X. Li, *et al.*, "Instruction tuning for large language models: A survey," *arXiv preprint arXiv:2308.10792*, 2023.

[65] Q. Dong, L. Li, D. Dai, *et al.*, "A survey on in-context learning," *arXiv preprint arXiv:2301.00234*, 2022.

[66] T. Brown, B. Mann, N. Ryder, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[67] M. Gruneberg, *French by Association*. Lincolnwood, IL: NTC Publishing Group, Jun. 1994.

[68] M. M. Gruneberg, *German by Association* (Link word), de. Lincolnwood, IL: NTC Publishing Group, Jan. 2001.

[69] M. Wilson, "Mrc psycholinguistic database: Machine-usable dictionary, version 2.00," *Behavior research methods, instruments, & computers*, vol. 20, no. 1, pp. 6–10, 1988.

[70] J. Wei, X. Wang, D. Schuurmans, *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 824–24 837, 2022.

[71] D. Podell, Z. English, K. Lacey, *et al.*, "Sdxl: Improving latent diffusion models for high-resolution image synthesis," *arXiv preprint arXiv:2307.01952*, 2023.

[72] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[73] *DALL·E 3*, https://openai.com/index/dall-e-3/, [Accessed 09-05-2024], 2023.

[74] D. Google, "Imagen 3: Our highest quality text-to-image model," *Google Research*, 2022, [Accessed 09-05-2024]. [Online]. Available: https://deepmind.google/technologies/imagen-3/.

[75] Midjourney, *Midjourney*, [Accessed 09-05-2024], 2024. [Online]. Available: https://www.midjourney.com/home.

[76] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 2623–2631.

[77] Student, "The probable error of a mean," *Biometrika*, pp. 1–25, 1908.

[78] A. Singhvi, M. Shetty, S. Tan, *et al.*, "Dspy assertions: Computational constraints for self-refining language model pipelines," *arXiv preprint arXiv:2312.13382*, 2023.

[79] A. Sauer, D. Lorenz, A. Blattmann, and R. Rombach, "Adversarial diffusion distillation," *arXiv preprint arXiv:2311.17042*, 2023.

[80] S. Lin, A. Wang, and X. Yang, "Sdxl-lightning: Progressive adversarial diffusion distillation," *arXiv preprint arXiv:2402.13929*, 2024.

[81] Y. Ren, X. Xia, Y. Lu, *et al.*, "Hyper-sd: Trajectory segmented consistency model for efficient image synthesis," *arXiv preprint arXiv:2404.13686*, 2024.

[82] J. Chen, C. Ge, E. Xie, *et al.*, "Pixart-\sigma: Weak-to-strong training of diffusion transformer for 4k text-to-image generation," *arXiv preprint arXiv:2403.04692*, 2024.

[83] P. Esser, S. Kulal, A. Blattmann, *et al.*, "Scaling rectified flow transformers for high-resolution image synthesis," in *Forty-first International Conference on Machine Learning*, 2024.