Virtual Dynamic Keyboard for Communication in Intensive Care

Louisa Spahl[a] and Andreas Schrader[b]
Institute of Telematics, University of Lübeck, Germany

Keywords: Active Communication, Intensive Care, Novel Interaction Devices, HCI, Virtual Keyboard.

Abstract: Effective communication is of great importance for intensive care patients in the weaning process to express their needs adequately. To support this process, the ACTIVATE patient application was developed, providing a selection of typically used texts via a novel interaction device BIRDY, intended to be used in bed. However, there are situations where patients would like to express more. Since the traditional layout of a static keyboard does not fit well with BIRDY gestures, we developed a virtual dynamic keyboard with letter prediction and minimal input gesture needs. We tested different text corpora and forecasting models, implemented a prototype based on the best candidate, and performed a preliminary user evaluation. The new virtual dynamic keyboard is shown to be superior compared to static layouts.

1 INTRODUCTION

In Germany, 2,131,216 people were admitted to intensive care units in 2017 and about 20% of them had to be ventilated (Destatis, 2018). Natural language communication with ventilated patients is usually difficult or impossible with relatives and hospital staff and intensifies stress in patients (Kordts et al., 2018). The breathing tube enters the patient's windpipe via the mouth or nose. It lies between the vocal cords and prevents them from moving so that no sound can be produced. Traditional methods, e.g., using writing tablets, are cumbersome and error-prone, and cause a high level of frustration. The presence of care staff is also a prerequisite.

The BMBF project ACTIVATE[1] has developed a special patient application for communication in intensive care units. With the help of a ball-shaped interaction device BIRDY, an application can be operated independently by patients and realize synchronous as well as asynchronous communication. The ACTIVATE patient application has a repertoire of typical sentences and actions for cognitively impaired patients. But even if most communication needs are covered in this way, the number of sentences is naturally limited. The application therefore also offers a virtual keyboard so that any text can be formulated by cognitively fit patients.

Due to the special situation of one-handed use in bed with limited mobility, BIRDY only supports a limited set of gestures, and leads to a cumbersome and time-consuming text entry process.

We therefore developed a dynamic keyboard for this special usage context to significantly speed up input and reduce patient frustration. The letters are dynamically rearranged after each input to minimize the number of gestures required. Three probability models are compared for this purpose: The stochastic Markov model (Jurafsky and Martin, 2023), the LSTM (Long Short-Term Memory) model (Hochreiter and Schmidhuber, 1997), and the Deep Learning Transformer model (Vaswani et al., 2017). Suitable text corpora are identified for training. For the evaluation of effectiveness of letter prediction and efficiency of implementation, we performed machine simulation to identify the best combination of corpora and model, which was then integrated into the ACTIVATE application and evaluated in a mixed-method laboratory study with test persons.

The paper is structured as follows. Chapter 2 provides an introduction to the ACTIVATE project. Chapter 3 describes comparable work, divided into models and dynamic keyboards. This is followed in Chapter 4 by a description of the prototype with conception, corpora, user interface, and implementation. The evaluation results are presented in chapter 5. The last chapter summarizes the results, and gives an outlook on further planned work.

[a] https://orcid.org/0009-0001-2084-7941

[b] https://orcid.org/0000-0001-7926-0611

[1] https://projekt-activate.de/

Spahl, L. and Schrader, A.
Virtual Dynamic Keyboard for Communication in Intensive Care.
DOI: 10.5220/0013099500003911

In *Proceedings of the 18th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2025) - Volume 2: HEALTHINF*, pages 41-52
ISBN: 978-989-758-731-3; ISSN: 2184-4305

2 PROJECT ACTIVATE

The ACTIVATE project, funded by the German Federal Ministry of Education and Research (BMBF), consists of several applications designed to enable communication between patients in intensive care and hospital staff during the weaning process (weaning from mechanical ventilation). Initially, only essential information about the current day (date, time, weather) or location (hospital, city, nursing staff) is given, more complex information about the therapy, physical needs or pain is added in later phases.

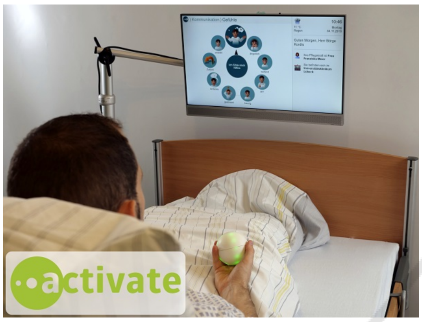

Figure 1: ACTIVATE bed-side patient application controlled via a ball-shaped novel interaction device.

The ACTIVATE system consists of (Kordts et al., 2018) (Figure 1):

- an in-bed patient application (GUI)
- a novel interaction device BIRDY
- a related mobile app for care staff

Since the application is to be operated lying in bed and the patient's hands are impaired by infusion tubes or swelling (e.g., from medication), conventional input devices are not well suited. BIRDY (Ball-shaped Interactive Rehabilitation Device) was therefore developed as a novel interactive device (Kordts et al., 2018). BIRDY contains sensors and actuators and has also been designed so that it can be used directly on the patient's body and supports cleanical disinfection. It has a rough surface to prevent slipping, is light enough (99 g) to be operated without effort, and has a size that can be easily grasped by one hand. A charging station at the patient's bedside can charge two devices wirelessly at the same time. When lying down, the movement of the patient's arms is restricted. BIRDY is therefore operated with one hand and only supports three gestures: left, right and select (press) (Kopetz et al., 2019). The sensor values are sent via Bluetooth to a single-board computer, which interprets the gestures and forwards them to a PC at the end of the bed, which in turn controls the user interface of the patient application. Messages are also sent to the staff app.

Typical texts of intensive care context have been identified in workshops and made available in the application via predefined selection fields (Figure 1). The actions in the ACTIVATE patient application are arranged in a circle to work well with the limited interactions of BIRDY. A virtual keyboard QWERTY format (German version of the QWERTY layout) is also provided for entering any text

3 STATE OF THE ART

First, we provide an overview of models for letter prediction and then present comparable approaches for virtual keyboards from literature.

3.1 Models

Neural networks have become increasingly popular in recent years. For Natural Language Processing problems, there are now various architectures, like Long-short term memory (LSTM) (Hochreiter and Schmidhuber, 1997), Transformer (Vaswani et al., 2017), or GPT-4 (Generative Pretrained Transformer) (OpenAI, 2023). All architectures have pros and cons (e.g., disappearing gradients, only possible on small text corpora, only trained in English), and the best approach must be found for the given problem.

(Verwimp et al., 2017) developed an LSTM that combines word and character embedding with representations of the word or character as a number vectors, respectively. In this approach, both the vector representation of the word as well as that of the individual characters are passed to the LSTM as input. A maximum number of characters is set. The total size of the embedding is kept constant, which is why the word embedding shrinks and is partially replaced by character embeddings, which reduces the number of parameters and thus the size and complexity of the model. Characters consist of a smaller vocabulary than words, which is reflected in a smaller embedding matrix. This special LSTM performs better than the comparable word-level model.

(Mangal et al., 2019) investigated the speed of LSTM, GRU (Gated Recurrent Unit) and bidirectional RNN. They used the script of a TV series to generate letters. They also implemented the models with different numbers of layers. All three GRU models required the least time per iteration of individual batches, closely followed by the LSTMs.

(Shakhovska et al., 2021) compared Markov chain, LSTM, and a hybrid variant based on a group of Ukrainian poems. The Markov chain required the least training time and delivered the best results.

Another word prediction network can be found in the Google Gboard. (Hard et al., 2018) trained a Coupled Input and Forget Gate (CIFG) network (Greff et al., 2015) for this purpose, a variant of the LSTM that predicts words in the virtual keyboard on the smartphone. A network is trained on the server, and fine-tuned user-specifically on the smartphone. The reason for this federated learning method is to improve data protection.

(Suliman and Leith, 2023) questioned the concept of federated learning and investigated whether it leads to increased data protection. They came to the result that sentence input can be reconstructed from the weights with a high degree of accuracy.

All works deal with the generation of words. Scientific papers on models for generating individual letters could not be found. In addition, the models mentioned were adapted according to their respective areas of application. This ranges from the creation of television manuscripts to the generation of texts in Ukrainian. For the use in intensive care context no scientific work could be found.

3.2 Virtual Dynamic Keyboards

A virtual dynamic keyboard CanAssist for people with disabilities was published in 2020[2]. The focus was on text input for the existing eye-tracking system. The letters were initially based on a dictionary approach, but later on a statistical model.

(Pouplin et al., 2014) developed a virtual dynamic keyboard for people with functional tetraplegia to accelerate the speed of text input. They used either a pointer mounted on the head as an input device, or a scanning system that scans all relevant positions on the keyboard one after the other, and the person only needs a switch to confirm the selection. This depended on the person's motor skills.

The software SibyLetter was used to predict the next letters using 5-grams and then restructure the keyboard accordingly (Schadle, 2004) (Wandmacher et al., 2008). One corpus used was the newspaper "Le Monde" with more than 100 million words. (Pouplin et al., 2014) found that the speed of text input does not increase with the virtual dynamic keyboard in a one-month test with participants.

(Ljubic et al., 2014) described a dynamic keyboard with a modified form of input. Tilting movements of a smartphone (right, left, up, and down) leads to the selection of certain areas on the screen. They compared the input speed for three types: (1) One letter is highlighted on the standard keyboard, (2) four letters are highlighted simultaneously and re-arranged when selected, and (3) the keyboard is halved after each selection until only four letters remain. All forms are based initially on the QWERTY keyboard. The third input form was the fastest, but showing only parts of keyboards might confuse users.

(Wojcik et al., 2018) describe a different keyboard layout for people with motor impairments. The five vowels (a, e, i, o, u) are arranged in a circle. All consonants are not visibly located between these vowels in alphabetical order (for example, b, c, and d are located between a and e). They conducted a user study with 20 users. Each user was asked to write 10 words with their finger and 10 words with their fist using a virtual dynamic keyboard projected onto a touch-sensitive mat (touchpad). Finger input was faster on average, both for vowels and consonants.

(Kristensson and Müllners, 2021) evaluated that the best word prediction is for a word length of six with three letters already entered.

(Agarwal et al., 2011) described a dynamic keyboard used for the secure input of passwords, without spyware being able to intercept the keystrokes. In this implementation, the letters on the virtual keyboard are randomly shuffled (without probability prediction).

A patent by (Griffin, 2013) describes a keyboard with word predictions in which the words lie on the same key as the next word to be entered. Visual highlighting is also described, such as displaying the letter key in a larger size or a different color or generally display the key or the letter back-lit, underlined, bolded, and/or italicized.

None of the works listed here used an input device comparable to BIRDY with only three interaction options. The most similar are the four tilting movements described in (Ljubic et al., 2014) and the scanning system described in (Pouplin et al., 2014). The input option is relevant for the design of the user interface in order to optimize text input. The most similar user interfaces compared to the circular menu of the ACTIVATE patient application are those of CanAssist and (Wojcik et al., 2018).

However, the input devices are different and the applications are adapted to them. (Pouplin et al., 2014) found no improvement in the speed of the dynamic keyboard. However, the test subjects used this keyboard less frequently than the familiar standard keyboard. This leads to the conclusion that the process may need a certain amount of practice and familiarization time to get used to.

[2] https://www.canassist.ca/EN/main/programs/free-downloads/dynamic-keyboard/dynamic-keyboard-development.html

4 PROTOTYPE

For designing a useful dynamic keyboard, the most probable next letters or words must be determined on the basis of the previous inputs and then arranged at a short distance to the current position.

For this, we compared different probability models and implemented the most successful model as a prototype dynamic keyboard inside the ACTIVATE user interface.

4.1 Concept

Extensive user surveys and workshops were already carried out for the existing static ACTIVATE keyboard during the analysis phase of the project (Kordts et al., 2018). Some of the requirements set out there can also be adopted for the design of the dynamic keyboard. For example, the system should be able to be used independently as possible without help. Tutorials could provide assistance. The system should be fault-tolerant and function automatically in order to save care staff time. Unnecessary acoustic or visual disturbances should be avoided. The components should be implemented interchangeably. Data communication to a backend server should always be encrypted, user privacy should be protected as much as possible, and the system should also be configurable by staff.

Additional functional and non-functional requirements were also identified, which relate to the specific aspects of the dynamic arrangement of textual elements on an interface.

At least four of the next most probable letters should be determined and displayed in the immediate vicinity to the right and left of the cursor. These should be clearly highlighted graphically. The system should be based on the design of the ACTIVATE application to prevent patients from getting used to it. The display of the letters or words should be large enough to be easily recognized from a distance of 2m from the head of the bed. The backend should display the most likely next letters as well as words. The keyboard should be able to respond correctly to input via BIRDY. Other input devices should not be excluded. The voice output should be supported by a corresponding read-aloud symbol. The existing technical implementation can be used for this purpose.

The next letters/words can only be output for the German language. The system should make it possible to conduct hospital-specific conversations. Models should provide a result in less than a second. The system may use the Internet and the models do not have to be local on the device.

4.2 Corpora

The application is initially focused on German hospitals, and therefore only German-language corpora were considered, since foreign-language corpora would not generate meaningful probabilities.

The specific use case of communication in the intensive care units requires the support of typical sentences in this environment. However, for the design of the ACTIVATE application, the main communication needs of the patients have already been recorded in numerous workshops and expert panels and made available in the application by selection in circular menus for various categories (e.g., “Ich bin durstig” (I am thirsty) or ”Ich habe Schmerzen” (I am in pain)). The keyboard should therefore support expressions that are not yet covered by the existing system. These can be conversations with doctors, nursing staff or relatives. It is assumed that patients will not have in-depth medical conversations directly after waking up during the weaning process, and specific medical terms may be irrelevant. It should also be noted that medical corpora are extremely rare, are usually only available in English and/or contain specific medical terms without conversational context.

Corpus 1: SdeWaC The SdeWaC corpus[3] is an extension of the deWaC corpus of the WaCky initiative (Baroni et al., 2009). It is available free of charge and contains more than 1.2 billion words from various web sources. Only texts with the mime type text/html and a size of 5 to 200 KB were retained.

Linguistically irrelevant texts such as warning messages or copyright declarations were also deleted. The remaining texts were cleaned of HTML and Javascript code, and duplicates (Baroni et al., 2009). Format “One sentence per line” was selected. An example is ”<year = “2007”><source = ”10475”><error = ”0”>Mehr als 7,100 Arbeitnehmer sind Datenbank-Spezialisten.” When processing the corpus for this application, the meta data, such as <year>, <source> and <error> were deleted.

Corpus 2: GGPONC The GGPONC text corpus is a specialist medical corpus. It was created in collaboration between the Hasso Plattner Institute for Digital Engineering gGmbH and the Friedrich Schiller University in Jena (Borchert et al., 2020). The corpus contains medical language based on the oncological S3 guidelines. An example sentence from the corpus is: ”Tabakkonsum ist ein wesentlicher Risikofaktor für die Entwicklung des Mundhöhlenkarzinoms.” (Tobacco consumption is a significant risk factor for the development of oral cavity carcinoma.)

[3] https://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/sdewac/

Corpus 3: Gutenberg-De The Gutenberg-DE corpus[4] contains texts from the Gutenberg-De project. A very small selection of copyright-free books was made as a comparative corpus.

- *Die Frau von dreißig Jahren* by Honoré de Balzac translated by Hedwig Lachmann
- *Student und Alkohol* by Dr. Leopold Loewenfeld
- *Das Haus* by Lou Andreas-Salomé
- *Eiszeit und Klimawandel* by Wilhelm Bölsche

The texts were chosen at random, but care was taken to use modern language and spelling. Due to the conditions of the German Copyright Act (UrhG, §64), the texts are at least 70 years old. This means that the new rules following the spelling reform of 1996 have not been taken into account. As a result, the probabilities of outdated special characters such as “ß” (instead of “ss”) differ from current texts.

All three corpora were cleaned up: uppercase letters are converted to lowercase, and all characters that are not a-zäöüß, digits 0-9, or space are removed. The above example results in ”mehr als 7 100 arbeitnehmer sind datenbank spezialisten”. The semantic meaning may change as a result of the filtering. For the statistical interpretation in the dynamic keyboard, however, the semantic distortion is irrelevant, as special characters are not supported anyhow.

4.3 User Interface

Based on the functional requirements, there are several options for keyboard design.

The original idea of arranging the letters in a circle (Fig. 2) was ruled out during the design process. The circular menu in the ACTIVATE patient application includes a limited set of elements for close communication and information display. A circular keyboard could appear overloaded with a total of 30 letter keys, the space bar, and additional function keys (such as Back and Delete) and exceeds the screen size.

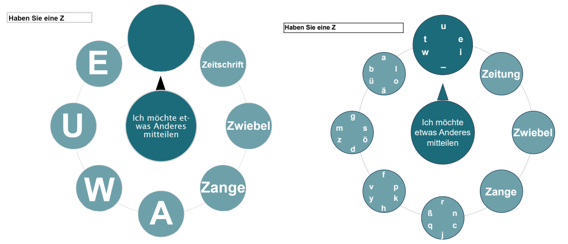

Figure 2: Circular design layout of the dynamic keyboard. Left: dedicated words on keys. Right: Grouping of letters.

Alternatively, only a limited number of letter keys could be displayed, which would mean that not all words could be written. A possible solution could use multiple letters per key as in CanAssist (Fig. 2). However, it requires at least two actions per letter, because first the group must be selected, and then the letter must be selected after redistributing the letters contained to a new circle.

Most people are familiar with a line-shaped keyboard. When entering words with the BIRDY input device, it does not matter whether the letters are arranged in a circle or lines, as BIRDY does not support up and down movements to switch between lines.

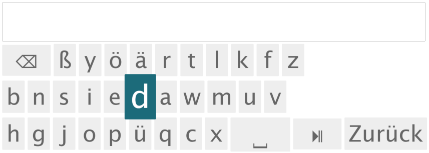

Figure 3: Linear design layout of the dynamic keyboard.

Letters and characters as well as action keys are displayed in lines, just like on a traditional computer keyboard. A left movement from the first letter of the first line enables a jump to the last letter of the last line and vice versa by a right movement. A keyboard that is arranged in lines therefore has the same functionality as a circular keyboard but can be displayed more clearly on a screen.

The size of the letters has been adopted from the static keyboard. The currently most likely letter is displayed larger and with a dark background in the middle. The next most likely letters are placed to the right and left around it, and can therefore be reached with just a few gestures.

All 30 letters ’a-zöäüß’ must be displayed so that all conceivable texts can be entered. The less likely a letter is, the further away it is positioned. In Fig. 3 these are the letters ’ß’ at the top left and ’x’ at the bottom right. There is also a delete button on the far left. The distance should not play a major role in everyday use, as the correctness of the input is of secondary importance. At the right end there is a space, the symbol for starting the read-aloud function, and a back button.

In a circular keyboard, it would still be conceivable to display complete words on keys, either by omitting or by grouping letters (Fig. 2). In a line-shaped keyboard, only four options remain:

1. directly after the next n most probable letters on the keyboard
2. at the end after the least probable letters
3. at a different position, e.g., at the top or bottom
4. next to its first letter

[4] https://www.projekt-gutenberg.org/info/texte/allworka.html.

With all four options, the keyboard may look too cluttered for the patient. In option 1, a visual break is made and words or letters would change in a loose sequence. In option 2, the path to the words is probably so long that it would not be worthwhile for patients to navigate to them, especially for short words. Option 3 has a similar problem to option 2, but could be a better alternative. Option 4, like option 1, also has the problem of a visual break. With each option, the keyboard appears cluttered with additional words, and we have decided against using complete words.

4.4 Implementation

The dynamic keyboard was implemented as a client-server architecture with Docker containers.

The Markov model and the LSTM are trained on the server. The Transformer model is a pre-trained external model that is accessed via an external interface. For the probability models, we only use 30 lowercase letters (a-zäüöß), 10 digits and the space character, i.e. a total of 41 characters. This means that the character set is significantly smaller than the word set of the German language, and the training of the models requires considerably less time and memory.

The Markov chain is a stochastic process that is trained quickly (in a few hours), and can answer queries quickly. For this purpose, a histogram of N-grams is determined for the respective corpus (using the ngrams() function of the Python nltk9 package[5]). We have implemented bigrams (N=2) and trigrams (N=3) for this purpose. The calculation is aborted as soon as saturation occurs, i.e. the probabilities of the N-grams no longer change or change only slightly. If certain trigram combinations did not occur in the selected corpus, the letters were nevertheless assigned to keys, but with a low probability. After estimation, the dictionary is stored in a document database (MongoDB[6]) to avoid re-training in subsequent calls.

The Transformer-based language model GPT from the company OpenAI has been widely known since ChatGPT[7] at the latest. We used a pre-trained GPT2 network benjamin/gerpt2-large (Minixhofer, 2020) based on German texts. The network completes a partial sentence with several words, often even an entire paragraph. Minixhofer uses the AutoModelForCausalLM model. The number of generated words can be set in the model using the max_new_tokens parameter (we tested the values 1, 2 and 10). For the dynamic keyboard, the network's return was adapted so that only individual letters are used. Transformer nets are not deterministic. To generate a letter probability from the sentence prediction, the network was called several times.

The LSTM was implemented based on the PyTorch Tutorial[8], which supports both word and letter predictions. Despite the relatively high-performance computer (GPU: NVIDIA RTX 3090 with 24 gigabytes of RAM, CPU: AMD Ryzon 9 5950X with 128 gigabytes of RAM) and splitting the corpus of more than 5 GByte into files of 180-360 MByte, the computing time for the training based on the cleaned SdeWaC corpus took several months. Due to the relative similarity of the trigrams based on the Gutenberg-DE corpus, further training was only carried out with this data. The corpus has a size of 8,778 lines and a total of 166,416 words and slightly more than 1 million letters, and the training could be completed in a few hours. The trained LSTM was persisted in a zip file.

The server was implemented in Python (version 3.10) and offers a RESTful API with five endpoints, which accesses the respective pre-trained data:

- give next letter using Transformer
- give next letter using Markov chain bigram
- give next letter using Markov chain trigram
- give next letter using LSTM
- write log file

In addition, code for the software-side evaluation of the predictions based on test data sets has also been implemented on the server.

The graphical user interface (client) is implemented as an extension of the ACTIVATE patient application with the help of the technologies used there (HTML, CSS, vanilla JavaScript and JavaScript framework Vue.js (Kopetz et al., 2021)). The client is based on the vue-keyboard package[9] and asks the server for a list of the next probable letters. The list returned by the server is transformed so that the probabilities of the letters decrease from the middle to the ends of the list. The elements of the transformed list are then distributed to three rows of keys on the virtual keyboard.

5 EVALUATION

We performed several steps of comparison and tests to evaluate our prototype implementation. First, we

[5] ttps://www.nltk.org/api/nltk.util.tml

[6] https://www.mongodb.com/de-de

[7] https://openai.com/

[8] https://closeheat.com/blog/pytorch-lstm-text-generation-tutorial

[9] https://github.com/MartyWallace/vue-keyboard,Version3.1.0

compared the theoretical minimum of steps for the different corpora to identify the ideal training data set. Second, we compared the models to identify the best forecasting system. Third, we performed a user test to evaluate human interaction.

5.1 Corpora Comparison

In an online survey conducted beforehand, participants were asked to write down texts they could imagine themselves saying as a patient in a weaning process. 361 example sentences from 74 participants were cleaned from redundancy and special characters resulting in a test set of 257 sentences used to compare with the three other different corpora introduced above. The Markov model with trigrams is used for the comparison. The next step is to check whether a specialist medical corpus, a high-quality literary text or a corpus collected from the Internet can change the results and further reduce the number of steps. This refers to movements to the right or left on the keyboard. The average number of steps calculated across all test sentences when using trigrams can be seen in Table 1. In one case, the average number is lower than one. This is based on cases, where the predicted letter placed in the middle is correct, and therefore, no steps had to be performed.

Table 1: Average number of steps (left, right - pressing ignored) when using trigrams across all test sets.

Korpus	Number of steps ∅
SdeWaC	**1.59**594697
GGPONC	**1.95**818220
Gutenberg-De	**1.59**026818
Example sentences	**0.95**412386

The difference between the SdeWaC and the medical corpus is relatively large. The medical corpus is similarly poor with trigrams as the normal corpus with bigrams. The selected medical corpus contains specialized medical text on carcinomas based on the oncological S3 guidelines. Words such as laryngeal carcinoma, screening, genotyping, cytology, etc. are not rare words in the corpus. However, these words are probably not used by people in the weaning process. Otherwise, they would already be available as standard text in the ACTIVATE patient application. The example sentences from the survey also mainly contain everyday language. The medical terms used in the survey are weaning process, ventilator, treatment, pain and healing process. The last three terms (or a modified form of these) are probably more commonly used normally in everyday language. For these reasons, the SdeWaC corpus fits the use case better. Nevertheless, it should be mentioned that the sentences from the survey are only a snapshot, and the sentences may still differ from those from the real situation in the weaning process.

The average number of steps in the SdeWaC and Gutenberg-De corpora are very similar. The corpora have no textual overlap. The SdeWaC contains both fictional and non-fictional texts. Although some texts are informative texts from news and vacation destination websites, they are not recognizable specialist literature websites like the texts in the GGPONC corpus. The Gutenberg-De corpus is based on fictional texts (stories, novels). The non-fictional texts therefore resemble the fictional Gutenberg texts with little technical language in terms of N-gram distribution. In the SdeWaC corpus, some technical terms from medicine can be found, but this is the exception. Therefore, SdeWac and GGPONC have different letter probabilities, and SdeWaC and Gutenberg-De corpus have similar numbers of steps. The average number of steps and the step frequency distribution are best for the test sets.

5.2 Model Evaluation

In the following, we present comparison results of letter forecast performance of each model.

5.2.1 Markov

The developed system is tested with sample sentences from the survey. The best input scenario is assumed for both the standard keyboard and the dynamic keyboard, i.e. simulating a user without errors, and not using back keys. With the standard keyboard the start position is at letter '1', with the dynamic keyboard the initial setup is starting with letter 'd' in the middle.

As expected, the standard keyboard is very inefficient. The average number of steps per letter tested with the example sentences is 13.83 for the standard keyboard, 2.07 for the Markov model bigram and 1.6 for the Markov model trigram (see Fig. 4) For both the bigram and the trigram, the probability that the next letter is 0 steps or one step away is higher than 50%. With the standard keyboard it is very arbitrary and the probabilities from 0 to 13 steps add up to more than 50%.

The measured time difference of the system time before and after the function call is 0.03 seconds on average (rounded), well below the maximum acceptable delay between input and feedback of 500 ms specified by (Wolff et al., 2011). The bi- and trigrams for the Markov model were created from the SdeWaC corpus and stored in the database in a Docker container.

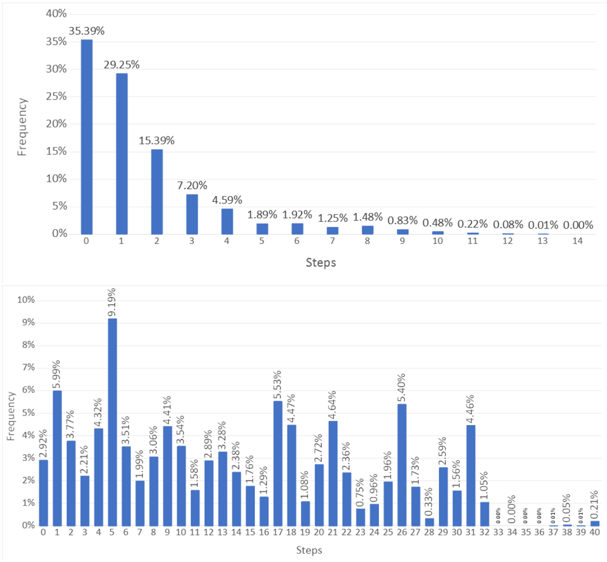

Figure 4: Probability of steps based on trigrams for dynamic (top, SdeWaC-corpus) versus static keyboard (bottom).

5.2.2 Transformer

The pre-trained GPT2 network is designed to generate several words up to sentences and paragraphs. For this work, however, the network should predict the next best possible letters. If no context is given to the network, but it is to generate further words with a small number of given letters, special characters are often returned. These are initially irrelevant for the dynamic keyboard. With the parameters *bad_words_ids* and *max_new_tokens* and the context sentence, ”Ich liege im Krankenhaus und werde beatmet. Jetzt möchte ich mit dem Arzt kommunizieren.“ (I'm in hospital and being ventilated. Now I want to communicate with the doctor.) which was added before each example sentence, acceptable sentences could be generated. The first generated letter is chosen as one of the likely next letters. To generate more than one letter, the same sentence must be given to the model multiple times (iteration). After each predicted letter, the iteration counter decreases by one. If the same letter is predicted multiple times, its probability increases, but the number of different letters remains the same. To prevent an endless loop, the maximum number of iterations is set to 20. With 30 letters in the german alphabet and one space character, the most distant letter is 15 steps away from the pointer (focus in the middle). This is half of the possible characters (a-zäöüß and space). The average number of steps for a sentence is 6.09 (rounded) for the transformer without context and 6.06 (rounded) for the transformer with context. The step frequency distribution for the transformer does not show a uniform decrease in probability like the Markov model trigram, as it is not based on a probability distribution but learned letter patterns. Additionally, we limit generation to the ten most probable next letters to reduce processing time, resulting in a maximum of five steps. If the results are compared with context and without context, it can be seen that steps zero and one are 2% more likely with context. From six steps at the latest, all remaining characters that were not generated by the transformer are added with no particular order. This means that a high probability of steps greater than five is possible, but can change with each evaluation.

The average processing time required is 17.84 seconds for the Transformer with context and 16.51 seconds without the specified hospital context. The times measured for the Transformer exceed the maximum feedback value specified by (Wolff et al., 2011) by far. Times for using the GUI would add to this.

5.2.3 LSTM

The evaluation of the LSTM resulted in an average of 5.84 (rounded) steps for one sentence. One step is the most frequent, followed by two and five steps. The distribution does not resemble the bi- or trigram distribution of the Markov model. Unlike the Transformer, no special characters are generated because the training corpus does not contain them. The LSTM only generates a maximum of the next ten possible letters. All other existing letters in the alphabet (n=21) are appended to the list of generated letters. This means that all letters can still be entered using the keyboard. Ten generated letters on the keyboard means that a maximum of five steps are required to reach them.

The step frequency distribution of the LSTM shows that the sum of the probabilities for zero to five steps is 57.98%, which is more than half of all step probabilities (steps 0-15). However, the result of the LSTM is still worse than that of the Markov model (bi- and trigram). With the bigram, the first two steps (0,1,2) already have a probability of 71.4% and with the trigrams even 80.03%. The measured processing time is 1.33 seconds (rounded). The response time is significantly better than the transformer network's, but still too high to be acceptable.

Since the transformer model was rather slow and created a lot of special characters that have to be filtered, and the LSTM model provided lower prediction quality, we decided to integrate the deterministic Markov model into the dynamic keyboard application.

5.3 User Evaluation

In the laboratory test described above, perfect users without any time spent on the gestures and without errors in selecting the correct letters, were simulated. For the usability test of the user interface and its use by real people, we also conducted an initial laboratory study with test subjects. The aim was to test the effectiveness of the new design in comparison with the static keyboard, as well as the perception of the changing letters on the keyboard among participants.

The qualitative study uses a mix of methods that includes observing the interactions of the test subjects, in particular their movements within the application, as well as a structured questionnaire afterward.

The semi-randomized study took place in our laboratory, where a hospital-like (intensive care) environment was created. The ACTIVATE application and BIRDY were provided at a hospital bedside. The inclination of the head section and the positioning of the screen were adapted to real conditions in the intensive care unit. The test person lies in bed and holds BIRDY in his hand while it lies on the bed.

Voluntary test subjects (n=9) were recruited for the study from among the students and employees of our university using a survey in our learning management system. Upon request, student participants received test points for the 30-minute test.

Inclusion criteria included a minimum age of 18 years and German language skills. Participants with physical impairments of arms and legs, dyslexia, visual impairment or blindness, mental impairments, and other illnesses that could impair the successful completion of the task (e.g., epilepsy) were excluded.

After the purpose of the study has been explained to the subjects and presented to them in writing, they signed a declaration of consent and a data protection declaration. The test person then lies down in the hospital bed and is given BIRDY in the hand. BIRDY is first calibrated with the help of the study staff. In particular, the magnetometer to determine the Earth's reference system is affected by constant interference and must be adjusted. The accelerometer and the gyroscope are less affected.

The system was briefly explained to the test person and the basics are shown. This was followed by the sentence input. To familiarize themselves with the system, the test subjects select three sentences provided by the ACTIVATE system: “Wann kommt meine Familie?” (“When is my family coming?”), ”Was ist passiert?” (“What happened?”), and ”Durst” (‘Thirst’) via the patient application.

Three further sentences were selected from the results of the survey, to avoid long input sequences: “danke” (“thank you”), “wer sind sie” (“who are you”), and ”ich liebe euch” (“I love you”). Please note, that the sentences are not following correct German grammar and spelling, since the keyboard does not provide capital letters. This is obviously of minor importance in the context of intensive care communication. The sentences are entered by the test subjects using both the dynamic and the standard keyboard.

In the background, the program measures the system time and records the number of BIRDY actions (step to the right, left, and confirmation by push gesture). After successful completion, the test subjects fill out a questionnaire. The questionnaire is pseudoanonymous and does not ask for any personal data. Participants can indicate wishes for further functions or additional characters.

Nine test subjects registered for and participated in the laboratory study. The age of participations was between 21 and 36 years, the average was 26.5 years. Six of the test subjects felt they belonged to the female gender, and three test subjects to the male gender. All test subjects successfully completed the first task of entering sentences with the ACTIVATE patient application. This task was intended to familiarize them with the system and the input device and is not discussed further here. In the second task, the test subjects were asked to enter sentences using the dynamic keyboard. This task was also successfully completed by all participants.

In the third task, entering three sentences using the standard keyboard, two test subjects had technical difficulties with the BIRDY input device, so that this could not be completed. For this reason, we decided to discontinue the respective tests. This affected the sentences ”wer sind sie” und ”ich liebe euch” for two test subjects.

In the step evaluation, the steps were counted from opening the respective keypad to starting the pronunciation via the corresponding symbol. With the dynamic keyboard, the symbol was always 16 steps (15 right and 1 pressing movement) away from the last input, because the pointer (highlighted letter) is always in the center of the dynamic keyboard (see above). The distance to the symbol on the standard keyboard depends on the input of the last letter. It should also be noted here that the test person can take a step to the left from the first keyboard symbol at the top left (number 1) to land at "Zurück" (Back) at the bottom right (3). The steps taken per participant, sentence and keyboard are compared with the best possible steps. For the standard keyboard in particular, a movement from top left to bottom right and vice versa was considered if this path was the shorter one.

5.3.1 Results: Number of Steps

The shortest path for the first sentence "danke", from opening the dynamic keyboard to the start of the output requires 27 steps. Of these, 11 steps were required to type the word/phrase and 16 steps to reach and activate the symbol. Three out of nine test subjects managed it in 27 steps, three were slightly worse (+2, +4, +6 steps) and three needed significantly more steps.

For the second sentence "wer sind sie", a minimum of 36 (20+16) steps were required to enter the text without errors. Four test subjects managed this. Two were slightly above this (+2 steps) and three test subjects needed more steps to enter the text. The third sentence to be entered, "ich liebe euch", required a minimum of 57 (41+16) steps. One test person made it, four were slightly above that (+2, +4) and four missed the target by a long way. For two of the last group, the problem was with the input device, which no longer responded as desired. These were the test subjects who dropped out later.

None of the nine test subjects managed to enter all sentences correctly using the dynamic keyboard without errors. This confirms that it is important to conduct a laboratory test with people to evaluate the system instead of just using software to calculate the smallest steps.

With the standard keyboard, no test person managed to enter one of the three sentences with the minimum steps. These were 74+16=90 for "danke", 155+16=171 for "wer sind sie" and 221+18=239 for "ich liebe euch" (the second summand is the path to the read-aloud symbol). As already mentioned, the steps to the symbol depend on the previous letter. In the first two sentences, by chance, there are also 16 steps (from the letter 'e') and for the last sentence, it is 18 steps (starting from 'h'). Only one test subject managed to get close to the best possible number of steps. This person managed 97 instead of 90 steps in the first set and 244 instead of 239 steps in the last set. Two test subjects dropped out during the last two sets, leaving only 7 relevant test subjects at the end,

Consequently, the dynamic keyboard offered the test subjects considerable added value in terms of compared to the standard keyboard. The dynamic keyboard requires fewer steps per sentence.

5.3.2 Results: Required Time

The evaluation of the required time is somewhat more complex, as there is no minimum reference time. For different reasons, each person takes a different amount of time to enter text into the keyboards.

In total, with the dynamic keyboard, it was possible to enter all 27 text entries. 15 sentences could be entered in less than one minute. The first two sentences in particular ("danke", "wer sind sie") required a shorter input time. With the standard keyboard, only two entries could be entered in under a minute. When calculating the average duration, the two test subjects who aborted were not taken into account. The previous sentences already reflect the difficulties with the BIRDY input device. The two test subjects waited in between, hoping that BIRDY would recover after a waiting period. The result can be seen in table 2.

Table 2: Average time needed to input the sentences once with the dynamic keyboard and once with the standard keyboard (N=7).

	Time format mm:ss,ms	
	Keyboard	
Sentences	Dynamic	Standard
danke	00:42,096	01:21,347
wer sind sie	00:40,601	03:12,840
ich liebe euch	00:58,927	03:08,014

The small sample size does not allow a quantitatively significant statement to be made. Nevertheless, it can be seen that typing with the dynamic keyboard is faster in all cases than with the standard keyboard.

5.3.3 Results: Questionary

A questionnaire with seven questions was presented to the subjects after the tests. All test subjects were comfortable with the dynamic keyboard and would continue to use it instead of the standard keyboard. Only one person had already used a different dynamic keyboard before, for all others it was new.

Six out of nine test subjects would like to see more characters on the keyboard. Numbers were mentioned in particular, periods, commas, question and exclamation marks. positioning of additional characters were preferred dynamically (n=3), statically at the edge of the keyboard (n=2), or dynamically only at the potential end of a sentence (n=1).

It was also suggested that a number written in full could be automatically converted into a sequence of digits. But there is no added value in this case. The text entered would only be shortened, which is irrelevant for pronunciation. There would only be an advantage if the input of a number is recognized early. Especially it could perhaps be an advantage with long numbers.

Eight out of nine test subjects see an additional benefit when using the dynamic keyboard compared to the ACTIVATE patient application. The reasons were similar for most of them. They stated that more individualized utterances were possible, and less cognitive effort or less concentration would be required. All of them saw an additional benefit of the dynamic keyboard compared to the existing standard keyboard. The dynamic keyboard is faster to type (provided the word is recognized correctly) and more intuitive.

Vertical movements by BIRDY were desired. This presumably related to the standard keyboard, as the letters of the laboratory test sentences on the dynamic keyboard were almost always on the same line (with the exception of the read-aloud symbol). Vertical movements were ruled out in the design of the device in user studies as not comfortable for the position in bed.

An idea was described to have another input device that takes over the function of the space bar, the delete key, and the pronunciation function. Although this could speed up input, it would lead to confusion for patients. The input device would have to be used in parallel with BIRDY in the other hand and the patient would have to memorize which input device is responsible for device is responsible for what and which gestures exist if they are different.

Suggestions were also made to predict and display words or to be able to memorize words and sentences. Predicting words and integrating them into the keyboard has already been implemented in variants of the prototype. In this case, a fixed number of words was dynamically added to the keyboard, depending on where the initial letter of the word was located. This is similar to the patent application by (Griffin, 2013).

6 CONCLUSIONS AND OUTLOOK

This work focuses on the development and evaluation of a dynamic virtual keyboard for intensive care units. Integrated into the ACTIVATE project's communication system, the keyboard uses the BIRDY input device, designed for circular menus with simple left, right, and pressure gestures. For traditional keyboard layouts, this control is cumbersome and requires many movements.

The dynamic keyboard uses statistical models to predict letters based on previous input, dynamically rearranging their positions to minimize interactions. Three models were implemented: Markov, LSTM, and Transformer. The Markov model with trigrams proved most efficient, averaging 1.6 steps per letter compared to 13.83 for a standard keyboard. In a laboratory evaluation, users typed 73% faster with the dynamic keyboard and preferred it over the standard one.

The design of the GUI layout was not the focus. And although graphic cues like highlighting letters as well as optional voice output are already provided, user-friendliness can further be optimized through an ISO-compliant standard UCX process. Especially, an integrated tutorial system would be beneficial for beginners.

Continuous model adaptation could enable user-specific customization. While it may be straightforward to simply adjust the Markov-Model based on the patient typing history over time and support real-time adaptation, it may even degrade prediction quality due to patients' errors in typing. Since manual correction is not a realistic option, another speech model for automatic syntax correction would be required. But most patients will not type much text anyhow, therefore the potential advantage is limited.

Larger user tests are planned to confirm results, with a future clinical study to determine effectiveness for real patients.

ACKNOWLEDGEMENTS

The authors would like to express their gratitude to ACTIVATE research team members Boerge Kordts and Jan Kopetz for supporting the integration of our results into the ACTIVATE system architecture and for very helpful concept discussions.

REFERENCES

Agarwal, M., Mehra, M., Pawar, R., and Shah, D. (2011). Secure authentication using dynamic virtual keyboard layout. In *Proceedings of the International Conference and Workshop on Emerging Trends in Technology*, ICWET '11, page 288–291. Association for Computing Machinery.

Baroni, M., Bernardini, S., Ferraresi, A., and Zanchetta, E. (2009). The wacky wide web: A collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43:209–226.

Borchert, F., Lohr, C., Modersohn, L., Langer, T., Follmann, M., Sachs, J. P., Hahn, U., and Schapranow, M.-P. (2020). GGPONC: A corpus of German medical text with rich metadata based on clinical practice guidelines. In *Proceedings of the 11th International Workshop on Health Text Mining and Information Analysis*, pages 38–48, Online. Association for Computational Linguistics.

Destatis, S. B. (2018). Gesundheit. Grunddaten der Krankenhäuser 2017. Fachserie 12 Reihe 6.1.1. Artikelnummer 2120611177004.

Greff, K., Srivastava, R., Koutník, J., Steunebrink, B., and Schmidhuber, J. (2015). LSTM: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28.

Griffin, J. T. (2013). Next letter prediction for virtual keyboard. https://worldwide.espacenet.com/patent/search-/family/046875664/publication/US9134810B2?q=pn%3DUS9134810B2.

Hard, A. S., Rao, K., Mathews, R., Beaufays, F., Augenstein, S., Eichner, H., Kiddon, C., and Ramage, D. (2018). Federated learning for mobile keyboard prediction. *ArXiv*, abs/1811.03604.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9:1735–80.

Jurafsky, D. and Martin, J. H. (2023). *N-gram Language Models*, chapter 3. Prentice Hall, 3rd edition. Draft.

Kopetz, J., Jochems, N., Henkel, A., Schley, A., Balzer, K., Kordts, B., and Schrader, A. (2021). *ACTIVATE. Sozio-technisches System zur Unterstützung der Kommunikation von Intensivpatienten*. BMBF Forschungsvorhaben, Universität zu Lübeck. Final report.

Kopetz, J. P., Burgsmüller, S., Vandereike, A.-K., Sengpiel, M., Wessel, D., and Jochems, N. (2019). Finding user preferences designing the innovative interaction device "birdy" for intensive care patients. In Bagnara, S., Tartaglia, R., Albolino, S., Alexander, T., and Fujita, Y., editors, *Proceedings of the 20th Congress of the International Ergonomics Association (IEA 2018)*, pages 698–707, Cham. Springer International Publishing.

Kordts, B., Kopetz, J. P., Balzer, K., and Jochems, N. (2018). Requirements for a system supporting patient communication in intensive care in germany. In Boll, S., Hein, A., Heuten, W., and Wolf-Ostermann, K., editors, *Zukunft der Pflege : Tagungsband der 1. Clusterkonferenz 2018 - Innovative Technologien für die Pflege*, pages 131–136. BIS-Verl. der Carl von Ossietzky Universität Oldenburg.

Kristensson, P. O. and Müllners, T. (2021). Design and analysis of intelligent text entry systems with function structure models and envelope analysis. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21. Association for Computing Machinery.

Ljubic, S., Glavinic, V., and Kukec, M. (2014). Predicting upper-bound text entry speeds for discrete-tilt-based input on smartphones. *Journal of Interaction Science*, 2.

Mangal, S., Joshi, P., and Modak, R. (2019). LSTM vs. GRU vs. bidirectional RNN for script generation. *CoRR*, abs/1908.04332. http://arxiv.org/abs/1908.04332[Accessed:(25.09.2024)].

Minixhofer, B. (2020). GerPT2: German large and small versions of GPT2. https://github.com/bminixhofer/gerpt2 [Accessed: (25.09.2024)].

OpenAI (2023). Gpt-4 technical report. https://arxiv.org/abs/2303.08774 [Accessed: (25.09.2024)].

Pouplin, S., Robertson, J., Antoine, J.-Y., Blanchet, A., Kahloun, J., Volle, P., Bouteille, J., Lofaso, F., and Bensmail, D. (2014). Effect of a dynamic keyboard and word prediction systems on text input speed in patients with functional tetraplegia. *Journal of rehabilitation research and development*, 51:467–480.

Schadle, I. (2004). Sibyl: Aac system using nlp techniques. In Miesenberger, K., Klaus, J., Zagler, W. L., and Burger, D., editors, *Computers Helping People with Special Needs*, pages 1009–1015, Berlin, Heidelberg. Springer.

Shakhovska, K., Dumyn, I., Kryvinska, N., and Kagita, M. K. (2021). An approach for a next-word prediction for ukrainian language. *Wireless Communications and Mobile Computing*, 2021:1–9.

Suliman, M. and Leith, D. (2023). Two models are better than one: Federated learning is not private for google gboard next word prediction. In *European Symposium on Research in Computer Security*, pages 105–122.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *CoRR*, abs/1706.03762.

Verwimp, L., Pelemans, J., Van hamme, H., and Wambacq, P. (2017). Character-word LSTM language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 417–427, Valencia, Spain. Association for Computational Linguistics.

Wandmacher, T., Antoine, J.-Y., Poirier, F., and Départe, J.-P. (2008). Sibylle, an assistive communication system adapting to the context and its user. *ACM Trans. Access. Comput.*, 1(1).

Wojcik, B., Morelli, T., and Hoeft, B. (2018). RingBoard – A Dynamic Virtual Keyboard for Fist Based Text Entry. *Journal on Technology and Persons with Disabilities*, page 83.

Wolff, S., Kohrs, C., Scheich, H., and Brechmann, A. (2011). Auswirkungen von prosodisch-motivationalen und verzögerten Rückmeldungen auf die Lernleistung und Hirnaktivitat in einer Mensch-Computer Interaktion. In *INFORMATIK 2011*, pages 238–238. GI e.V., Bonn.