

Research Article

An Approach for a Next-Word Prediction for Ukrainian Language

Khrystyna Shakhovska ¹, Iryna Dumyn ¹, Natalia Kryvinska ²,
and Mohan Krishna Kagita ³

¹Artificial Intelligence Department, Lviv Polytechnic National University, Lviv 79013, Ukraine

²Department of Information Systems, Comenius University in Bratislava, Bratislava 81499, Slovakia

³School of Computing and Mathematics, Charles Sturt University, Melbourne, Australia

Correspondence should be addressed to Iryna Dumyn; irka.shvorob@gmail.com

Received 20 June 2021; Accepted 29 July 2021; Published 15 August 2021

Academic Editor: Rajesh Kaluri

Copyright © 2021 Khrystyna Shakhovska et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Text generation, in particular, next-word prediction, is convenient for users because it helps to type without errors and faster. Therefore, a personalized text prediction system is a vital analysis topic for all languages, primarily for Ukrainian, because of limited support for the Ukrainian language tools. LSTM and Markov chains and their hybrid were chosen for next-word prediction. Their sequential nature (current output depends on previous) helps to successfully cope with the next-word prediction task. The Markov chains presented the fastest and adequate results. The hybrid model presents adequate results but it works slowly. Using the model, user can generate not only one word but also a few or a sentence or several sentences, unlike T9.

1. Introduction

A personal predictive text input system, also known as T9, has become an integral part of our text input lives. Therapists often recommend word prediction as a method of typing speed improvement for users with physical limitations. Even though papers claim that word suggestion affects writing skills, boosted speed could not be mentioned as its benefit if using a standard keyboard. Users should look away while typing to check predicted words; it could be a reason for word prediction failure because it could slow the user more than boosting given by word suggestion. For those cases when the typist must distract from the document, this effect is not related. The goal of research [1] was to conclude whether word prediction applications would improve typing speed when a user should look away from the document. The experiment shows that seven out of ten participants improved typing speed with word prediction. So the result indicates that word prediction could be helpful for users.

However, errors can often be observed when generating the next word. Therefore, this work is aimed at analysing the existing methods of the next-word prediction based on

the entered text and testing them in Ukrainian language content. A possible obstacle will be the lack of Ukrainian language corpus and models supporting Ukrainian language (for example, GPT, BERT, and GPT2 for English). Also, a novelty in contrast to T9 will be the ability to predict not just one word but also a given number of words/sentences.

The work is aimed at developing a predictive system of Ukrainian language text input, which should improve the correctness of the next word. The research methods are machine learning algorithms: Markov chains and LSTM and their hybrid.

In the paper [2], the authors using a recurrent neural network trained language model. It was developed with the help of federated learning (FL). Federated learning is distributed, on-device learning framework currently called for next-word prediction. The significant advantage of method implementation is training on client devices because it helps to consolidate privacy. The federated averaging algorithm, which is used on client devices, shows better prediction recall than server-based training using stochastic gradient descent. One reason is training on a better quality dataset for this use case because a user customizes the dataset. Furthermore, one

more benefit is no need to export user data to servers. Long short-term memory (LSTM) with a Coupled Input and Forget Gate (CIFG) language model trained on the server and baseline n-gram model was compared to the federated learning model trained from scratch. It was shown that the created model outperforms the keyboard next-word prediction task. To add, this is one of the first federated language modelling applications in a commercial setting.

In the following paper [3], the authors had the same idea of using federated learning without direct access to the user data for commercial purposes. Apart from that, to improve search suggestion of virtual keyboard quality by global-scale setting training and evaluate results, the proposed method is a logistic regression model applied by federated learning. The first step is training the baseline model, which generates query candidates on the server side, the second step is triggering model trained by federated learning. The aim was to improve the query click through rate (CTR) by applying a baseline model for taking suggestions and the triggering model to remove low-quality recommendations. The comparison between wanted Δ CTR in training metrics and received Δ CTR in live experiments shows that real-life deployments significantly improve CTR. The result of this paper is also an early end-to-end example of FL in a commercial.

In paper [4], the authors research various sequences to sequence deep learning models in scripts for TV series generation: conversations and scenarios. LSTM, bidirectional recurrent neural network (RNN), and gated recurrent units (GRU) were analysed. Input sequence consists of n characters. Therefore, targets consist of a similar amount of characters. One possible exception is that character could be shifted one to the right. Everything is stored in a pickle file what allows to train a model for script generation tasks with no need for human intervention. The LSTM shows the most reliable performance, and then, in the second place is GRU and last but not least bidirectional RNN. The loss analysis demonstrates that the least lost is in bidirectional RNN and then LSTM and the highest is in GRU. Execution time is the least in LSTM, GRU works a bit more, and bidirectional RNN has the highest execution time.

Sequence to sequence text generation is demonstrated in [5]. Bangla text generation is analysed using a deep learning approach. LSTM is used for analysing text sequences and next-word prediction. In this paper, Bangla Text was used. This model is in demand for this language because tools for Bangla are limited. No metrics are displayed in the article. Results are presented on two sentences example, which has a satisfactory accuracy rate.

Text mining and opinion mining are highly needed in practice. Sentiment analysis (opinion mining) is based on sentiment lexicon—a set of predefined keywords that correspond to a particular sentiment. This paper [6] proposed a new opinion mining method developed using hidden Markov models (TextHMMs) for text classification based on word sequence instead of a predefined sentiment lexicon. Text patterns are used for the representation of sentiment through ensemble TextHMMs. The semantic cluster represents hidden variables in TextHMMs based on the appear-

ance of the words. Next, determine the sentiment score of sentences with the help of fitted TextHMMs. LSA (latent semantic analysis) is used in the method and provides boosting and clusters of words. The evaluation shows that the model outperforms some existing techniques and can classify implicit opinions. The method is applied to existing reviews from the online market.

In paper [7], the hidden Markov models were used for text prediction for the Polish language. Model is using an input text to learn the potential letters' sequence. For input, punctuation should be removed, and only spaces are considered as separate letters. From the cleaned text, transition matrix is created. The transition matrix is filled by occurrence. All sequences are sorted based on their frequency in the input text to increase the program's efficiency. The model evaluation showed that HMM could be successfully used for word generation but still is not enough for creating whole sentences. The displayed method can also be used in speech recognition.

Unsupervised word embeddings become highly popular, and it appears in many applications. It leads to the question of whether similar methods can improve semantic representations of word sequences. In paper [8] is presented efficient and, at the same time, a simple unsupervised objective for training distributed terms of sentences. A model called Sent2Vec can be interpreted as an extension from C-BOW to a bigger sentence context. A combination of a sentence with the help of an unsupervised objective function optimizes sentence words within a model. This method works better than all other unsupervised competitors except for SkipThought on most benchmark tasks. But SkipThought vectors demonstrate low performance on sentence similarity tasks while the proposed method is state of the art for these evaluations on average. It points out the robustness of the provided sentence embeddings for general purpose showing the relevance of well-grounded and straightforward representation models compared to the models using deep architectures.

Big data technics are used for natural text processing too. Particularly, linear discriminant analysis (LDA) is used in [9].

OpenAI pretrained a neural network GPT based on the Transformer architecture on a large amount of text. Based on this development, Google created its own neural network BERT. They significantly improved the result by making the neural network bidirectional, in contrast to GPT. OpenAI increased its GPT by a factor of 10 at once and trained it on an even larger volume of text—on 8 million Internet pages (a total of 40 GB of text). The resulting GPT-2 network is currently the largest neural network, with an unprecedented number of parameters of 1.5 billion (the BERT had 340 million in the largest model and the standard BERT 110 million). As a result, GPT-2 was able to generate entire pages of connected text. The main limitation of GPT-2 is English supporting [10, 11].

To summarize, federated learning for next-word prediction was used and showed fair results. RNN, in particular, LSTM, also demonstrated reliable results. The Markov model was used for sentiment analysis and gain satisfactory

accuracy rate. But it should be highlighted that the Markov chain has sequential nature to be used in next-word prediction.

This paper is aimed at comparing two sequential models: LSTM and Markov model in Ukrainian next-word prediction.

The contribution of the paper is given as follows:

- (i) The dataset of Ukrainian poems with repeated patterns in texts is prepared. This allows to train models without Ukrainian language corpus
- (ii) The structure of long short-term memory (LSTM) neural network with chosen hyperparameters is chosen
- (iii) The Markov chain's parameters (frequency, probability, and transitions) are experimentally calculated
- (iv) The hybrid model as combination of LSTM and Markov chain is developed. The prediction accuracy for all models is evaluated manually. The prediction accuracy of the proposed hybrid model is the best. However, time complexity is the high

The paper is organized as follows. Materials and Methods describe the proposed approaches. Results and Discussion present the prepared dataset and experimental setup. Conclusions underline the main results and limitations of the work.

2. Materials and Methods

2.1. Recurrent Neural Networks: LSTM. Neural networks [12] are a collection of algorithms that is aimed at recognizing patterns. Their nature is very similar to the human brain. Machine perception, labelling, or clustering of source data helps to interpret sensory data. All real-world data (images, sound, text, or time series) must be converted into vectors for a neural network to recognize numerical patterns. Artificial neural networks (ANN) consist of many deeply interconnected processing elements (neurons) that collaborate to solve a problem.

ANN typically includes numerous processors that run parallel and are organized in tiers. The same as optic nerves get input in human visual processing first level gets the initial input. Each subsequent level accepts output from the level prefacing it, not from the raw input—similar to the neurons distant from the optic nerve that gets signals from those closest to it. The last level gives the system result.

Recurrent neural networks [13] are generalizations of a direct transmission neural network that has internal memory. The RNN is repetitive in nature because it performs the same function for each data entry, but at the same time, the current output depends on the previous calculation. After receiving the original data, it is copied and sent back to the periodic network. The decision is based on an analysis of the current input and the output from the previous input.

RNNs can use their internal state (memory) to process input sequences when direct communication neural net-

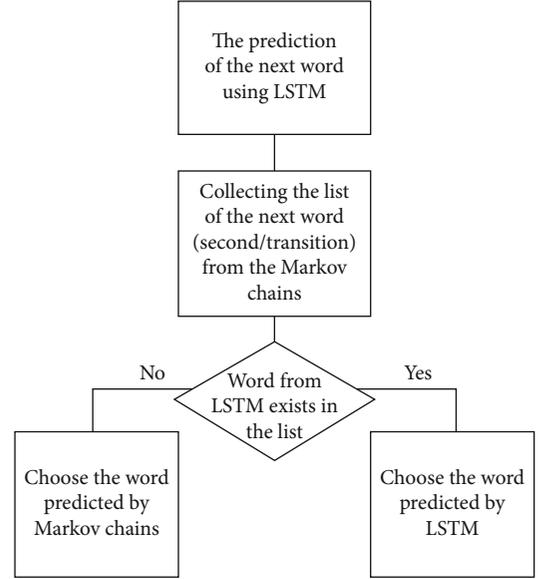


FIGURE 1: Hybrid model structure.

works cannot. The internal state helps them in duties such as speech recognition or unsegmented, connected handwriting recognition. Unlike RNN, other neural networks' inputs are independent of each other. All RNN inputs are interconnected.

First take $X(0)$ from the input sequence and then output $h(0)$, which together with $X(1)$ is the next step input. Therefore, $h(0)$ and $X(1)$ are the input data for the next step. Thus, $h(1)$ is the input from $X(2)$ for the next step and so on. In this way, he continues to memorize the context while learning.

The current state formula is the following:

$$h_t = f(h_{t-1}, x_t). \quad (1)$$

Application of the activation function:

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t), \quad (2)$$

where W is the weight, h is the single latent vector, W_{hh} is the weight in the previously latent state, W_{xh} is the weight in the current input state, \tanh is the activation function that implements nonlinearity that reduces activation to the range $[-1, 1]$.

Output:

$$y_t = W_{hy}h_t, \quad (3)$$

where y_t is the output state and W_{hy} is the output weight.

Advantages of the periodic neural network are as follows:

- (i) In RNN's data sequence, each sample is supposed to depend on the previous ones
- (ii) Extension of the effectiveness of neighbourhood of pixels is done by usage of convolutional layers



FIGURE 2: Wordcloud of original text.

Drawbacks of the periodic neural network are as follows:

- (i) Gradient problems of disappearance and explosion
- (ii) RNN training is a complicated task
- (iii) In the case of tanh or relu activation function models cannot process very long sequences

Long-term short-term memory networks (LSTMs) [14] are a modified version of periodic neural networks that make it easier to remember past data in memory. Here, the problem of the vanishing RNN gradient is solved. LSTM [15] is well suited for classifying, processing, and forecasting time series based on time lags of unknown duration and trains the model with backpropagation. There are three gates in the LSTM network:

- (1) Entrance gate: learn what value from the input should be used to modify the memory. The sigmoid function decides which values to pass through 0.1 and the tanh function provides a weighting to the values transmitted, determining the level of their importance in the range from -1 to 1:

$$\begin{aligned} i_t &= \sigma(W_i * [h_{t-1}, x_t] + b_i), \\ \tilde{C}_t &= \tanh(W_c * [h_{t-1}, x_t] + b_c). \end{aligned} \quad (4)$$

- (2) Forget gates: learn what parts should be thrown out of the block. The sigmoid function determines this. Consider the previous state (h_{t-1}) and the input con-

tent (x_t) and output a number from 0 (skip it) to 1 (save it) for each number in the state of cell C_{t-1} :

$$f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f). \quad (5)$$

- (3) Output gate: the input and memory of the unit are used to solve the output. The sigmoid function decides which values to pass through 0.1, and the tanh function provides a weighting to the transmitted values, determining their level of importance in the range from -1 to 1 and multiplying by the sigmoid output:

$$\begin{aligned} o_t &= \sigma(W_o[h_{t-1}, x_t] + b_o), \\ h_t &= o_t * \tanh(C_t). \end{aligned} \quad (6)$$

2.2. *Markov Chains*. Markov chains [16] are one of the most valuable classes of stochastic processes:

- (i) Supported by many sophisticated theoretical results but at the same time are flexible and simple
- (ii) Helpful to have a clue about random dynamic models
- (iii) Central to quantitative modelling by themselves

A stochastic matrix (or Markov matrix) [17] is an $n \times n$ square matrix P such that each element P is nonnegative and each row P has the sum 1. Each row P can be considered as a function of the probability mass for n possible results. It

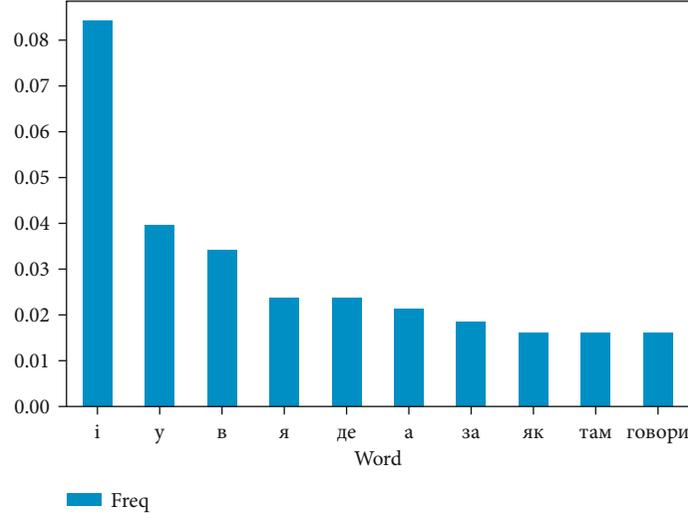


FIGURE 3: Frequency of words in original text.

Vocabulary size: 61
 Total sequences: 110
 Mode: "sequential_7"

Layer (type)	Output shape	Param #
embedding_7 (Embedding)	(None, 1, 10)	610
lstm_7 (LSTM)	(None, 50)	12200
dense_7 (Dense)	(None, 61)	3111

Total params: 15,921
 Trainable params: 15,921
 Non-trainable params: 0

FIGURE 4: Models' structure.

is easy to verify that if P is a stochastic matrix, then the k th power P_k is the same for all $k \in N$.

Markov chains are highly related to stochastic matrices. First, let S be a finite set with n elements $\{x_1, \dots, x_n\}$. The set S is called the state space, and x_1, \dots, x_n is the value of the state. A Markov chain $\{X_t\}$ on S is a sequence of random variables on S with a Markov property. So for any date t and any state $y \in S$,

$$P\{X_{t+1} = y | X_t\} = P\{X_{t+1} = y | X_t, X_t - 1, \dots\}. \quad (7)$$

In other words, knowledge of the current state is enough to know the probabilities for future states. In particular, the dynamics of the Markov chain are completely determined by a set of values:

$$P(x, y) := P\{X_{t+1} = y | X_t = x\} (x, y \in S). \quad (8)$$

By construction,

- (i) $P(x, y)$ is the probability of transition from x to y per unit time (one step)

TABLE 1: Model evaluation.

Loss	Accuracy
0.5750	0.7455

TABLE 2: The LSTM predicted results.

Word	Prediction
Крила (1 word)	крила має
Крила (4 words)	крила має а крила має
Немає (4 words)	Немає то буде воля немає

- (ii) $P(x, \cdot)$ is the conditional distribution of X_{t+1} , given $X_t = x$

We can consider P as a stochastic matrix, where

$$P_{ij} = P(x_i, x_j), \quad 1 \leq i, j \leq n. \quad (9)$$

Alternatively, if we take the stochastic matrix P , we can form a Markov chain $\{X_t\}$ as follows:

TABLE 3: The Markov chains' predicted results.

Initial word	Second word	Transitions
{'a': 0.020997, 'землі': 0.0026247, 'немає': 0.0052493, 'в': 0.03412073, 'живе': 0.00262467, 'вони': 0.002624671, 'у': 0.039370, 'або': 0.00262467, 'людина': 0.002624671...}	{'а': {'й': 0.125, 'як': 0.125, 'крила': 0.25, 'з': 0.125, 'вона': 0.125, 'твоє': 0.125, 'на': 0.125}, 'землі': {'немає': 1.0}, 'немає': {'поля': 0.5, 'пари': 0.5}, 'в': {'цьому': 0.07692307692307693, 'сяйві': 0.07692307692307693, 'далеких': 0.076923, 'гріб': 0.076923, 'одному': 0.076923, 'ступу': 0.076923, 'україні': 0.076923, 'семі': 0.076923, 'довгу': 0.076923, 'день': 0.3076923}, 'живе': {'на': 1.0}, 'вони': {'ті': 1.0}, ...}	{{('а', 'й'): {'правда': 1.0}, ('й', 'правда'): {'крилатим': 1.0}, ('правда', 'крилатим'): {'грунту': 1.0}, ('крилатим', 'грунту'): {'не': 1.0}, ('не', 'треба'): {'END': 1.0}, ('грунту', 'не'): {'треба': 1.0}, ('землі', 'немає'): {'то': 1.0}, ('немає', 'то'): {'буде': 1.0}, ('буде', 'небо'): {'END': 1.0}...}}

(i) Subtract X_0 from some specified distribution

(ii) For each $t = 0, 1, \dots$, find X_{t+1} with $P(X_t, \cdot)$

(ii) Punctuation removal

(iii) Tokenization (Python Natural Language Toolkit is used for this)

2.3. *Hybrid Model.* For achieving better results of the next-word prediction model, it was decided to develop a hybrid of LSTM and Markov chains.

The proposed algorithm for next-word prediction consists of the following (Figure 1).

3. Results

3.1. *Dataset Description.* As the work is aimed at developing personalized system in full cycle system, the model should be trained on user's input text. In such case, the base functionality of the system would be similar to T9 [18].

Each person has own linguistic style and uses the same repeated patterns in texts or messages.

Ukrainian poems contain a huge amount of different variations of phrase combinations. That is why such text corpus could be used as a training set for the developed prediction model. The specific group of Ukrainian poems [19] was used due to having no ability to gather the required type of dataset for training. Also, Ukrainian story was tested, but in story, there are not much repeated patterns which could help to train a model. Thus, the decision to use poems as a training set is substantiated by the fact that poems have more often used required repeated patterns for building prediction lines.

For the current investigation, a group of poems which consists of 1641 words was used. Based on this text corpus, it is possible to build up to 10 of phrase combinations [20–22].

3.2. *Data Preparation.* Although clearing the text affects the models' results, it is better to clean the text minimally because predicting the next word requires the original text.

The text before cleaning is demonstrated in Figures 2 and 3. The highest frequency has prepositions and connecting words. Regular cleaning process will remove them and predicted sentences could be incomplete.

So for data cleanup, it could be determined in the next steps:

(i) Lowering words

Stop-words removing, lemmatization, and stemming are not used in this phase, because the main idea is to save the structure of the sentence and to find consonant words.

3.3. Model Testing

3.3.1. *LSTM Model.* The figure below demonstrates the structure of the used LSTM (Figure 4).

The model consists of the embedding layer, LSTM, and neural layers. The embedding layer is highly important for NLP tasks as it helps to achieve better results focusing on keywords.

The training was based on Lina Kostenko's poem "Wings." Results of model evaluation on the validated dataset are displayed in Table 1.

Even though the text was small and the dimension of the dictionary was only 61 words, an accuracy of 75% was achieved, which is not bad for the start of the research. With increasing training text, greater accuracy will be achieved.

In the next step, the prediction on validated text is given.

Variants of one and several word prediction were tested (Table 2).

From the current demonstration, it is obvious that the prediction of the next word is correct.

3.3.2. *Markov Chain Model.* As the previous example showed, it is better to increase the sample; a collection of popular Ukrainian poems was used for training.

Below is a part of the trained model of Markov chains (Table 3).

The generated poem using Markov chain is given below (Table 4).

A graphic example of Markov chains on own text is shown in Figure 5.

3.3.3. *Hybrid Model.* The hybrid model was executed 100 times. The final model was generated Markov output only 44 times and combined output 56 times. The example of generated text is given in Table 5.

TABLE 4: The generated poem using Markov chain.

Ukrainian poem	Translated poem
людина нібито не літає...	A man supposedly does not fly...
стане початком тоді мій кінець	Will be the beginning, then my end
і чом твій усміх – для мене весела весна	And why your smile is a happy spring for me
гей ви зорі ясні тихий місяцю мій	Hey you stars of light, my quiet moon
немає поля то буде небо	No field, it will be heaven
зпоза хмар	Outside the clouds
їй виспівує осанна	Hosanna sings to her
говори зі мною	Talk to me
серце мліло не хотіло	The heart did not want to beat
будуть приходити люди	People will come

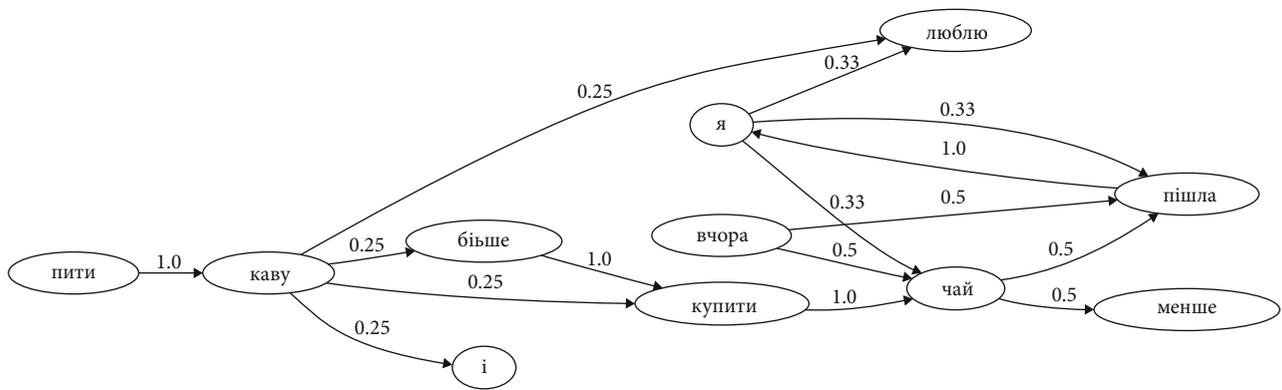


FIGURE 5: An example of the created model.

TABLE 5: Generated poem using the hybrid approach.

Output generated by 1—Markov and 2—LSTM	Text phrase
1	чом ваші очі сяють тим чаром
2	і тремтить райдуги в крилі
1	в день вольній новій
2	радощі й тугу нестимуть мені
1	так ніхто не кохав через тисячі літ
1	завтра на цій землі
2	його вистачить всім –
1	я на лиш знаю і одне засвоїв

TABLE 6: Generated poem using the hybrid approach.

Expert no.	The best model		
	LSTM	Markov chain	Hybrid
1	2	4	4
2	1	4	5
3	2	4	4
4	1	4	5
5	3	4	3
6	1	4	5

As shown from Table 6, the accuracy of the hybrid model is adequate.

The three models' comparison by time is given in Table 7.

3.3.4. *Model Comparison.* The content of generated texts was evaluated manually. Six students of Applied Linguistics Department at Lviv Polytechnic National University have evaluated the quality of generated text by three models. Unfortunately, GPT-2 adaptation to Ukrainian language is presented but it cannot be executed on a small dataset (<https://kaif.revo.ua/>). Each expert has analysed 10 texts generated by each model. The results of expert evaluation are given in Table 6.

4. Discussion

The possible improvements of the proposed model include

- (i) adding punctuation to the prediction
- (ii) adding suggestion of word form
- (iii) completing the system for a user with full functionality

TABLE 7: Comparison by time.

Model	Training time (s)	Execution time (s)
LSTM	70.53	0.38
Markov chains		0.01
Hybrid		0.31

The proposed algorithm is oriented on Ukrainian language. However, it can be retrained for other languages.

5. Conclusions

Word prediction is helpful for users because it can boost typing speed and help to omit errors. A personalized, predictive text input system is a relevant research topic for all languages, especially for Ukrainian, as currently products have limited tools which support the Ukrainian language.

Based on state-of-the-art research for analysis, three algorithms were chosen: LSTM, Markov chains, and hybrid. All algorithms are relevant for the next-word prediction task because both have a sequential nature (current output depends on previous). The Markov chains performed the task the fastest and qualitatively for development. Thus, it was chosen for the final outcome.

The novelty of the research is that, unlike T9, the developed system can generate not only the next word but also a few words, whole sentence, or several sentences.

Future improvement should be focused on punctuation and word form prepositions. Also, it will include a function for personalization of the system by developing a prediction model with users' linguistic style analysis.

Furthermore, provide a more accurate evaluation on marked data, currently working on their gathering.

Data Availability

The row data used to support the findings of this study have been deposited in the repository (10.6084/m9.figshare.14844654.v1).

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgments

This research was funded by National Research Foundation of Ukraine and Comenius University, Slovakia.

References

- [1] D. Anson, P. Moist, M. Przywara, H. Wells, H. Saylor, and H. Maxime, "The effects of word completion and word prediction on typing rates using on-screen keyboards," *Assistive Technology*, vol. 18, no. 2, pp. 146–154, 2006.
- [2] A. Hard et al., "Federated learning for mobile keyboard prediction," 2018, <https://arxiv.org/abs/1811.03604>.
- [3] T. Yang, G. Andrew, H. Eichner et al., "Applied federated learning: improving Google keyboard query suggestions," 2018, <https://arxiv.org/abs/1812.02903>.
- [4] S. Mangal, P. Joshi, and R. Modak, "Lstm vs. gru vs. bidirectional rnn for script generation," 2019, <https://arxiv.org/abs/1908.04332>.
- [5] M. S. Islam, S. S. Sharmin Mousumi, S. Abujar, and S. A. Hossain, "Sequence-to-sequence Bangla sentence generation with LSTM recurrent neural networks," *Procedia Computer Science*, vol. 152, pp. 51–58, 2019.
- [6] M. Kang, J. Ahn, and K. Lee, "Opinion mining using ensemble text hidden Markov models for text classification," *Expert Systems with Applications*, vol. 94, pp. 218–227, 2018.
- [7] G. Szymanski and Z. Ciota, "Hidden Markov models suitable for text generation," in *WSEAS International Conference on Signal, Speech and Image Processing (WSEAS ICOSSIP 2002)*, pp. 3081–3084, Corfu, Greece, 2002.
- [8] M. Pagliardini, P. Gupta, and M. Jaggi, "Unsupervised learning of sentence embeddings using compositional n-gram features," 2017, <https://arxiv.org/abs/1703.02507>.
- [9] G. T. Reddy, M. P. K. Reddy, K. Lakshmana et al., "Analysis of dimensionality reduction techniques on big data," *IEEE Access*, vol. 8, pp. 54776–54788, 2020.
- [10] J. S. Lee and J. Hsiang, "Patent claim generation by fine-tuning OpenAI GPT-2," *World Patent Information*, vol. 62, article 101983, 2020.
- [11] D. Ham, "End-to-end neural pipeline for goal-oriented dialogue systems using GPT-2," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 583–592, Seattle, WA, USA, 2020.
- [12] M. Antony and P. Bartlett, *Neural Network Learning: Theoretical Foundations*, Cambridge university press, 2009.
- [13] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative study of CNN and RNN for natural language processing," 2017, <https://arxiv.org/abs/1702.01923>.
- [14] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM neural networks for language modeling," in *13th Annual Conference of the International Speech Communication Association*, Portland, OR, USA, 2012.
- [15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [16] P. Gagniuc, *A Markov Chains: From Theory to Implementation and Experimentation*, John Wiley & Sons, 2017.
- [17] S. P. Meyn and R. L. Tweedie, *Markov Chains and Stochastic Stability*, Springer Science & Business Media, 2012.
- [18] Y. V. Krak, A. V. Barmak, R. A. Bagriy, and I. O. Stelya, "Text entry system for alternative speech communications," *Journal of Automation and Information Sciences*, vol. 49, no. 1, pp. 65–75, 2017.
- [19] "The most popular poems of Ukrainian poets known all over the world," https://maximum.fm/najpopulyarnishivirshi-ukrayynskih-poetiv-yaki-znayut-u-vsomu-sviti_n169157.
- [20] R. S. Bolia, W. T. Nelson, M. A. Ericson, and B. D. Simpson, "A speech corpus for multitaler communications research," *The Journal of the Acoustical Society of America*, vol. 107, no. 2, pp. 1065–1066, 2000.

- [21] S. Hakak, M. Alazab, S. Khan, T. R. Gadekallu, P. K. R. Maddikunta, and W. Z. Khan, "An ensemble machine learning approach through effective feature extraction to classify fake news," *Future Generation Computer Systems*, vol. 117, pp. 47–58, 2021.
- [22] P. Ashokkumar, G. Siva Shankar, G. Srivastava, P. K. R. Maddikunta, and T. R. Gadekallu, "A two-stage text feature selection algorithm for improving text classification," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 20, no. 3, pp. 1–19, 2021.