

# Making answer algorithm for chat-bot

Khrystyna Shakhovska

ISN Department, Lviv Polytechnic National University,  
UKRAINE, Lviv, 12 S. Bandery str., email:  
kristin.shakhovska@gmail.com

*Abstract –This paper explores pro and cons of existing chat-bots and investigate methods of their improvement. Particularly, is proposed usage of Rabin-Karp and Knut-Pratt algorithms for making answer to user and is demonstrated their effectivity.*

Keywords: chat-bot, hashing, prefix-function.

## I. Introduction

A robot or bot, as well as an internet bot, a www-bot, etc., is a special program that performs automatically and / or on a given schedule of any action through the same interfaces as an ordinary user. During the discussion of computer programs, mainly used for internet. Chat-bots take a significant part in our life. Therefore, we started exploring the most used bots and found a disadvantage. The aim of the paper is the recognition of user's gender for making chat-bot's answer more human likeness

The purpose of the work is to develop an algorithm for analyzing and parsing the user's text for automatically generating the response of the chat bot, taking into account the topics of correspondence and morphology of the text. The algorithm's work will be based on prefix function and hash function. Also, a comparison of the developed algorithm with the existing ones will be made.

## II. Existing bots

Bots which are using push-button interface to give the answer are the most common. However, the subject of our research is the bots that "understand" the natural language (with the nature language interface). The best online artificial intelligence (AI) online chats- Mitsuku, Rose, Poncho, Right Click, Insomno Bot, Dr. AI та Melody.

1. Mitsuku is one of the best bot in AI. Also is the current Laurel Laureate of Loebner. The Loebner Prize is an annual competition for methods of artificial intelligence, which defines intelligence that is most similar to human. Competition format is a standard Turing test [2]. This bot can talk about something, unlike other ones, done for a specific task.
2. Rose. The ChatBot, which won the Loebner Prize in 2014 and 2015 [3].
3. RightClick. Launches a program that creates websites. He asks general questions during the conversation: "What is the area of your interests?" and
4. "Why do you want to create a website?" Based on the analysis of the replies received, the chat bot creates custom templates. Quite adequately responsive to non-site-related topics [4].
5. Poncho is a meteorological specialist. He sends notifications at least twice a day with the user's consent and is smart enough to answer such questions as "Should I take an umbrella today?" [5].

6. Insomno Bot is designed for "night owls". This applies to all people with sleep problems. This bot has the ability to keep conversation on any topic [6].
7. Dr. AI bot asks for symptoms, body parameters and disease history, then lists the most and least likely causes of the symptoms and sorts them in order of severity [7].
8. Baidu melody. This application collects medical information about people, and then passes it to doctors in a form that facilitates their use for diagnostic purposes [8].

## III. Used methods

The prefix function [9, 10] of the string  $\pi(S, i)$  defines the length of the largest prefix of the string  $S[1..i]$ , which does not matches this line and at the same time is its suffix. It is the length of the longest beginning of a line, which is also its end. For line  $S$  it is convenient to represent a prefix function in the form of a vector of length  $|S|-1$ . You can consider the prefix-function of the length  $|S|$ , putting  $\pi(S, 1) = 0$ . Example of the prefix of the function for the line «abcdabcabcdab»:

$\pi(S[i])$	$a^a$	$b^a$	$c^a$	$d^a$	$a^a$	$b^a$	$c^a$	$a^a$	$b^a$	$c^a$	$d^a$	$a^a$	$b^a$	$c^a$	$d^a$	$a^a$	$b^a$
$\pi(S[i])$	0	0	0	0	1	2	3	1	2	3	4	5	6	7	4	5	6

Hashing [11] is converting an array of input data of arbitrary length (an array of rows) into (output) a bit string of fixed length executed by a certain algorithm. A function that implements an algorithm and performs a conversion is called a hash function or a convolution function. Output data is called an input array, key, or message. The result of conversion (output) is called Hashing is used in the following cases:

- for building associative arrays;
- to find duplicates in series of data sets;
- for constructing unique identifiers in data sets;
- to calculate checksums from the data (signal) for further detection of errors (which occurred accidentally or intentionally) that arise during storage and / or transmission of data;
- to store passwords in security systems in the form of a hash (for the password recovery for the hash code you need a function that is inversed to the used hash function);
- for the development of an electronic signature (in practice, not just the message, but its "hash-image") is signed, etc.

## IV. Proposed method

The algorithm of the chat bot will consist of searching for the keyword (words) of the conversation and the formation of the person's termination of the verb. We search for a keyword by means of hashing. To do this, we use the calculation formula:

$$h(S) = S[0] + S[1] * P + S[2] * P^2 + S[3] * P^3 + \dots + S[N] * P^N,$$

where  $P$  is a simple number. Choose  $P$ , which is approximately equal to the number of characters in the input alphabet. For example, if the strings are composed only of small Ukrainian letters, then the good choice will be  $P = 37$ . We use the Rabin-Karp algorithm to search the substring in the string for  $O(N)$ .

Let the text of user is  $T$  and line  $S$  in it, consisting of small Cyrillic letters, are given. It is necessary to find all occurrences of the line  $S$  in the text  $T$  for the time  $O(|S| + |T|)$ . The keyword search algorithm consists of the following steps:

1. Let's count the hash for line  $S$ .
2. Let's count the hash value for all prefixes of line  $T$ .
3. We will choose all subclasses  $T$  of the length  $|S|$ . Each of them can be compared with other lines of length  $|S|$  in time  $O(1)$ .

When the keyword has been found, you must define the individual verb ending. So, we try to find the verb as the word located before or after the keyword found. We test the ending using a prefix function: by Knutt-Pratt algorithm., which does not contain explicit line comparisons and is executed for  $O(n)$  actions. Here is an algorithm scheme:

1. We will count the value of the prefix function  $\pi[i]$  for  $i \in [1..n-1]$  ( $\pi[0] = 0$ ).
2. To calculate the current value  $\pi[i]$ , use the variable  $j$ , which indicates the length of the current sample considered. First  $j = \pi[i-1]$ .
3. We test a sample of length  $j$ , for which we compare the characters  $s[j]$  and  $s[i]$ . If they are the same, then we consider  $\pi[i] = j + 1$ ,  $i = i + 1$ . If the characters are different, then we reduce the length  $j$ , assuming it is equal to  $\pi[j-1]$ , and repeat this step of the algorithm from the beginning.
4. If we have reached the length of  $j = 0$  and so did not find the same characters, then stop the sampling process and consider  $\pi[i] = 0$ ,  $i = i + 1$ .

## V. Advantages of proposed method

Due to big complexity of well-known algorithm they are not as effective as we wish. According to this, we propose usage of Rabin-Karp and Knut-Pratt algorithms, due to their effectivity. In consequence of hashing we reduce a quantity of comparison which let the algorithm works faster. To add, we can find special endings with linear complexity. Results of research can be used not only for chat-bots and also for finding keywords in the text. An aggregated comparison of algorithms is given in Table 1. As you can see, the developed algorithm does not dominate only KMP-search in the case when we immediately find the required sample.

TABLE 1  
COMPARISON OF ALGORITHMS

Algorithm	Complexity
Developed algorithm	$O( S  +  T )$
Linear search	$O((T-S+1) * T)$
KMP-search	from $O( S  +  T )$ to $O((T-S+1) * T)$
LSA	$O(n^2 k^3)$ , $n= S   T $
SVM with tf-idf scheme	$O( Q  S  T )$

## Conclusion

After analyzing existing methods, we see that the complexity of the developed algorithm for finding keywords is less than of well-known algorithm. Using the prefix function to form a bot response allows you to work with Cyrillic texts. The proposed algorithm improves work of chat bots, which determines the gender of the interlocutor. This brings the bot closer to the level of human conversation.

## References

- [1] Shevat, Amir (2017). Designing bots: Creating conversational experiences (First ed.). Sebastopol, CA: O'Reilly Media. ISBN 9781491974827.
- [2] Mitsuku // [Internet source]. – Access mode: <http://www.mitsuku.com/>
- [3] Rose // [Internet source]. – Access mode: <https://www.robeco.nl/service-contact/index.jsp>
- [4] Right click// [Internet source]. – Access mode: <https://rightclick.io/#/>
- [5] Poncho // [Internet source]. – Access mode: <https://poncho.is/>
- [6] Insomnobot// [Internet source]. – Access mode: <http://insomnobot3000.com/>
- [7] Dr.A.I// [Internet source]. – Access mode: [https://www.healthtap.com/login?redirect\\_to=/symptoms](https://www.healthtap.com/login?redirect_to=/symptoms)
- [8] Baidu Melody's // [Internet source]. – Access mode: <http://research.baidu.com/baidus-melody-ai-powered-conversational-bot-doctors-patients/>
- [9] Kormen, T., Leyzerson, Ch., Ryvest, R., Shtayn, K. Alhorytm: postroenye y analiz = Introduction to Algorithms / Pod red. Y. V. Krasnykova. – 2-e yzd. – M.: Vyl'yams, 2005. – 1296 s. – ISBN 5-8459-0857-4.
- [10] Donald Knuth; James H. Morris, Jr, Vaughan Pratt (1977). «Fast pattern matching in strings». SIAM Journal on Computing 6 (2): 323–350. DOI:10.1137/0206024.
- [11] Donal'd Knut. Yskusstvo prohammyrovanyya. Tom 3. Sortyrovka y poysk = The Art of Computer Programming, vol.3. Sorting and Searching. – 2-e yzdanye. – M.: «Vyl'yams», 2007. – S. 824. – ISBN 0-201-89685-0.
- [12] Urvacheva, V. A. "Obzor metodov ynformatsyonnoho poyska." Vestnyk Tahanrohskoho ynstytuta ymeny AP Chekhova 1 (2016).
- [13] Landauer, Thomas K. Latent semantic analysis. John Wiley & Sons, Ltd, 2006.
- [14] Aizawa, Akiko. "An information-theoretic perspective of tf-idf measures." Information Processing & Management 39.1 (2003): 45-65.