Kazakh Chatbot Using Deep Learning

Nurlan Zhaniyar
Robotics and Mechatronics
Nazarbayev University
Astana, Kazakhstan
nurlan.zhaniyar@nu.edu.kz

Jabrail Chumakov
Robotics and Mechatronics
Nazarbayev University
Astana, Kazakhstan
jabrail.chumakov@nu.edu.kz

Zhansultan Keneskhan
Robotics and Mechatronics
Nazarbayev University
Astana, Kazakhstan
zhansultan.keneskhan@nu.edu.kz

Abstract—ChatBot can be described as software that can chat with people using artificial intelligence. Its industrial usage and main tasks are to quickly respond to users and provide them all the necessary information. Chatbots have been implemented in many different languages, however, there is still no open-source Kazakh-speaking bots. The creation of such a bot will help both to provide better customer service to Kazakh speaking users and will increase more interest in learning the Kazakh language itself. This paper investigates how to create a chatbot using Feed Forward Neural Network as well as it will provide some NLP algorithms for reading Kazakh letters and some functions to work with the text. Furthermore, it will be discussed where the bots can use the best and what are the current drawbacks that can be improved in the future.

Index Terms—Kazakh, Language, Chatbot, Encoding, Stemming

I. Introduction

In our culture, computers play a significant role in today's world. Computers provide us with information; they engage us in a lot of ways and support us. Now is the age of many web-based services such as E-business, entertainment, virtual support, and many more. In the field of online services, where everything is now connected with the web, there is a dramatic increase. It is a very user-friendly solution to user-computer use of everything. There are various customer service forms available, such as chat support and phone services. But it takes time for all such human-to-human support systems to address customer queries. The waiting time often rises as the number of customers increases, which results in low customer satisfaction.

The outline of natural and instinctive connection modalities is one of the important targets in the field of Human-Robot Interaction (HRI). Specifically, various initiatives have been devoted to developing frameworks to interact in a distinctive language with the customer. A chatbot is a computer designed to counterfeit a text or spoken area of intelligent communication. But this paper is only based on the Chatbot text. The user input is recognized by Chatbot as well as access information to provide a predefined acknowledgement by using pattern matching. "For instance, if the user provides the bot with a sentence such as “Сәлем” The chatbot would most likely answer something like, "Сәлем! Келгеніңізге рахмет" as the chatbot responds based on the user's pattern.

An answer from a predefined pattern is given to the user when the input is brought into existence in the database. Using pattern comparison, a chatbot is implemented in which the order of the sentence is remembered and a saved answer pattern acclimatizes to the exclusive variables of the sentence. They are unable to register and respond to complex questions and are unable to conduct compound operations. Chatbot technology is a relatively new one. The application of a Chatbot in the future can be seen in different fields. The methods used to develop and execute a Chatbot are covered in this paper. Comparisons are made, outcomes are discussed and the conclusion at the end is drawn.

A Chatbot refers to a chatting robot. It is a computer program that simulates communication. The interaction with the user is all about it. The interaction is really easy with a Chatbot. It responds to the questions the user asks. When creating a chatbot, the question is asked how will the user communicate with it? And it is very relevant how the conversation with the user and the Chatbot would be. Using diagrams, the architecture of a Chatbot is described as follows:

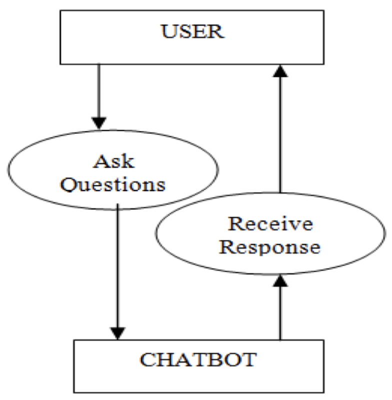


Fig. 1. Diagram of priciple of the chatbot

A. Selection of OS

Windows and macOS were used for this project due to the fact that they are user friendly, as well as robust.

B. Selection of Software

Pytorch software was used for programming in python. Because it contains basic workspace and it is mostly used for python applications.

C. Creating a chatbot

For creating a Chatbot, a program has to be written. Pytorch programming language is used for programming. The Chatbot is created in such a way to help the user with particular questions, improve the communication and amuse the user.

D. Creating a chat

Special patterns were created that is known to the user in order to easily understand.

E. Pattern Matching

It is deep learning used in the design of a Chatbot. The input is matched with the inputs saved in the database and the corresponding response is returned.

F. Conversational and Entertaining

The Chatbot responses are a way known to the user. The conversation follows a Basic Kazakh language and interacts in an easy to read manner. The conversation between the user and the Bot is entertaining. Chatbot tries to provide with funny and interesting responses.

Rest of the paper is organized as follows. Section II is an implementation of the chatbot, which is provides preprocessing and architecture of the work. Section III briefs about existing results. Section IV discussion of the project and finally section V concludes the paper.

II. Implementation

A. Tokenization

Tokenization is one of the most common functions in the Natural Language Tool Kit. It separates the sentence into small blocks called tokens. The tokens can be designed in three ways:

- Word tokens: Білім-арзан-білу-қымбат
- Subword tokens: Бі-лім-ар-зан-бі-лу-қым-бат
- Character tokens: Б-і-л-і-м-а-р-з-а-н-б-і-л-у-қ-ы-м-б-а-т

Tokenization is a significant tool since it breaks the raw text into an array of these blocks. In our case, we used default delimiter as space, so the tokens are the words.

Here is an example:

- Нан — тамақтың атасы, Ынтымақ — көптің батасы.

Which will be processed to

- [Нан, —, тамақтың, атасы, , , Ынтымақ, —, көптің, батасы, .]

B. Simplification and Punctuation Removal

The next step of processing the input is getting rid of unnecessary punctuation symbols. So the input converges into this:

- [Нан, тамақтың, атасы, Ынтымақ, көптің, батасы]

Our additional unique feature is changing all Kazakh letters to their Cyrillic analogues. It just runs through the input and whenever it detects the letter it quickly changes to the following substitute:

- ә->а
- і->и
- ң->н
- ғ->г
- ү->у
- ұ->у
- қ->к
- ө->о
- һ->х

On one hand, this feature makes the bot more user friendly since nowadays there are not many people who use Kazakh keyboard while texting the message. On the other hand, if the bot understands only entirely grammatical messages, people will eager to write more correctly. For this reason, this feature is left as an option.

C. Stemming

The next valuable step of input processing is stemming. It is the process of reducing variation in words to their root forms such as collecting a group of words to the same stem even if the stem itself is not a valid word in the Language. Since it might hard for the machine to understand that connect, connection, connecting are actually different forms of one word, there is a crucial need in a function that will tell the machine that all these words are connect. There is also another term called lemmatization. The difference between these two is that lemmatization tries to reduce the words to an existing root, while stemming does not care about that and might cut some letters of the root. Example of it can be seen on the Figure 2.

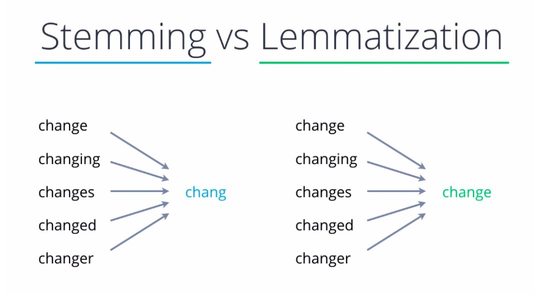


Fig. 2. Stemming vs Lemmatization

- Lemmatization:
  (change,changed,changing) transforms to (change,change,change)

- Stemming:
  (change,changed,changing) transforms to (chang,chang,chang)

There are several algorithms among the English stemmers: Porter, Lancaster, Snowball. There is a possibility of choosing other languages in snowball stemmer. However, currently, there is no option for the Kazakh language. For that reason, we created our own.

Firstly, at least some part of the stem must be detached, so the function will no cut some parts of stem similar to suffixes or endings. The approach was partially copied from Russian stemming, so the region from the beginning till the first non-vowel after the first vowel was isolated. The second part of the word is translated to bytes, using 8-Bit Universal Character Set Transformation Format(UTF-8) encoder. After that, a special list of endings and suffixes passes through the word. When the ending is detected inside the word the function deletes it from the word only if it is really in the end. Often inside the word, there might appear letter sequences that are similar to the endings. Their removal might cause some troubles, so it is necessary to put the condition that ending must be detected in the end.

D. Final Preparation

During the training process, all unique stemmed words are sent to one big array of words. Later, during the chatting the input message will undergo the same process, but the final stemmed words are going to be compared with the main array, so this is how it is going to look:

- "сәлем, хал қалай?"
- [сәлем, таң, қайырлы,кеш, қалай, сау, бол]

Since it has 2 of 7 the array words the output is going to look this way:

- [1,0,0,0,1,0,0]

This vector is ready to be sent to the first layer of the Neural Network.

E. Neural Network Structure

Feed-Forward Neural Network model was chosen among the other artificial networks. This is probably the first and most simple Neural Network that was designed. The information here moves only in one direction: from the input layer to output through some hidden layers. The whole concept of Neural Networks with layers can be seen in figure 3. It is often used in supervised learning, the learning process with the given correct output. In our case, we needed the network to work as a classifier. Since the single-layer network is too simple and can separate only linear patterns, that would not help us to differentiate many different topics of conversation. For this reason, we found 2 hidden layers as an optimal value for the bot. Adding more would cause over-fitting or bigger computational expenses not showing better results.

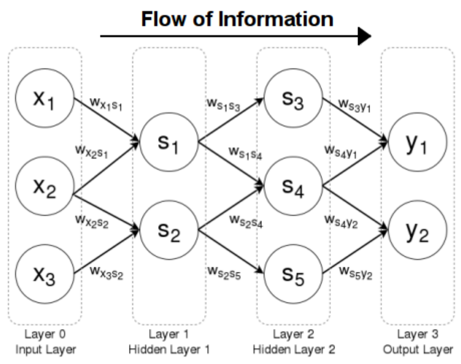


Fig. 3. Feed Forward Neural Network

F. Neural Network Training

Initially, we feed our model with different sentences assigned to some tag. Tags can be considered as the topic of conversation. During the training, each line between the layers starts to gain some weight. The more frequently the word was mentioned in one tag the higher weight it has. with each sentence, the model assigns the new unique stemmed words, which did not appear before, to the new input unit inside the first layer. By the end of the training, the first layer will have as many units as the number of unique stemmed words among all training set. The number of units in two hidden layers was set to 8 and the final layer will have the same number of units as the number of tags.

G. Neural Network Chatting

Once the message is sent to the bot it starts to process it following all steps written in the implementation part. The whole process can be checked in Figure 4. Then after it reaches the final form it is sent to the network model, wherein the final layer unit with the highest probability will be chosen as a subject of a conversation. After that, it will randomly choose the hard-coded response assigned to the topic(tag).

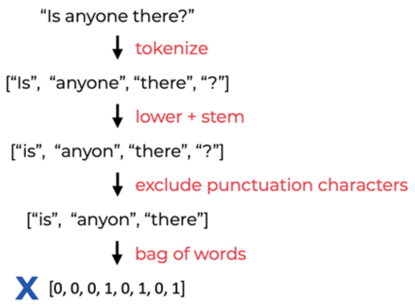
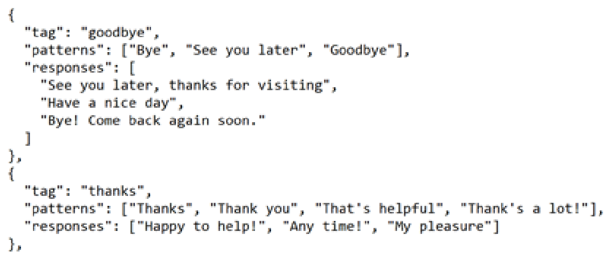

"Is anyone there?"
tokenize
["Is", "anyone", "there", "?"]
lower + stem
["is", "anyon", "there", "?"]
exclude punctuation characters
["is", "anyon", "there"]
bag of words
X [0, 0, 0, 1, 0, 1, 0, 1]

Fig. 4. JSON File

H. Data Structure

All tags with assigned sentences or patterns as well as responses are stored in one .json file. A JSON file is the key-value data format file where which is commonly used for data storage. A simple example can be seen in Figure 4.

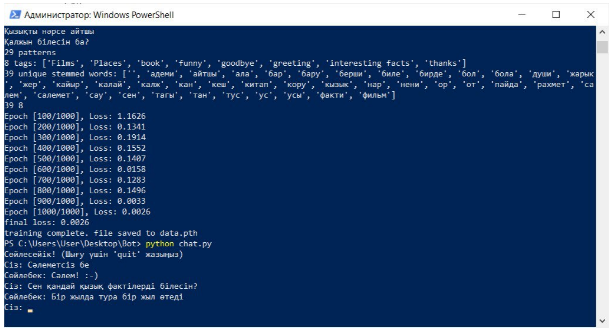


```
{
  "tag": "goodbye",
  "patterns": ["Bye", "See you later", "Goodbye"],
  "responses": [
    "See you later, thanks for visiting",
    "Have a nice day",
    "Bye! Come back again soon."
  ]
},
{
  "tag": "thanks",
  "patterns": ["Thanks", "Thank you", "That's helpful", "Thank's a lot!"],
  "responses": ["Happy to help!", "Any time!", "My pleasure"]
},
```

Fig. 5. JSON File

III. Results

As a result of creating the chatbot, it was found that it is possible to use features such as tokenization and stemming, which help the bot to ignore punctuation signs and understand words more easily (for example, when reading the same root words). In addition, a function has been added to the bot that allows you to correctly understand words written in Russian letters. This is necessary in order to avoid installation of the Kazakh keyboard layout in case of its absence, as well as to answer the user's question if he made a grammatical mistake while writing the phrase. It should be noted that the chatbot trains based on the available dataset. Thus, it learns the phrases and words that are present there and learns to correctly perceive them in different contexts. The result of the bot's performance can be seen in Figure 6.

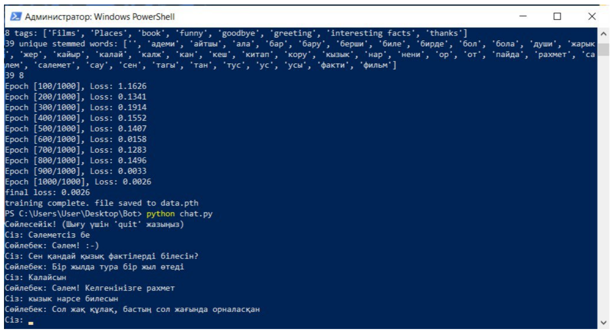


Администратор: Windows PowerShell

```
Қызықты нәрсе айтшы
Қалжын білесін ба?
29 patterns
8 tags: ['Films', 'Places', 'book', 'funny', 'goodbye', 'greeting', 'interesting facts', 'thanks']
39 unique stemmed words: ['', 'адеми', 'айтшы', 'ала', 'бар', 'бару', 'берши', 'биле', 'бирде', 'бол', 'бола', 'души', 'жарык
', 'жер', 'кайыр', 'калай', 'калж', 'кан', 'кеш', 'китап', 'кору', 'кызык', 'нар', 'нени', 'ор', 'от', 'пайда', 'рахмет', 'са
лем', 'салемет', 'сау', 'сен', 'тагы', 'тан', 'тус', 'ус', 'усы', 'факти', 'фильм']
39 8
Epoch [100/1000], Loss: 1.1626
Epoch [200/1000], Loss: 0.1341
Epoch [300/1000], Loss: 0.1914
Epoch [400/1000], Loss: 0.1552
Epoch [500/1000], Loss: 0.1407
Epoch [600/1000], Loss: 0.0158
Epoch [700/1000], Loss: 0.1283
Epoch [800/1000], Loss: 0.1496
Epoch [900/1000], Loss: 0.0033
Epoch [1000/1000], Loss: 0.0026
final loss: 0.0026
training complete. file saved to data.pth
PS C:\Users\User\Desktop\Bot> python chat.py
Сөйлесейік! (Шығу үшін 'quit' жазыңыз)
Сіз: Сәлеметсіз бе
Сөйлебек: Сәлем! :-)
Сіз: Сен қандай қызық фактілерді білесін?
Сөйлебек: Бір жылда тура бір жыл өтеді
Сіз:
```

Fig. 6. Working process

It should be noted that the bot perfectly perceives information and can easily distinguish which tag to classify the entered input. For instance, in case of the presence of an error in the sentence, the bot will understand the closest word similar to the entered one and replace it. Thus, it will give a response in the correct way. Figure 7 shows how it actually works.

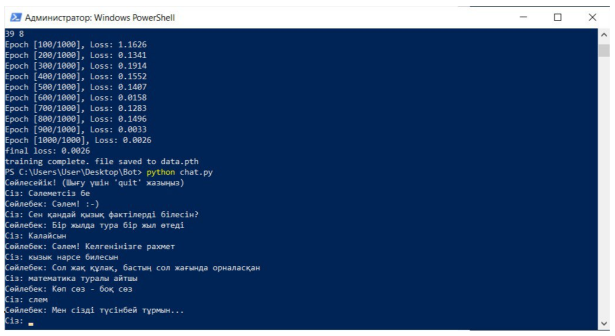


Администратор: Windows PowerShell

```
8 tags: ['Films', 'Places', 'book', 'funny', 'goodbye', 'greeting', 'interesting facts', 'thanks']
39 unique stemmed words: ['', 'адеми', 'айтшы', 'ала', 'бар', 'бару', 'берши', 'биле', 'бирде', 'бол', 'бола', 'души', 'жарык
', 'жер', 'кайыр', 'калай', 'калж', 'кан', 'кеш', 'китап', 'кору', 'кызык', 'нар', 'нени', 'ор', 'от', 'пайда', 'рахмет', 'са
лем', 'салемет', 'сау', 'сен', 'тагы', 'тан', 'тус', 'ус', 'усы', 'факти', 'фильм']
39 8
Epoch [100/1000], Loss: 1.1626
Epoch [200/1000], Loss: 0.1341
Epoch [300/1000], Loss: 0.1914
Epoch [400/1000], Loss: 0.1552
Epoch [500/1000], Loss: 0.1407
Epoch [600/1000], Loss: 0.0158
Epoch [700/1000], Loss: 0.1283
Epoch [800/1000], Loss: 0.1496
Epoch [900/1000], Loss: 0.0033
Epoch [1000/1000], Loss: 0.0026
final loss: 0.0026
training complete. file saved to data.pth
PS C:\Users\User\Desktop\Bot> python chat.py
Сөйлесейік! (Шығу үшін 'quit' жазыңыз)
Сіз: Сәлеметсіз бе
Сөйлебек: Сәлем! :-)
Сіз: Сен қандай қызық фактілерді білесін?
Сөйлебек: Бір жылда тура бір жыл өтеді
Сіз: Калайсын
Сөйлебек: Сәлем! Келгенінізге рахмет
Сіз: кызык нарсе билесын
Сөйлебек: Сол жақ құлақ, бастың сол жағында орналасқан
Сіз:
```

Fig. 7. Russian to Kazakh letters replacing

Also, in case if the user enters a phrase with error in the root or is not exist in the any availabe tag, it will answers that cannot recognize the input phrase. This happens for the reason that stemming was used in the bot. That means that the root of the word must be spelled correctly, or at least in Russian. Otherwise, the bot will perceive the phrase as missing in its dataset. An example of this can be seen in Figure 8.



```
Администратор: Windows PowerShell
39 8
Epoch [100/1000], Loss: 1.1626
Epoch [200/1000], Loss: 0.1341
Epoch [300/1000], Loss: 0.1914
Epoch [400/1000], Loss: 0.1552
Epoch [500/1000], Loss: 0.1407
Epoch [600/1000], Loss: 0.0158
Epoch [700/1000], Loss: 0.1283
Epoch [800/1000], Loss: 0.1496
Epoch [900/1000], Loss: 0.0033
Epoch [1000/1000], Loss: 0.0026
final loss: 0.0026
training complete. file saved to data.pth
PS C:\Users\User\Desktop\Bot> python chat.py
Сөйлесейік! (Шығу үшін 'quit' жазыңыз)
Сіз: Сәлеметсіз бе
Сөйлебек: Сәлем! :-)
Сіз: Сен қандай қызық фактілерді білесің?
Сөйлебек: Бір жылда тура бір жыл өтеді
Сіз: Калайсын
Сөйлебек: Сәлем! Келгеніңізге рахмет
Сіз: кызык нарсе билесын
Сөйлебек: Сол жақ құлақ, бастың сол жағында орналасқан
Сіз: математика туралы айтшы
Сөйлебек: Көп сөз - бөқ сөз
Сіз: слем
Сөйлебек: Мен сізді түсінбей тұрмын...
Сіз:
```

Fig. 8. Phrase that isn't in any available tag

IV. Discussion

The chatbot still has a big way to improve. The first thing to notice is the stemming algorithm. The Kazakh language has a lot of tricky moments that are not similar to any language present in current Natural Language Processing stemming tools. One word might have 3 or even 4 endings. An array of endings, which can be seen in the appendix, was done intuitively. Some endings appeared twice since they can be placed both before and after other endings in different words. Endings that are usually added to the word first, like plural ones(лар,лер,дар,дер,тар,тер) were put near the end of the array because they need to be removed last. The last goes suffixes since they appear before all endings. So the algorithm for word cutting is quite simple but still effective, however further improvements with specialists need to be done. One of the possible implementations is to let the word go through the array several times however it did not show much better results.

Another big question that is still quite challenging for us is how will the bot create the sentences on its own. So far we got only to the moment where the bot understands what the user asks from him and replies to him with the prepared answer. This type of bot is very useful when people enter some website and need to find something immediately. The chatbot will pop up and by understanding the question will give advice or send a proper link. So it is a really good alternative to the call center and might be helpful to the national EGOV page, Kaspi bank app, or any delivery/shop webpages.

V. Conclusion

For web services and systems such as scientific, entertainment, commercial systems, and academia, deep learning conversational agents are becoming popular. By querying missing data from the user to provide a satisfactory response, so human-robot interaction starts becoming a more efficient way. A chatbot is one of the easy ways to transport data from a computer without having to worry about proper keywords to look up information in a search or visit several web pages to collect information; users can quickly type their query and retrieve information in the Kazakh language. In this paper, details on the chatbot design and implementation were presented. A chatbot is a great user interaction tool for fast interaction. They support us by offering entertainment, saving time, and answering hard-to-find questions. The Chatbot must be simple and conversational. Since there are many designs and methods for creating a chatbot, commercial considerations can be at odds. Researchers need to communicate and should agree on a common approach to the development of a Chatbot. In this project, we examined the creation of chatbots and the applications of chatbots in different fields. The general purpose was that a chatbot must be simple, user-friendly, easy to understand, and compact in the knowledge base. Although chatbot technology still developing and it is necessary to make improvements to find a common approach to the design of a Chatbot.

References

[1] J. McGonagle, J. A. García, and S. Mollick, "Feedforward Neural Networks," Brilliant Math amp; Science Wiki. [Online]. Available: https://brilliant.org/wiki/feedforward-neural-networks/. [Accessed: 01-Dec-2020].

[2] E. Loper and S. Bird, NLTK: the Natural Language Toolkit. S.l.: S.n.

[3] KeithxKeithx 2 and Nathan SmithNathan Smith 64511 gold badge1111 silver badges2121 bronze badges, "SnowballStemmer for Russian words list," Stack Overflow, 01-Oct-1966. [Online]. Available: https://stackoverflow.com/questions/45696028/snowballstemmer-for-russian-words-list. [Accessed: 01-Dec-2020].

[4] "nltk.stem package¶," nltk.stem package - NLTK 3.5 documentation. [Online]. Available: https://www.nltk.org/api/nltk.stem.html. [Accessed: 01-Dec-2020].

[5] Tartarus, Russian stemming algorithm. [Online]. Available: http://snowball.tartarus.org/algorithms/russian/stemmer.html. [Accessed: 01-Dec-2020].

[6] P. Loeber, pytorch chatbot, 2020. [Online]. Available: https://github.com/python-engineer/pytorch-chatbot. [Accessed: 01-Dec-2020].

Appendix
Array of endings

['шама', 'шеме', 'шалық', 'шелік', 'дайын', 'дейін', 'тайын', 'тейін', 'мын', 'мін', 'бын', 'бін', 'пын', 'пін', 'айын', 'ейін', 'мыз', 'міз', 'быз', 'біз', 'пыз', 'піз', 'сың', 'сің', 'сыңдар', 'сіңдер', 'сыз', 'сіз', 'сыздар', 'сіздер', 'ыңыз', 'іңіз', 'сы', 'сі', 'дан', 'ден', 'тан', 'тен', 'нан', 'нен', 'да', 'де', 'та', 'те', 'да', "із", 'де', 'ған', 'ген', 'қан', 'кен', 'ға', 'ге', 'қа', 'ке', 'на', 'не', 'дың', 'дің', 'тың', 'тің', 'ның', 'нің', 'ды', 'ді', 'ты', 'ті', 'ны', 'ні', 'ін', 'дікі', 'тікі', 'нікі', 'ба', 'бе', 'па', 'пе', 'ма', 'ме', 'бен', 'пен', 'мен', 'лы', 'лі', 'ғы', 'гі', 'қы', 'кі', 'дай', 'дей', 'тай', 'тей', 'дық', 'дік', 'тық', 'тік', 'лық', 'лік', 'паз', 'ғыш', 'гіш', 'қыш', 'кіш', 'шек', 'шақ', 'шыл', 'шіл', 'нші', 'ншы', 'ау', 'еу', 'дап', 'деп', 'тап', 'теп', 'лап', 'леп', 'даған', 'деген', 'таған', 'теген', 'лаған', 'леген', 'ла', 'ле', 'дар', 'дер', 'тар', 'тер', 'лар', 'лер', 'ар', 'ер', 'ғар', 'гер', 'қар', 'кер', 'дыр', 'дір', 'тыр', 'тір', 'ғыз', 'гіз', 'қыз', 'кіз', 'атын', 'етін', 'йтын', 'йтін', 'ушы', 'уші', 'ып', 'іп', 'май', 'мей', 'ғалы', 'гелі', 'қалы', 'келі', 'ша', 'ше', 'лай', 'лей', 'сын', 'сін', 'са', 'се', 'бақ', 'бек', 'пақ', 'пек', 'мақ', 'мек', 'йын', 'йін', 'йық', 'йік', 'тым', 'тім', 'дым', 'дім', 'ым', 'ім', 'ім', 'ың', 'ің', 'ын', 'ін', 'ыңыз', 'іңіз', 'дар', 'дер', 'тар', 'тер', 'лар', 'лер', 'ар', 'ер', 'дас', 'дес', 'тас', 'тес', 'лас', 'лес',]