Savchenko L. V. Fenomen etnokodiv duhovnoi kultury u frazeologii ukrainskoi movy [The phenomenon of ethnic codes of spiritual culture in the phraseology of Ukrainian language] / L. V. Savchenko // FILOLOGICHNI STUDIYI. Vyp. 9, Ch. 2. 204 s. [in Ukrainian].

Alina Teletska
Vasyl' Stus Donetsk National University
Vinnytsia
Research Supervisor: V. I. Kalinichenko, PhD in Philology, Ass. Prof.
Language Advisor: V. I. Kalinichenko, PhD in Philology, Ass. Prof.

AUTOMATIC PARSING SYSTEM OF USER STORIES

Introduction. Modern society is experiencing a rapid development of the digital world. Nowadays, the issue of human-machine communication is considered rather topical. It is necessary to transform the natural, human-understandable language into a formalized form in order for the computer system to recognize the requests given by the user. This process is carried out due to the work of a linguistic processor which creates a model of the language system.

In this piece of research I have used the high-level general-purpose programming language – *Python* to implement the software. In addition, the NLTK library was chosen to fulfill its tasks. *Natural Language Toolkit* is a set of linguistically oriented modules in Python. NLTK is a collection of executable scripts designed to process text files [1: 119-153].

Review of recent publications. This paper is based on scientific works about automatic syntax text analysis of such Ukrainian and foreign scholars as Volodymyr Voloshyn, Tetiana Hriaznukhina, Nataliia Darchuk, Noam Chomsky, Daniel Yurafsky, James Martin and oth.

Objectives of the paper. The piece under consideration aims at developing a software application that will perform an automatic syntax-level check of *User Stories* based on the work of the NLTK library.

There can often be misunderstanding between the user and the performer. Differences in knowledge across industries are a problem of communication. In order to be successful, they use *User Stories*, statements that briefly describe the client's need and provide a quick way for the contractor to address a particular issue. Another advantage of these constructions is the ability to collect a large number of requests, classify them and prioritize their execution.

The most common format for writing is:

As a <role>, I want <goal> for <reason> [2].

The work of the system of automatic parsing "User Stories" consists of 4 blocks:

1. Filling in the template form:
   - Displaying of the sentence to the user:

Welcome to the program that will help in formulating the User Story.

User Story has the following structure:
As a <role>, I want <goal> for <reason>.

– Writing a *User Story.*

2. Sentence analysis:

– loading of the database of marked words of the morphological level;
– segmentation written by *User Story*;
– recognition of tokens in the sentence;
– assign appropriate codes to the database;
– connection building between tokens in a sentence [3].

Our database consists of a set of entities with the following structural parts:

• The codes of grammatical categories. In our database, the following categories play a role in the analysis: gender, number, case, person, time, or form (for verbs).

• Combinations of grammatical categories. Each of them is specific to a certain word form.

• Entities that contain word lists of the most used words to build *User Stories* by a user in IT sphere. Each word was assigned a corresponding entity code of the second block.

3. Check writing syntax:

- search of the corresponding word form in the database;
- transferring word code from the base to the tokens of the syntactic structure;
- building connections between words in the sentence and checking its accuracy.

4. Output of the analysis results.

After pressing the button, the user receives a response depending on the result – "Everything is correct!" or "Incorrect text input!".

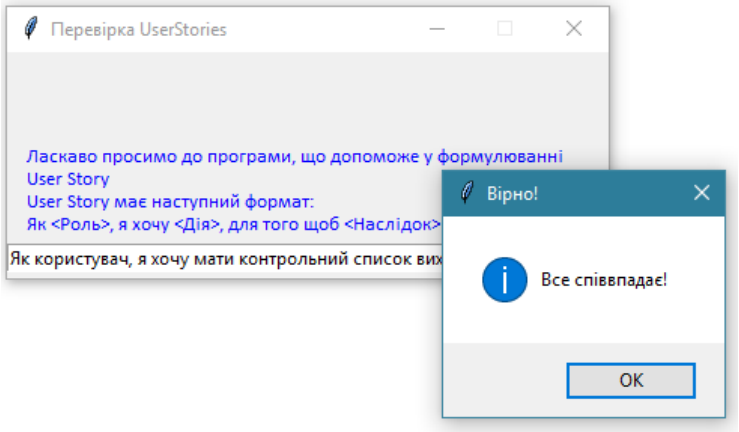

Figure 1. Result – "Everything is correct!"

At the next stage of software development, the verification of words in the database was added.

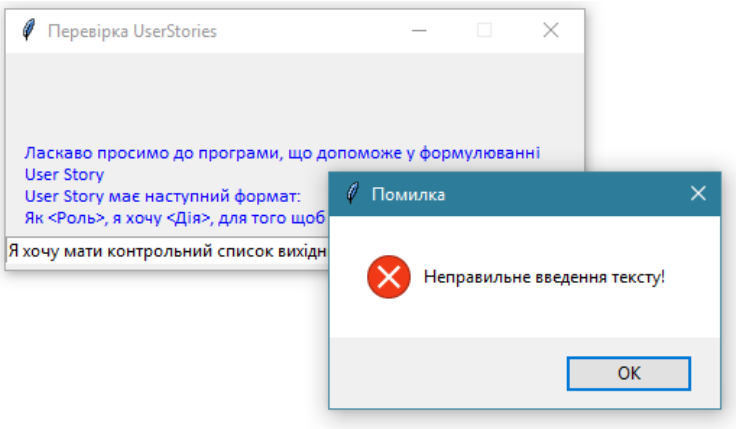

Figure 2. Result – "Incorrect text input!"

Results of the research. We have analyzed the quality of the software application, the text processing and highlighted the prospects for its development.

An algorithm must be developed to create a list of tokens missing from the database. The developer will be able to analyze it and find the words that can expand the base among the wrong word forms. This way the program will work more precisely with a focus on active vocabulary.

Conclusion. Further research may be related to creating an application which will replace the wrong wordform with the correct one. To do this, we need to add to the database all forms of available tokens, forming a complete paradigm for each word.

Another option for writing syntax is machine learning. Machine learning algorithms require samples of a huge amount of *User Stories*, so that we should create a system that can collect these syntax constructions. As a result, one can expect a new, improved training data base to be developed.

References

1. Jurafsky D. Speech and Langauge Processing [Electronic resource] / D. Jurafsky, J. Martin., 2000. Retrieved from: https://s3.amazonaws.com/academia.edu.documents/26250409/j00-006.pdf?response-content-disposition=inline%3B%20filename%3DSpeech_and_language_processing

2. Krasadakis G. How (and Why) to Write Great User Stories [Electronic resource] / G. Krasadakis. Retrieved from: https://www.freecodecamp.org/news/how-and-why-to-write-great-user-stories-f5a110668246/.

3. Geitgey A. Natural Language Processing is Fun! [Electronic resource] / A. Geitgey. Retrieved from: https://medium.com/@ageitgey/natural-language-processing-is-fun-9a0bff37854e.

Anastasiia Vdovychenko
Vasyl' Stus Donetsk National University
Vinnytsia