RUDN Journal of Language Studies, Semiotics and Semantics
2313-2299 (print), ISSN 2411-1236 (online)
Вестник Российского университета дружбы народов.
Серия: ТЕОРИЯ ЯЗЫКА. СЕМИОТИКА. СЕМАНТИКА

2024 Vol. 15 No. 1 195–210
http://journals.rudn.ru/semiotics-semantics

DOI: 10.22363/2313-2299-2024-15-1-195-210
EDN: FKVAOI
UDC [811.112.2:811.111]'42:004.9

Research article / Научная статья

Natural Language Processing and Fiction Text: Basis for Corpus Research

Alexey I. Gorozhanov[1] ✉, Innara A. Guseynova[1], Darya V. Stepanova[2]

[1]Moscow State Linguistic University, *Moscow, Russian Federation*
[2]Minsk State Linguistic University, *Minsk, Belarus*
✉ a_gorozhanov@mail.ru

Abstract. The study deals with NLP procedures on the material of the fiction texts in German and in English, which are considered as strong cultural texts. The aim of the study is to develop a model of such a technical device to process, analyze and interpret a fiction text, which would reveal the full potential of popular NLP tools within the corpus approach. The general methods used in the study are analysis and synthesis. Special methods are additionally used to solve certain specific issues: descriptive method, modelling and qualitative and quantitative analysis. The scientific novelty lies in the fact that the authors apply the crucial principles of the classical theories of text interpretation according to the latest methods and tools of the applied linguistics. As a practical result, special software has been developed, which is able to process SQL based linguistic corpora, automatically built with spaCy NLP library and Python programming language. This software can be used for a fiction text interpretation, as well as for compiling learning materials in Home Reading. It is assumed that the development of special software for strong cultural texts stimulates the search for scientific solutions and at the same time allows one to understand the essential differences that exist between natural and artificial intelligence.

Keywords: natural language processing, fiction text, linguistic corpus, F. Kafka, J. London, applied linguistics, spaCy

Article history:
Received: 01.09.2023
Accepted: 15.12.2023

For citation:
Gorozhanov, A.I., Guseynova, I.A. & Stepanova, D.V. (2024). Natural Language Processing and Fiction Text: Basis for Corpus Research. *RUDN Journal of Language Studies, Semiotics and Semantics,* 15(1), 195–210. https://doi.org/10.22363/2313-2299-2024-15-1-195-210

© Gorozhanov A.I., Guseynova I.A., Stepanova D.V., 2024
This work is licensed under a Creative Commons Attribution 4.0 International License
https://creativecommons.org/licenses/by-nc/4.0/legalcode

Обработка естественного языка и художественный текст: база для корпусного исследования

А.И. Горожанов[1], И.А. Гусейнова[1], Д.В. Степанова[2]

[1]Московский государственный лингвистический университет, *Москва, Российская Федерация*
[2]Минский государственный лингвистический университет, *Минск, Республика Беларусь*
a_gorozhanov@mail.ru

Аннотация. Рассматриваются процедуры обработки естественного языка (NLP) на материале художественных текстов на немецком и английском языках, которые рассматриваются как сильные культурные тексты. Целью исследования является разработка модели такого инструмента обработки, анализа и интерпретации художественного текста, который раскрывал бы весь потенциал популярных инструментов NLP в рамках корпусного подхода. Общими методами, используемыми в исследовании, являются анализ и синтез. Для решения отдельных задач дополнительно применяются специальные методы: описательный метод, моделирование и качественно-количественный анализ. Научная новизна заключается в том, что авторы совмещают основополагающие принципы «классической» теории интерпретации текста и новейшие методы и инструменты прикладной лингвистики. В результате было разработано специальное программное обеспечение, способное работать с лингвистическими корпусами на основе баз данных SQL, автоматически построенными с помощью библиотеки spaCy и языка программирования Python. Созданное приложение можно использовать для интерпретации художественного текста, а также для составления учебных материалов для дисциплины «Домашнее чтение». Предполагается, что разработка специального программного обеспечения для сильных культурных текстов стимулирует поиск научных решений и в то же время позволит понять существенные различия, существующие между естественным и искусственным интеллектом.

Ключевые слова: обработка естественного языка, художественный текст, лингвистический корпус, Ф. Кафка, Дж. Лондон, прикладная лингвистика, spaCy

История статьи:
Дата поступления: 01.09.2023
Дата приема в печать: 15.12.2023

Для цитирования:
Gorozhanov A.I., Guseynova I.A., Stepanova D.V. Natural Language Processing and Fiction Text: Basis for Corpus Research // Вестник Российского университета дружбы народов. Серия: Теория языка. Семиотика. Семантика. 2024. Т. 15. № 1. С. 195–210. https://doi.org/10.22363/2313-2299-2024-15-1-195-210

Introduction

The context of this applied study is shaped by the fields of natural language processing (NLP), corpus linguistics and more widely — computational linguistics, which are often overlapping each other throughout their development or related to one other as part-to-whole [1]. In fact, today NLP and corpus linguistics are found in increasing proportion within the same research projects, as evidenced by numerous publications from high-rank scientific journals.

For example, NLP libraries can be used for *text mining* in the digital corpus of books [2]. Fonseca et al. [3] apply NLP tools along with the linguistic corpora based on the XML tagging, analyzing the genre compositional structure. The NLP

tools are also used for automated text analysis of a large corpus of journal articles. The authors come to the conclusion that '*the creation of such corpora … can lead to more valid and reliable analysis of historical texts and thus contribute to the deeper understanding of different historical eras*' [4. P. 13].

The increasing use of NLP tools by corpus researchers is understandable. On the one hand, a linguistic corpus is a very powerful tool for language research, especially when it comes to functional comparability and identifying differences in the use of individual linguistic phenomena [5]. On the other hand, building a corpus is an extremely time-consuming task that requires a large amount of human and material resources. In addition, the linguistic corpus is not always a flexible tool. In other words, its parameters must be determined once and for all, since it is not practical to change the rules of the game at the moment when several million tokens have been staked. In this regard, there arises a natural need to build a functional linguistic corpus in a fully automatic mode, at least in order to test the parameters one has chosen and change them quickly if necessary.

An example is our previous study, in the framework of which an original corpus editor, a special grammatical corpus was built for E.M. Remarque's novel "The Night in Lisbon". The XML tagging was applied. The basic token of the corpus was a sentence, in which the presence of certain grammatical features was marked: the mood, the tense and the voice for the verbs, degrees of comparison for the adjectives, the case for the prepositions, etc. The problem was that once the text of the novel had been completely marked up, it would have been impossible to introduce a new feature without having to do all the work again, sentence by sentence [6].

A linguistic corpus, as a collection of texts, reflects the phenomenon of a certain historical period of time. However, among such texts one can distinguish the so-called *strong* and *weak* cultural texts. Strong texts have philological vitality that let them exist for centuries in different cultures [7]. The strong texts rightly refer to fiction and poetry. Weak texts exist *hic et nunc*; they translate current knowledge data. These texts contribute to the development of terminological systems, as well as the promotion of innovations and modern goods and services. In our research, we focus on strong texts of the world culture that require analysis and interpretation.

The aim of this study is to develop a model of such a technical device for the analysis and interpretation of a fiction text, which would reveal the full potential of popular NLP tools within the corpus approach.

To achieve this, it is necessary to solve the following tasks:

1) to describe the potential of one of the most popular NLP libraries for the fiction text processing;
2) to determine the *corpus parameters* of the research, i.e. to answer the question to what extent the corpus approach will be applied;
3) to build a model of a technical device for creating and managing a structured fiction text bank (corpus);
4) to develop and test a beta version of a GUI application.

The scientific novelty of the study lies in the fact that we apply the crucial principles of the classical theories of text interpretation in conjunction with the latest methods and tools of the applied linguistics.

Material and Methods

The study material can be divided into two major groups. The first group represents the linguistic material and includes the German texts of the three novels written by Franz Kafka: "The Castle", "The Trial" and "Amerika", and the English texts of Jack London's series of stories "Smoke Bellew. Smoke and Shorty". So, it is what we actually study from the linguistic point of view.

Despite of the fact that F. Kafka's novels were written almost 100 years ago, they do not lose their relevance both in terms of the content and in terms of the linguistic material [8–10]. The work by J. London is also still of interest to many linguists and specialists in literature, who see great potential in the texts of this writer for analysis and practical use in their professional fields [11–13].

The works of F. Kafka and J. London can be rightfully ranked among the strong texts, which are able to integrate different cultural traditions of fiction. At the same time, they represent *complexity*, since the software tools for their interpretation must correspond to the uniqueness of strong texts, and *ease*, since these texts are relevant for different cultures and are aimed at integrating common knowledge among representatives of the different ethnic communities.

The second group describes several technical tools, such as Python programming language, SQLite database, PyQt GUI library and spaCy NLP library. It is through what we come to the study of the linguistic material. Today Python is a leading powerful digital tool that is used not only to solve scientific issues, but also in programming in general[1]. The SQLite database can be connected to stand-alone applications, as the PyQt GUI library is a tool for creating graphical interfaces with Python to use them beyond the world wide web. The latter is very popular and is applied in many research projects [14–16].

SpaCy NLP library was released about eight years ago, and now it is a widely used toolkit for processing texts in 23 languages[2]. The accuracy of the procedure is quite high, especially for the English language. However, there are groups of researchers who are trying to improve the accuracy of the library and compare its effectiveness with similar products: StanfordNLP, NLTK, OpenNLP, ect. [17].

The general methods used in the study are analysis and synthesis. Special methods are additionally used to solve specific individual issues: descriptive method for tasks 1 and 2, modelling for task 3 and qualitative and quantitative analysis [18] for task 4.

[1] see TIOBE Index for April 2023: https://www.tiobe.com/tiobe-index.
[2] spaCy Models & Languages: https://spacy.io/usage/models#languages.

Theoretical Background

The field of description and interpretation of a fiction text as a branch of the linguistics has its long tradition and is based on the works by a number of scientists. The theoretical basis of our study is formed by the works of the representatives of the Russian school of German studies: O.I. Moskalskaya, E.V. Gulyga, E.I. Shendels, L.A. Nozdrina, Y.M. Kazantseva, etc. [19. P. 90–92].

The above-mentioned researchers paid much attention to the issues related to theoretical grammar, interpretation of (fiction) texts, analysis of the elements of philosophical and grammatical concepts as text-forming categories. In the field of computational linguistics, the most important studies for us were conducted by R.K. Potapova [20] and A.V. Zubov [21]. Considering the fundamental *classics*, we simultaneously apply in our research the achievements of modern scientists from the field of corpus linguistics and NLP. Thus, we will be able to combine traditional interpretation methods with the modern software tools to achieve objective results.

In modern scientific publications dedicated to using spaCy NLP library in corpus research, as well as to processing unordered text arrays, we can highlight several areas. A large number of research projects are devoted to building ready-made software solutions using spaCy [22–25]. Usually this library is used as a software tool without making a special graphical interface, e.g., for automatic text summarization [26] or information extraction from different file formats [27]. A slightly different way is to develop the functionality of the library or to add new languages to it [28; 29]. Many researchers solve highly specialized tasks with the help of the library. So, Soni and Rambola [30] use spaCy for aspect-level sentiment analysis and opinion mining. Chantrapornchai and Tunsakul [31] present two machine learning-based methodologies used to extract particular information from full texts about restaurants, hotels, shopping, and tourism. Singh et al. [32] detect named entities from sentences written in the Punjabi language's Gurmukhi script. This list can be continued with other relevant items. Furthermore, we rely on the results of our own research in the field of corpus linguistics and text interpretation in which we aim to get a synergistic effect through a combination of precise tools and mental interpretation [33; 34].

Study and Results

To solve the first task of our study, it was necessary to understand what exactly spaCy library was able to offer for the fiction texts analysis. First of all, it should be mentioned that the library is not a ready-made software and can be used in its original form by Python programmers only. This condition makes its usage difficult for a wide range of the linguists who do not have programming skills.

The most important spaCy tools are the following: *sentencizer*, *tokenizer*, *lemmatizer* and *morphologizer*. This means that the library is able, firstly, to parse the text into sentences and tokens. Secondly, for each of the tokens the library determines a particular part of speech it refers to. And thirdly, for each token, spaCy determines its morphological attributes and their values and the initial form of the

word — a *lemma* (e.g. an infinitive for a verb or a singular nominative form for a noun, personal pronoun, ect.).

At the next stage of the study, it was necessary to understand to what extent we would use the corpus approach, and what exactly that could mean. As we have already singled out the difficulties that are associated with creating a corpus in the classical sense, that is, using manual mark-up. However, since we applied a corpus approach, we determined that the tool for studying linguistic phenomena in the fiction texts should be a linguistic corpus. As a result, we chose a linguistic corpus, created in a fully automatic way. Nevertheless, this alone did not give an understanding of what the corpus we needed would look like.

In fact, it was necessary to make a choice between two options: the XML mark-up or the SQL mark-up. The first option was convenient, because the XML database can be read by human. But in the course of our research, we came to a conclusion that the XML files were rather "bulky" and working with them was not as convenient as working with the SQL files, though the SQL data base for a novel was larger than the XML database of the same fiction work.

The structure of our SQL database as a relational database can be represented in the form of two tables: one for the sentences (Table 1).

Table 1

Database table for the sentences

	ID	Sentence number	Sentence text
Data type	integer	integer	text

And one for the tokens (Table 2).

Table 2

Database table for the tokens

	ID	Token number	Sentence number	Token text	Token part of speech	Token lemma	Token attributes and values
Data type	integer	integer	integer	text	text	text	text

The both tables are related through the sentence number, so for each token one can track its sentence. The cell TOKEN ATTRIBUTES AND VALUES deserves special attention, since the attributes and their value are listed there all at once, and not distributed between the cells. This somewhat simplified approach can be justified by the fact that different tokens can have a different number of attributes, so there cannot be the same number of cells in a row for all tokens. For example, for a verb it looks like:

Mood=Ind|Number=Sing|Person=3|Tense=Past|VerbForm=Fin

and for an adjective:

Case=Acc|Degree=Pos|Gender=Fem|Number=Sing

The pair *attribute & value* is separated by a vertical bar without spaces.

The experimental SQL databases (like all applications mentioned in this paper as our own) were generated automatically by a Python CLI application, which was developed in the Laboratory for Fundamental and Applied Issues of Virtual Education at Moscow State Linguistic University. As a basis, the TXT files of the novels were taken, which had been previously slightly formatted.

So, we opted for Python as a programming language, SQLite as a database and PyQt as a GUI library, not to mention spaCy as an NLP library. The main requirement that the corpus manager was supposed to suit was that it had to be flexible. The idea was that the developer can quickly alter it, and that related also to the graphical user interface. Here we relied on the concept of rapid application development (RAD) that *'has become a major trend in the field of software development'* [35. P. 1].

To meet these requirements the graphical shell of the corpus manager had to be developed in a separate file, as a Python class, and all programme logic was included into the main Python programme file, in which the graphical class was imported as a module. The graphical shell was created in Qt Designer, which generated Python class code automatically. From this file the developer got GUI objects names to refer to them in the main file. As a result, we were able to make changes to the graphical shell (location of objects, colours, etc.) without having to modify the main file.

Having determined the structure of the linguistic corpus and the methodology for its development, let us turn to the programming logic as a set of functions in the main file, which are expected to perform specific tasks, e.g., clear the output area or close the application.

These tasks are the following:

1) open a new linguistic corpus;
2) output a frequency list of all lemmas, from the most frequently used to the least frequently used;
3) output the given sentence in the context of several sentences;
4) output all sentences containing the given lemma;
5) output all sentences containing the given list of lemmas (e.g. 'Dorf', 'Schnee', 'kalt');
6) output all sentences containing the given token;
7) output all sentences containing the given list of tokens (e.g. 'Dorfes', 'Schnee', 'kalten');
8) output all sentences containing the given part of speech;
9) output all sentences containing the given parts of speech (up to three parts of speech);
10) output all sentences containing the given morphological attributes of the tokens (e.g. the tense or the person, up to three attributes);
11) output all sentences containing the given values of the morphological attributes of the tokens (e.g. the past tense or the 1st person, up to three values);

12) output all sentences after the manual request according to the SQL rules (e.g. SELECT * FROM tokens WHERE id > 1 AND id < 30);
13) complex request — output all sentences containing the given morphological attributes with the given values (up to three attribute & value pairs);
14) complex request — output all sentences containing the given part of speech with up to three given attribute & value pairs;
15) complex request — output all sentences containing the given lemma with up to three given attribute & value pairs;
16) complex request — output all sentences containing the given token with up to three given attribute & value pairs.

Moreover, a number of technical parameters had to be taken into account to make the interface more friendly. This mainly concerned the activation and deactivation of the GUI objects or widgets. For example, one may not press the request button until a new linguistic corpus is opened.

It was assumed that the basic language of the corpus manager would be Russian. The translation of the interface into other languages would be possible.

According to the parameters of the model we have built the following graphical shell (Figure 1):

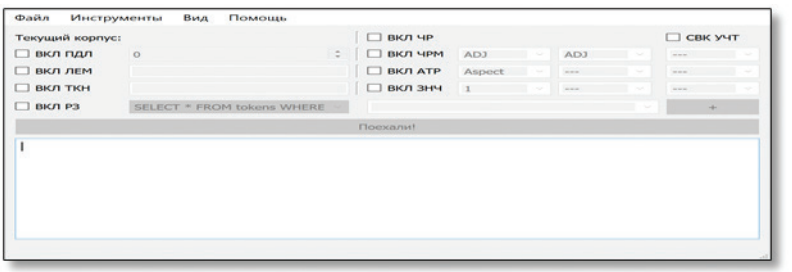

Fig. 1. Graphical shell for the corpus manager
Source: compiled by the authors.

The menu bar is located at the top of the main window followed by the settings block with the Engage button at the bottom. Then the output area is situated. The most complex part is the settings block, which contains multiple check boxes that activate or deactivate related widgets. Other objects are Qt spin boxes, Qt combo boxes, Qt line edits, Qt push buttons and a Qt label.

To start working with the corpus manager the user must open a new linguistic corpus in the *File* > *Open Corpus* menu (Файл > Открыть корпус). The number of sentences and tokens in the current data base appears in the top left corner (e.g. Текущий корпус: amerikasql.db 4025 / 100230). Then the Engage button (Поехали!) becomes active, and a request can be launched. To output the given sentence in the context of several sentences one can switch on the sentences check box (ВКЛ ПДЛ) and input a number into the Qt spin box. The result appears in the output area. The programme is configured in such a way that the context includes five sentences before and five sentences after the given sentence.

To get the frequency list the user can click the *Tools* > *Frequency list* menu (Инструменты > Частотный список). For example, from "Amerika" we get the following (ten first lines are listed):

, : *8844*
der : *6819*
ich : *5593.* : *3450*
doublequote : *2997*
und : *2476*
sein : *1933*
sich : *1855*
haben : *1260*
Karl : *1211*

Here we see the so-called *raw sorting* according to spaCy rules, since there are also punctuation marks among the tokens. However, at position ten we notice the name of the main character Karl. All simple requests function in a similar way. The user switches on the checkbox, enters the data and gets the result. For the manual request two options for the beginning of the request text are foreseen: *SELECT * FROM sents WHERE* and *SELECT * FROM tokens WHERE*. The user needs to add the necessary parameters only. To clear the output area one should click the *View* > *Clear the area* menu (Вид > Очистить поле). The output area is an editable object, so the text can be easily copied from it with the hotkeys.

The complex requests are a little trickier. If one switches on a few checkboxes, enters the data into the appropriate input widgets and presses the Engage button, the programme performs all requests one by one as a simple sequence. To activate the complex request as a special procedure the corresponding checkbox in the top right corner (СВК УЧТ) must be activated. After that four options become available.

If *СВК УЧТ & ВКЛ АТР & ВКЛ ЗНЧ* are checked and other checkboxes are unchecked, the programme selects the sentences that contain tokens with the given morphological attributes and values. For example, for the request *Degree=Cmp & Case Dat* in the demo corpus, which consists of 29 sentences, one gets the following result:

[‚Degree‘, ‚Case‘] : *[‚Cmp‘, ‚Dat‘]* : *zum (Case=Dat); Übernachten (Case=Dat); früheren (Degree=Cmp)* : *19* : *„Und man muß die Erlaubnis zum Übernachten haben?“ fragte K., als wolle er sich davon überzeugen, ob er die früheren Mitteilungen nicht vielleicht geträumt hätte.*
Токенов: *3/0 (Tokens*: *3/0)*
Предложений: *1 (Sentences*: *1)*
Предложений всего: *29 (From Sentences*: *29)*

Here we receive sentence number 19 only, which includes tokens that match the request: the comparative degree (of adjective) and the dative case (of the preposition and the noun). Statistics are displayed at the end of each output. In this case it is the number of the tokens and sentences found and the total number of sentences in the corpus. 3/0 means that three tokens are found totally, but no token meets all the given parameters. Obviously, an adjective *früheren* can stay in the dative case, but in this

sentence it is used in the accusative case. Therefore, if one changes the request and inputs *Degree=Cmp & Case Acc*, one gets the following result:

[‚Degree‘, ‚Case‘] : [‚Cmp‘, ‚Acc‘] : ihre (Case=Acc); Sessel (Case=Acc); besser (Degree=Cmp) : 12 : Die Bauern waren auch noch da, einige hatten ihre Sessel herumgedreht, um besser zu sehen und zu hören.
[‚Degree‘, ‚Case‘] : [‚Cmp‘, ‚Acc‘] : die (Case=Acc); Erlaubnis (Case=Acc); sich (Case=Acc); die (Case=Acc); früheren (Degree=Cmp); früheren (Case=Acc); Mitteilungen (Case=Acc) : 19 : „Und man muß die Erlaubnis zum Übernachten haben?“ fragte K., als wolle er sich davon überzeugen, ob er die früheren Mitteilungen nicht vielleicht geträumt hätte.
Токенов: 10/1 (Tokens: 10/1)
Предложений: 2 (Sentences: 2)
Предложений всего: 29 (From Sentences: 29)

This time two sentences match the parameters, and one token has *Degree=Cmp & Case=Acc*. The second complex request is *СВК УЧТ & ВКЛ ЧР & ВКЛ АТР & ВКЛ ЗНЧ*. The checkboxes must be set as shown below (Figure 2):

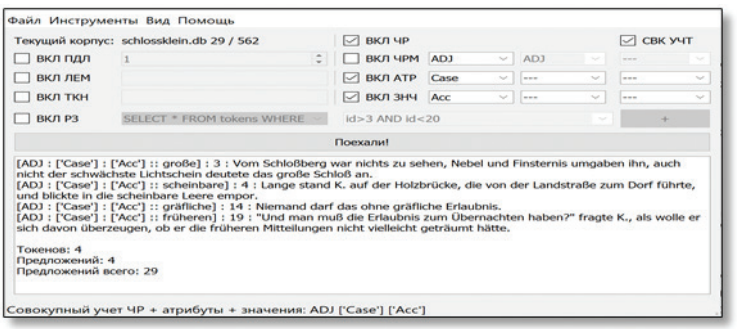

Fig. 2. Output for the request СВК УЧТ & ВКЛ ЧР & ВКЛ АТР & ВКЛ ЗНЧ
Source: compiled by the authors.

In this case only one part of speech can be selected, but all three attribute & value pairs are available. As it can be seen from our example, the programme has found sentences where the adjectives stay in the accusative.

The third and the fourth complex requests are *СВК УЧТ & ВКЛ ЛЕМ & ВКЛ АТР & ВКЛ ЗНЧ* and *СВК УЧТ & ВКЛ ТКН & ВКЛ АТР & ВКЛ ЗНЧ*. The difference between them and the second complex request is that instead of a part of speech, a lemma and a token are selected, and all three attribute & value pairs are also available. In the case of lemma one can find all occurrences of a word in particular forms. For example for lemma=stehen & attribute=Number & value=Sing the output can be like this:

[stehen : [‚Number‘] : [‚Sing‘] :: stand] : 4 : Lange stand K. auf der Holzbrücke, die von der Landstraße zum Dorf führte, und blickte in die scheinbare Leere empor.
[stehen : [‚Number‘] : [‚Sing‘] :: stand] : 11 : Ein junger Mann, städtisch angezogen, mit schauspielerhaftem Gesicht, die Augen schmal, die Augenbrauen stark, stand mit dem Wirt neben ihm.

In such a way one can separate homonyms from each other. For example, for the English *square* one can get the following output by the simple lemma request (J. London's corpus):

[square] : square : 2848 : "I'm going to have a square meal before I start," Smoke said.
[square] : squared : 3100 : Smoke squared his shoulders and laughed non-committally.
[square] : square : 6758 : "A square deal!"
[square] : square : 6990 : He's got about twenty thousand square miles of huntin' country here all his own.
[square] : square : 7288 : An' he croaked one square through the chest."

But for the request lemma=square & attribute=Tense & value=Past one gets the verb *squared* only, and for lemma=square & attribute=Degree & value=Pos the result will include three sentences with the adjectives. To get the noun the request combination should be lemma=square & attribute=Number & value=Sing, because spaCy does not assign *number* to the English adjectives.

Discussion

Thus, we have clearly stated that the GUI application operates within the given parameters. Now it is necessary to evaluate critically the benefits of the work that has been done and identify the fields, in which the created tool could be effectively applied. First of all, it may be used for the fiction texts interpretation. Let us give some examples.

Using the simple lemma request one can easily calculate the concentration of the modal verbs (*müssen, sollen, können, dürfen, wollen, mögen*) among the sentences in the given fiction work. We identify this parameter as the author's modal *fingerprint*, because it tends to be unique for every writer. Therefore, for Kafka's novels the concentration factor is equal to 30.7 % ("The Castle"), 31.8 % ("The Trial"), 31.7 % ("Amerika"), that is, they are almost identical.

We can also consider the grammatical category of the person in the novel "The Castle". The distribution between the 1st, 2nd and the 3rd person in the novel is the following (Figure 3):

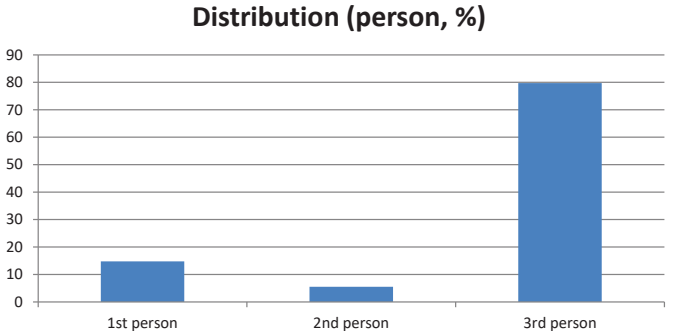

Fig. 3. Distribution between the 1st, 2nd and the 3rd person in "The Castle"
Source: compiled by the authors.

If we hypothesize that such a distribution may influence subconsciously the reader's perception of the novel, then these data contribute to the explanation, how the idea of alienation is expressed at the linguistic level.

The distribution between the tokens in the present and in the past allows one analyze the temporary structure of a fiction text. Not to mention the fact that one can get the contexts of the author's use of the certain language units or phenomena.

To be just, let us note that the critical point of using *raw* NLP libraries is that one always has to take into account the possibility of an error, which can lay, in our opinion, up to 20 %.

An automatically generated tool can serve as a basis for more complex and advanced corpus research. If one alters slightly the proposed database structure by putting extra cells into the rows of the sentence and token tables, one could additionally mark up the corpus units, e.g., semantic or stylistic features. This would greatly enrich the analysis and open new prospects for the research.

From the point of view of linguodidactics, the proposed software may also be applied as a tool for compiling textbooks in such a complex discipline as *Home Reading*. Learning materials for it should contain various exercises, including grammatical training of certain phenomena, but based on the given fiction work text. For example, if one needs to study the construction *would + infinitive*, one can easily find sentences that contain it, applying a simple token request, and gets the following output (J. London's corpus):

[would] : *would* : *219* : *He had not dreamed the invitation would be accepted.*
[would] : *would* : *253* : *Despite the fact that the Indian packers had jumped the freight from eight cents a pound to forty, they were swamped with the work, and it was plain that winter would catch the major portion of the outfits on the wrong side of the divide.*
[would] : *would* : *277* : *She was dressed as any woman travelling anywhere would be dressed.*
ect.

The most interesting is the combination of several grammatical features in one sentence. For the German grammar it may be, e.g., the reflexive verbs and indefinite pronouns ("Amerika"):

[‚Reflex', ‚PronType'] : [‚Yes', ‚Ind'] : 1575 : Wir haben ihm unser Vertrauen geschenkt, haben ihn einen ganzen Tag mit uns geschleppt, haben dadurch zumindest einen halben Tag verloren und jetzt — weil ihn dort im Hotel irgend jemand gelockt hat — verabschiedet er sich, verabschiedet sich einfach.
ect.

Using various request options, the author of the learning materials can achieve the results he or she needs.

Conclusions

Let us conclude that an attempt to systematize and automate the work with linguistic corpora makes it possible to solve not only clearly defined and specific research tasks, but also opens the prospects of partial automation of the fiction texts

interpretation. The partial automation of this procedure makes it possible to solve some particular research problems of the corpus linguistics. For example, the analysis of the gender factor, realized through the systematic use of the lexical and grammatical means.

Our research has yielded positive results. We have described the potential of spaCy NLP library for the fiction texts processing. The strengths of the library are its sentencizer, tokenizer, lemmatizer and morphologizer, though an error can reach up to 20 %. The corpus parameters of the research have been determined. The language material has been organized as an SQL based linguistic corpus, what contributes to building a model of a technical device for creating and managing a structured fiction text array. Finally, we have developed and tested a beta version of a GUI application that is able to process several types of requests.

The development of special software for the fiction texts processing stimulates the search for scientific solutions and at the same time allows one to understand the essential differences that exist between natural and artificial intelligence.

References / Библиографический список

1. Tsujii, J. (2021). Natural language processing and computational linguistics. *Computational Linguistics*, 47(4), 707–727. https://doi.org/10.1162/COLI_a_00420
2. O'Neill, H., Welsh, A., Smith, D.A., Roe, G. & Terras, M. (2021). Text mining mill: Computationally detecting influence in the writings of John Stuart Mill from library records. *Digital Scholarship in the Humanities*, 36(4), 1013–1029. https://doi.org/10.1093/llc/fqab010
3. Fonseca, C.A., Guelpeli, M.V.C. & De Souza Netto, R.S. (2021). Representation of structured data of the text genre as a technique for automatic text processing. *Texto Livre*, 15. https://doi.org/10.35699/1983-3652.2022.35445
4. Szabó, M.K., Ring, O., Nagy, B., Kiss, L., Koltai, J., Berend, G. & Kmetty, Z. (2020). Exploring the dynamic changes of key concepts of the Hungarian socialist era with natural language processing methods. *Historical Methods*, 54(1), 1–13. https://doi.org/10.1080/01615440.2020.1823289
5. Malyuga, E.N. & McCarthy, M. (2021). "No" and "net" as response tokens in English and Russian business discourse: In search of a functional equivalence. *Russian Journal of Linguistics*, 25(2), 391–416. https://doi.org/10.22363/2687-0088-2021-25-2-391-416
6. Gorozhanov, A.I. & Guseynova, I.A. (2020). Corpus analysis of the grammatical categories' constituents in fiction texts considering the linguo-regional component. *Journal of Siberian Federal University. Humanities & Social Sciences*, 13(12), 2035–2048. https://doi.org/10.17516/1997-1370-0702. (In German).
7. Denisova, G.V. (2020). *Intertekst v sovremennoj sociokul'turnoj real'nosti Rossii i Italii*. Moscow: Kanon+. P. 272. (In Russ.).
   Денисова Г.В. Интертекст в современной социокультурной реальности России и Италии. М.: Kanon+, 2020. С. 272.
8. Milne, P.W. (2022). Praescriptum: Kafka's two bodies. *Philosophy Today*, 66(3), 587–603. https://doi.org/10.5840/philtoday2022324451
9. Itkin, A. (2021). Kafka's worlds. *German Quarterly*, 94(4), 493–508. https://doi.org/10.1111/gequ.12241

10. Roca, J.B. & Rius, N.I. (2020). Kafka and disease. between reality and writing [Kafka y la enfermedad. Entre la realidad y la escritura] *Revista Chilena De Literatura,* 102, 233–247. https://doi.org/10.4067/S0718-22952020000200223
11. Logue, M. (2022). Patrick MacGill: A path to socialism shared with Jack London. [Patrick MacGill: el Camino hacia el Socialismo junto a Jack London]. *Estudios Irlandeses*, 17, 54–64. https://doi.org/10.24162/EI2022-10645
12. Hernandez, A. (2021). Jack London's poetic animality and the problem of domestication. *Journal of Modern Literature*, 45(1), 40–55. https://doi.org/10.2979/jmodelite.45.1.03
13. López, J.I.G. (2020). Jack London, the socialist dream of a young poet. *Revista De Estudios Norteamericanos*, 24, 9–112. https://doi.org/10.12795/REN.2020.I24.05
14. Li, J., Lian, Z., Wu, Z., Zeng, L., Mu, L., Yuan, Y. & Ye, J. (2023). Artificial intelligence–based method for the rapid detection of fish parasites (ichthyophthirius multifiliis, gyrodactylus kobayashii, and argulus japonicus). *Aquaculture*, 563. https://doi.org/10.1016/j.aquaculture.2022.738790
15. Hachemi, A. & Zeroual, A. (2022). Computer-assisted program for water calco-carbonic equilibrium computation. *Earth Science Informatics*, 15(1), 68–704. https://doi.org/10.1007/s12145-021-00703-5
16. Li, W., Pu, H., & Wang, R. (2021). Sign language recognition based on computer vision. In: *Priceeding of 2021 IEEE International Conference on Artificial Intelligence and Computer Applications, ICAICA 2021*. pp. 919–922. https://doi.org/10.1109/ICAICA52286.2021.9498024
17. Schmitt, X., Kubler, S., Robert, J., Papadakis, M. & Letraon, Y. (2019). A replicable comparison study of NER software: StanfordNLP, NLTK, OpenNLP, SpaCy, gate. In: *Priceeding of 2019 6th International Conference on Social Networks Analysis, Management and Security, SNAMS 2019*. pp. 338–343. https://doi.org/10.1109/SNAMS.2019.8931850
18. Ajani, D.T. (2019). Grammatico-Semantic Content of Primitives in the Major Themes of News Watch's Reports on Nigerian Politics. *The international journal of humanities & social studies,* 7(12), 327–337. https://doi.org/10.24940/theijhss/2019/v7/i12/HS1912-066
19. Kraeva, I.A. et al. (2022). *Germanistika i lingvodidaktika v Moskovskom i Minskom gosudarstvennykh lingvisticheskikh universitetakh*: *Istoki, razvitie, perspektivy*. (In Russ.).
Краева И.А. Германистика и лингводидактика в Московском и Минском государственных лингвистических университетах: истоки, развитие, перспективы. Казань: Бук, 2022.
20. Potapova, R.K. (2012). Diskursivnaya sostavlyayushchaya sovremennoi korpusnoi lingvistiki (primenitel'no k ustno-rechevym bazam dannykh). *Bulletin of Moscow State Linguistic University*, 639, 157–167. (In Russ.).
Потапова Р.К. Дискурсивная составляющая современной корпусной лингвистики (применительно к устно-речевым базам данных) // Вестник Московского государственного лингвистического университета. 2012. № 639. С. 157–167.
21. Zubov, A.V. (2006). Korpusnaya lingvistika: vozmozhnosti i perspektivy. In: *Proceedings of Conference* "*Russkii yazyk*: *Sistema i funktsionirovanie"*, Minsk. pp. 22–27. (In Russ.).
Зубов А.В. Корпусная лингвистика: возможности и перспективы // Русский язык: система и функционирование. Минск: РИВШ, 2006. С. 22–27.
22. Kim, C., Choi, S., Jeong, J. & Lee, E. (2022). Automatic risks detection and comparison techniques for general conditions of technical documents in purchasing order. In: *Proceedings of ACM International Conference Proceeding Series*. pp. 236–241. https://doi.org/10.1145/3543712.3543721
23. Fantechi, A., Gnesi, S., Livi, S. & Semini, L. (2021). A spaCy-based tool for extracting variability from NL requirements. In: *Priceeding of ACM International Conference Proceeding Series, Part F171625-B*. pp. 32–35. https://doi.org/10.1145/3461002.3473074

24. Eyre, H., Chapman, A.B., Peterson, K.S., Shi, J., Alba, P.R., Jones, M.M. & Patterson, O.V. (2021). Launching into clinical space with medspaCy: A new clinical text processing toolkit in Python. In: *Proceedings AMIA ... Annual Symposium Proceedings*. AMIA Symposium, 2021. pp. 438–447.
25. Partalidou, E., Spyromitros-Xioufis, E., Doropoulos, S., Vologiannidis, S. & Diamantaras, K.I. (2019). Design and implementation of an open source Greek POS tagger and entity recognizer using spaCy. In: *Proceedings 2019 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2019*. pp. 337–341. https://doi.org/10.1145/3350546.3352543
26. Jugran, S., Kumar, A., Tyagi, B.S. & Anand, V. (2021). Extractive automatic text summarization using SpaCy in Python NLP. In: *Proceedings of 2021 International Conference on Advance Computing and Innovative Technologies in Engineering*, ICACITE 2021. pp. 582–585. https://doi.org/10.1109/ICACITE51222.2021.9404712
27. Channabasamma, Suresh, Y. & Manusha Reddy, A. (2021). A contextual model for information extraction in resume analytics using NLP's spaCy. *Inventive computation and information technologies*. Springer. pp. 395–404. https://doi.org/10.1007/978-981-33-4305-4_30
28. Harahus, M., Juhar, J. & Hladek, D. (2022). Morphological annotation of the Slovak language in the spaCy library with the pretraining. In: *Proceedings of 32nd International Conference Radioelektronika, Radioelektronika 2022*. https://doi.org/10.1109/RADIOELEKTRONIKA54537.2022.9764935
29. Kumar, D., Choudhari, K., Patel, P., Pandey, S., Hajare, A. & Jante, S. (2022). STAT simple text annotation tool (STAT): Web-based tool for creating training data for spaCy models. In: *ICT Analysis and Applications*. Singapore: Springer Nature. https://doi.org/10.1007/978-981-16-5655-2_29
30. Soni, P.K. & Rambola, R. (2021). Deep learning, WordNet, and spaCy based hybrid method for detection of implicit aspects for sentiment analysis. In: *Proceedings of 2021 International Conference on Intelligent Technologies, CONIT 2021*. https://doi.org/10.1109/CONIT51480.2021.9498372
31. Chantrapornchai, C. & Tunsakul, A. (2021). Information extraction on tourism domain using spaCy and BERT. *ECTI Transactions on Computer and Information Technology*, 15(1), 108–122. https://doi.org/10.37936/ecti-cit.2021151.228621
32. Singh, N. & Hussain, A. (2022). Rapid application development in cloud computing with IoT. In: *IoT and AI technologies for sustainable living*: *A practical handbook*. pp. 1–28. https://doi.org/10.1201/9781003051022-1
33. Gorozhanov, A.I., Guseynova, I.A. & Stepanova, D.V. (2022). Instrumentarii avtomatizirovannogo analiza perevoda khudozhestvennogo proizvedeniya. *Issues of Applied Linguistics*, 45, 62–89. https://doi.org/10.25076/vpl.45.03 (In Russ.).
Горожанов А.И., Гусейнова И.А., Степанова Д.В. Инструментарий автоматизированного анализа перевода художественного произведения // Вопросы прикладной лингвистики. М.: Национальное объединение преподавателей иностранных языков делового и профессионального общения в сфере бизнеса, 2022. № 45. С. 62–89. https://doi.org/10.25076/vpl.45.03
34. Gorozhanov, A.I. (2021). Metod komparativnogo analiza gruppy tekstov (na materiale nemetskoyazychnykh nauchnykh statei. *Bulletin of Moscow State Linguistic University*, 5(847), 48–59. https://doi.org/10.52070/2542-2197_2021_5_847_48 (In Russ.).
Горожанов А.И. Метод компаративного анализа группы текстов (на материале немецкоязычных научных статей) // Вестник Московского государственного лингвистического университета. Гуманитарные науки. 2021. № 5(847). С. 48–59. https://doi.org/10.52070/2542-2197_2021_5_847_48
35. Singh, N., Kumar, M., Singh, B. & Singh, J. (2022). DeepSpacy-NER: An efficient deep learning model for named entity recognition for Punjabi language. *Evolving Systems*, 14, 673–683. https://doi.org/10.1007/s12530-022-09453-1

Information about the authors:

Alexey I. Gorozhanov, Dr.Sc. in Philology, Associate Professor, Professor of the Department of Grammar and History of the German Language, Faculty of German Language, Moscow State Lingustic University (38, b. 1, Ostozhenka str., Moscow, Russian Federation, 119034); *Research interests*: German studies, applied linguistics, foreign language teaching; *e-mail*: a_gorozhanov@mail.ru
ORCID: 0000-0003-2280-1282; SPIN-code: 1753-4920, AuthorID: 592980.

Innara A. Guseynova, Dr.Sc. in Philology, Associate Professor, Vice-Rector, Moscow State Lingustic University (38, b. 1, Ostozhenka str., Moscow, Russian Federation, 119034); *Research interests*: applied linguistics, media, lexicology; *e-mail*: guseynova@linguanet.ru
ORCID: 0000-0002-6544-699X; SPIN-code: 1635-5260, AuthorID: 662483.

Darya V. Stepanova, PhD in Philology, Associate Professor, Minsk State Lingustic University (21, Zakharova, Minsk, Belarus, 220034); *Research interests*: applied linguistics, Media Discourse, lexicography; *e-mail*: daryastepanova79@gmail.com
ORCID: 0000-0002-2857-4386; SPIN-code: 5291-8660, AuthorID: 1131273.

Сведения об авторах:

Горожанов Алексей Иванович, доктор филологических наук, доцент, профессор кафедры грамматики и истории немецкого языка, факультет немецкого языка, Московский государственный лингвистический университет (119034, Российская Федерация, г. Москва, ул. Остоженка, 38, стр. 1); *научные интересы*: германистика, прикладная лингвистика, методика обучения иностранным языкам; *e-mail*: a_gorozhanov@mail.ru
ORCID: 0000-0003-2280-1282; SPIN-код: 1753-4920, AuthorID: 592980.

Гусейнова Иннара Алиевна, доктор филологических наук, доцент, проректор, Московский государственный лингвистический университет (119034, Российская Федерация, г. Москва, ул. Остоженка, 38, стр. 1); *научные интересы*: прикладная лингвистика, медиадискурс, лексикология; *e-mail*: guseynova@linguanet.ru
ORCID: 0000-0002-6544-699X; SPIN-код: 1635-5260, AuthorID: 662483.

Степанова Дарья Валерьевна, кандидат филологических наук, доцент, Минский государственный лингвистический университет (220034, Республика Белорусь, г. Минск, ул. Захарова, 21); прикладная лингвистика, медиадискурс, лексикография; *e-mail*: daryastepanova79@gmail.com
ORCID: 0000-0002-2857-4386; SPIN-код: 5291-8660, AuthorID: 1131273.