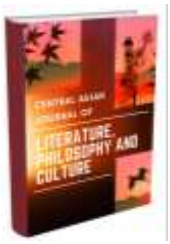

CENTRAL ASIAN JOURNAL OF LITERATURE, PHILOSOPHY, AND CULTURE

https://cajlpc.casjournal.org/index.php/CAJLPC

Volume: 07 Issue: 01 | January 2026 ISSN: 2660-6828

Article

Sorting and Storing Search Results in The Uzbek Language Corpus. Lexical Search Parameters

Yuldashev Aziz Uyg'un O'g'li[*1]

1. Teacher at TashDO'TAU

* Correspondence: yuldashevaziz@navoiy-uni.uz

Abstract: The advancement of corpus linguistics in Uzbekistan requires the establishment of efficient digital tools for linguistic data retrieval and analysis. Despite the creation of several electronic corpora, comprehensive studies on search mechanisms—specifically sorting, storing, and lexical filtering in the Uzbek language corpus—remain limited. Current corpus systems lack a detailed methodological framework for organizing search results, handling lexical parameters such as lemmas and morphemes, and ensuring user-friendly data export and management. This study aims to analyze and systematize search mechanisms for the Uzbek language corpus by focusing on sorting, storing, and lexical search parameters, and adapting international corpus practices to the morphological complexity of Uzbek. The insights gained through the findings also unveil how even the very data that is presented within some of these resources can be made even more meaningful and discoverable through combining alphabetical, frequency and metadata-based sorting with lemma- and morpheme-based search capabilities to improve search functionality as a whole. Moreover, exporting (CSV, XML, JSON) and history-saving functions make sure that the software will be usable in the long-term for research. The research presents a general model for combining computational and linguistic principles to increase the efficiency of corpus search and an adaptive model of dealing with agglutinative structures. The proposed system strengthens the methodological foundation of Uzbek corpus linguistics, facilitates corpus-based research and teaching, and supports the development of computational linguistics in Uzbekistan by transforming the Uzbek corpus into an interactive, analytical, and educational digital resource.

Keywords: corpus of the Uzbek language, search engine, lemma, morpheme, word group, affix, synonym, collocation, semantic field.

Citation: O'g'li, Y.A. Sorting and storing search results in the Uzbek language corpus. Lexical search parameters. Central Asian Journal of Literature, Philosophy, and Culture 2026, 7(1), 32-38.

Received: 10th Aug 2025
Revised: 16th Sep 2025
Accepted: 24th Oct 2025
Published: 06th Nov 2025

1. Introduction

Sorting search results within a linguistic corpus is a crucial process that ensures the organization, accessibility, and analytical value of retrieved data. In corpus linguistics, sorting refers to arranging the results of user queries according to specific criteria and purposes, which allows researchers to analyze patterns of word usage and contextual relationships more efficiently. Typically, search results are presented in KWIC (Keyword in Context) format, where each keyword appears alongside its surrounding context, generating a large number of concordance lines. So to convert this data to something workable, it has to be tabulated. It sorts to show the eligible topmost cases in a clearer way with organised content overview for the corpus. Good sorting allows for more effective viewing of linguistic patterns, collocational tendencies, and semantic connections between words, i.e. lexical items.

In practice, different sorting methods are performed, such as alphabetical, by frequency, and based on metadata with different analysis advantages according to the study objective. For instance, sorting it alphabetically helps in recognizing collocational

tendency whereas frequency order shows the most frequently used lexical items in a corpus. Furthermore, metadata-based sorting by easier keywords, (e.g., genre, author, or year of publication) allows for both diachronic and stylistic analyses of linguistic phenomena. These advanced sorting mechanisms provide essential support for corpus-based linguistic research, language teaching, and computational language processing, and are thus especially valuable within the Uzbek language corpus.A well-structured corpus search and sorting system not only facilitates efficient data retrieval but also strengthens the methodological foundation of modern Uzbek corpus linguistics [1].

2. Materials and Methods

The methodological framework of this study is based on a combination of analytical, descriptive, and computational approaches designed to examine the processes of sorting, storing, and searching data within the Uzbek language corpus. The research integrates both theoretical and practical methods from corpus linguistics and computer linguistics to develop and evaluate the efficiency of search mechanisms. This paper is based on an experimental investigation of lexical search parameters (lemma, morpheme, and affix search), as well as sorting and storage procedures (alphabetical, frequency, and metadata-based sorting). These tools and software environments (e.g., AntConc; NVivo) are heavily used to test search algorithms and to verify the organization of data in the main corpus. The data for lemmatization and morphological analysis testing was collected by building a text corpus with lexically and morphologically tagged texts. The methodology process adopted the general corpus search workflow: feed in user queries, fetch results based on matching criteria, Sort algorithms, and save outputs in different formats (CSV, JSON, and XML). Comparative search system (e.g., Russian National Corpus & English-Corpora) Out of this process, a necessary survey for methods that are relevant to adapt to the structure of the Uzbek language was made using the Kaggle. org site. This study utilized qualitative interpretation and quantitative measurements between search accuracy, contextual relevance, and data export efficiency. This integrative methodological approach ensures a comprehensive evaluation of corpus search functions and their applicability in linguistic research, education, and computational language processing [2].

3. Results and Discussion

Sorting criteria can be based on several different parameters. The following sorting methods are widely used in the practice of corpus linguistics:

1. **Alphabetical sorting.** The method of arranging the result lines in alphabetical order according to the context to the left or right of the keyword is widespread in KWIC concordances. In this case, sorting is carried out according to the first word after the keyword or sorting is carried out according to the word before the keyword. As a result, the user can quickly see which words the keyword is most often associated with, typical combinations. For example, in programs such as AntConc, it is possible to sort concordance lines using a three-stage sorting (word 1 to the left of the keyword, word 2, etc.). Contexts located in this order alphabetically are useful in analyzing the combinatorics of language units. Alphabetical sorting can also be understood as sorting the results by context. Alphabetical sorting arranges lines according to the spelling order of words in context. This method is very convenient for viewing linguistic regularities and collocational connections [3].
2. **In the frequency sorting method,** the results are sorted according to the frequency of occurrence of the unit or its contexts. If the search results are a list of different words (if the user has found several different lexemes according to a template), then they can be arranged in descending order of frequency of occurrence in the corpus. For example, the Hermetic Concordance software claims that the list of all the different words extracted from the text can be sorted either alphabetically or by frequency. As a result of frequency sorting, the most frequent examples and general cases appear first, and less frequent ones are given later. This approach allows the researcher to immediately see and analyze the relative distribution of the search unit in the corpus [4].

3. **Sorting by text source or metadata.** If the texts in the corpus have extralinguistic parameters (genre, author, year, style, etc.), the search results can also be sorted by this metadata. For example, presenting the results in chronological order by the year the text was created helps the user see changes in word usage over time. In addition, the method of grouping the results by text genres can also be used. In this case, examples from fiction are given first, then examples from the scientific style. In practice, sorting the results by text identifier and word position is often used as a standard sorting method. This preserves the order of occurrence in the texts. For example, in the concordance system proposed by S. Bahodirov, N. Murodova, the results are sorted by the title of the work (text title) and the ordinal number of the word in the text. As a result, within each text, words are displayed in the order in which they appear, making it easier to study the context within the same text [5].
4. **Sort by size of context.** Sometimes results can be sorted based on how much context (words) is retrieved around a keyword. For example, some concordance systems may differentiate between whether the keyword occurs at the beginning or end of a sentence, or sort based on whether punctuation marks are present around the keyword. Context size can be understood as the length of the sentence in which the keyword occurs, or the number of words retrieved before and after the keyword. If the user chooses to display context more broadly or narrowly, the results are sorted accordingly, giving context from shorter sentences first or context from longer sentences first. Such settings are usually included in the view settings and are presented to the user through the interface. For example, in software such as NVivo, the context for KWIC results is a narrow view of 5 words by default, which the user can expand [6].
5. **Sorting by relevance.** If the search engine is more sophisticated and can calculate the level of relevance to the user's query through a scoring function (using vector models or ML models), the results will also be sorted by relevance scores. This is a typical method for Internet search engines. However, such a scoring ranking is usually not used in linguistic corpus searches, since the user usually wants to see all examples and provides a more precise search through filters. However, if the user has made a very general query, such as "description of pictures in Uzbek language text", the system may offer a sort by approximate relevance [7].

The above sorting types should be controlled by the user through the viewing settings. For example, the new interface of the Russian National Corpus (RNC) has special buttons for selecting the order in which results are displayed and configuring viewing parameters, with which the user can determine how many examples to display in one window and in what order to sort. The sorting parameters are saved in the user's browser and are used for subsequent searches.

Sorting the results is also a tool to provide analytical convenience to the user. For example, by sorting alphabetically by the left side of the keyword, you can determine in which collocational rows the word is most often used (for example, by sorting the results of the search for *"ko'ngil*"* and see the frequency of combinations such as *"ko'ngil uchun"*, *"ko'ngil bilan"*, *"ko'ngil bo'lib"*). Or, conversely, by sorting by the right side of the word, you can analyze what adjectives or determiners it comes with. **Figure 1** below provides a general diagram of the corpus search process, which shows the sequence of steps from receiving a user query to sorting the results. As shown in this diagram, sorting plays an important role in the formatting process of search results, and it can also influence the process of refining a user query or submitting a new query [8].

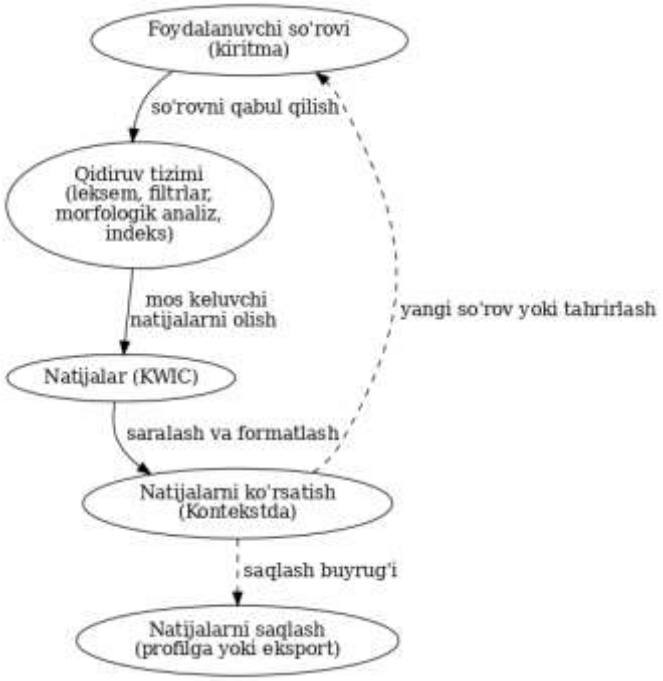

Figure 1. General flow chart of the corpus search process

Saving (exporting) search results. The function of saving the results found in corpus systems for the user is also very important. Saving search results is the ability of the user to export the list of search results in the desired format or save them in their profile for later review. Such capabilities are very useful in scientific research, analyzing data obtained from the corpus, and downloading and analyzing the results into other programs. Saving search results can be implemented in several ways:

1. **Save to the user profile.** If a user is registered with a corpus system, they can save specific search results or query formulas to their profile. This allows the user to later access their profile and view their previous results or rerun the same query with a single click. For example, on platforms such as the British National Corpus (BNC) or Sketch Engine, users can save concordance results or graphical analyses using the “Favorites” or “Saved searches” feature. The save to profile option saves results online, so even if the user changes devices, they can still access their profile and view the results online [9].
2. **Exporting the results to a file.** This is one of the most common and convenient methods. The user can download search results to his computer in CSV, Excel (XLS/XLSX), JSON, XML or TXT formats. CSV/Excel formats allow you to save the results in a table format (for example, in columns: word itself, left context, right context, source text identifier, etc.). This allows you to open them in programs such as MS Excel or Google Sheets and sort, filter or perform statistical analysis. JSON or XML formats store the results in a structured data format (in the form of a tree), which is convenient for processing through a programming environment. For example, in the Hermetic Search KWIC program, after creating a concordance, the user has the option to perform any desired repeated searches and write the results to a file. If the user specifies a specific file name, the search results are written to this file, otherwise they are only displayed on the screen. In corpus systems, the "Export results" button usually performs this function [10].
3. **Save search history.** All searches performed by the user through the system can be saved as a history. The purpose of this function is to allow the user to review his previous searches and return to them if necessary. For example, in the concordance program presented at the TashSULLU scientific conference, each query text entered by the user, the selected filters and the search type are recorded in the database as history in the SearchHistory object. This history can later be displayed within the

user's session or profile (for example, in the form of a "Recent searches" list). Saving history makes it easier for the user to return to the previous search only if he performs several stages of searches (for example, first a broader query, then an internal search from its results). In addition, this information is also important for search statistics, as the system administrator knows which words or queries are being searched most often [11].

4. **Temporary storage of results in the system.** If the user wants to compare several sets of results during the same session, the system allows him to *"keep"* the current results in memory. For example, the user temporarily saves the results of the first search in the buffer, and then performs a second search. Then the system has the opportunity to compare these two sets of results using the *"Compare"* function. This is, of course, a relatively complex task and is not found in all corpora, but it is important for scientific research.
5. **Print the results.** There is also the possibility of printing (printing) the results. Therefore, the corpus interface usually also offers the function of converting the results into a simplified textual form, convenient for printing. This is also a type of export - the system adapts the page to the printer or downloads it as PDF. It is noted as historical information that in the 2005 version of the Open Source Shakespeare corpus mentioned above, users were given the opportunity to save and print search results. Therefore, the demand for preserving and, if necessary, documenting the results has existed for a long time among corpus users [12].

The software part of the system uses specific components to implement the above-mentioned "storage" methods. For example, for CSV/Excel export, the results are first converted to an array or table and formatted with the necessary delimiters (for example, commas or semicolons for CSV). For JSON export, the results are converted to a structured object (for example, a list of dictionaries in Python or an array of objects in JavaScript) and JSON serialized. The user profiling storage mechanism, on the server side, records the query text and a link to the result set, associated with the user ID - such data is usually stored in the database.

Lexical search parameters. Lexical search parameters are settings in a corpus search system that enable the user to impose lexical-oriented constraints and filters between language units in the query. These include lemma, morpheme, word class (part of speech), affix, synonym series, collocation and semantic field. With those parameters, the user can more accurately guide the search and analyse language phenomena in greater depth. Below, the content of each parameter, the methods of their application in the search, and the issues of programmatic implementation are considered in detail [13].

Search by lemma. **A lemma** is the lexical base form of a word, the initial (neutral) form, usually corresponding to the infinitive or singular form. Searching the corpus by lemma allows the user to search for all the different forms of a word in one go. This is especially important for a rich agglutinative language like Uzbek, since one lexeme (for example, a verb) can occur in dozens of different forms. When searching by lemma, the user enters a lexical form, and the system finds all its grammatical forms (as a result of inflection, declension) and returns results. For lemmatic search to work, the corpus system must first associate words in the texts with their lemmas. This requires that the texts be morphologically tagged or that online lemmatization be performed during the search process. In many corpora, the primary solution is to automatically analyze texts and add a lemma tag to each word. Within the framework of the national corpus of the Uzbek language, all possible forms of a word are entered into the dictionary through a special morphological analyzer and the main form identifier of the word is attached. Thus, during the search, it becomes possible to quickly search for the lexeme entered by the user, and to search for words indexed by dictionary form [14].

Lemma search is a very convenient tool for the user. For example, a researcher who searches for the word *"yurgan"* in a simple search will see only the results of *"yurgan"*. However, in a search by lemma, as a result of searching for the lemma *"yur-" (or "yurmoq")*, all forms such as *"yurdi"*, *"yuribdi"*, *"yurayog"*, *"yuradir"* can be found. The example of the English-Corpora.org website provides information about the existence of methods such as

searching for a lemma by writing it in capital letters in English. It is also desirable to create an analogous simplicity for the user in Uzbek. For lemma search, you can enter a word by setting a special character (pre-programmed) or by selecting the "Lemma" option in the search form. To solve this, the Russian National Corpus interface has a separate button called "Лемма" - if the user selects the "Лемма" option when entering a word, the system searches for all word forms of this lemma.

From the point of view of programming lemmatic search, the indexing mechanism occupies an important place. As mentioned above, if the texts are analyzed in advance and written to a concordance table associated with the lemma of each word, the search works very quickly. The concordance system created by the researchers of TashSULLU also used a similar approach: they divided all the texts into words using a special script controlled by the Django platform and wrote each of them to the database with its lemma, context, and position. As a result, when the user enters a search term, a matching lemma is found through the Uzbek dictionary (list of lemmas) of more than 85 thousand words preloaded into the system, and all matching word forms are retrieved from the concordance table by the ID of this lemma. This method works very quickly in real time, since all matches are obtained through a ready-made index [15].

Lemma search is widely used in linguistic research. It is convenient for observations within the paradigm, that is, for viewing all forms of a lexeme. For example, in what contexts a particular noun under study is used in different conjugations or the frequency of use of verb tenses and moods can be determined by searching the corpus for lemmas. If the corpus has a statistical module, it is also possible to create tables analyzing the results of lemma search in terms of different grammatical forms (for example, what percentage of verbs are used depending on tenses).

From a programming point of view, stemming methods can also be used to improve lemma search. **Stemming** is a simple cutting off of affixes at the end of a word and reducing it to the root. However, stemming sometimes leads to errors (the words "oshdi" and "oshdiq" can both have the stem "osh-", but one of them is the verb "oshmoq", and the other is the noun "oshiq"). Therefore, in modern corpus linguistics, lemma (full morphological analysis and dictionary linking) is preferred. Given the complex morphology of the Uzbek language, the adaptation of foreign developments for lemma search is important. If the corpus is integrated with such a lemmaizer, when the user enters a word, it can first be brought to the lemma (or several possible lemmas are found) in the background, and then the search can be performed.

4. Conclusion

The article substantiates the necessity of creating efficient search mechanisms for the Uzbek language corpus and outlines effective ways to organize them. It emphasizes that sorting and storing search results are essential for ensuring the systematic study of linguistic phenomena, allowing users to observe language patterns, frequency distributions, and collocational relationships with greater precision. The integration of lemma-based and morpheme-based searches, along with search options by word class, affix, synonym, collocation, and semantic field, provides linguists with advanced analytical tools for exploring lexical, morphological, and semantic structures in depth. Apart from facilitating data access, such mechanisms also provide a basis for increased reliability and reproducibility in linguistic research. Functions for exporting and saving history makes long-term research management possible, with the ability to store, compare and statistically analyze data throughout your research. In addition, the knowledge from well-established corpora, such Russian National Corpus and English-Corpora. This research implements approaches from global best practice organisations to the complex morphology and syntax of the Uzbek language (https://nlp.iitb.ac.in/) These results show that the quality of the search system plays a fundamental role in improving digital linguistics and the automation of language processing, as well as Corpus-based language pedagogy. As a result, such a system fosters the growth of computational linguistics in Uzbekistan, promotes cross-disciplinary interaction, and guarantees that the Uzbek

language corpus can be a working corpus for research, education, and technological applications of natural language processing in the country.

REFERENCES

[1] L. Anthony, «AntConc (Windows, Macintosh OS X, and Linux)». 2011 г.

[2] T. McEnery и A. Hardie, *Corpus Linguistics: Method, Theory and Practice*. Cambridge: Cambridge University Press, 2012. doi: 10.1017/CBO9780511981392.

[3] J. Sinclair, *Corpus, Concordance, Collocation*. Oxford: Oxford University Press, 1991.

[4] N. Abdurakhmonova, «Formal-Functional Models of the Uzbek Electron Corpus», *ANGLISTICUM. Journal of the Association-Institute for English Language and American Studies*, т. 10, вып. 8, сс. 59–66, 2021.

[5] N. Abdurakhmonova, «Formal-Functional Models of the Uzbek Electron Corpus», *ANGLISTICUM. Journal of the Association-Institute for English Language and American Studies*, т. 10, вып. 8, сс. 59–66, 2021.

[6] Hermetic Systems, «Hermetic Concordance Software». 2023 г.

[7] V. Zaxarov, B. Mengliyev, и Sh. Xamroyeva, *Korpus Lingvistikasi / Corpus Linguistics: A Textbook*. Tashkent: GlobeEdit, 2021.

[8] Russian National Corpus, «Manual and Settings Page of the Russian National Corpus». 2023 г.

[9] S. Baxodirov и N. Muradova, «Mualliflik Korpusi Qidiruv Tizimini Ishlab Chiqish / Development of the Author Corpus Search System», *Computer Linguistics: Problems, Solutions, Prospects*, т. 1, вып. 1, 2025.

[10] S. Baxodirov и N. Muradova, «Mualliflik Korpusi Qidiruv Tizimini Ishlab Chiqish / Development of the Author Corpus Search System», *Computer Linguistics: Problems, Solutions, Prospects*, т. 1, вып. 1, 2025.

[11] QSR International, «NVivo Help: Word Frequency Queries». 2023 г.

[12] Russian National Corpus, «Search Interface of the Russian National Corpus». 2023 г.

[13] P. Baker, «Sociolinguistics and Corpus Linguistics», *Journal of Language and Society*, т. 39, вып. 3, сс. 345–368, 2010, doi: 10.1177/0261927X10365823.

[14] A. Kilgarriff *и др.*, «The Sketch Engine: Ten Years On», *Lexicography*, т. 1, вып. 1, сс. 7–36, 2014, doi: 10.1007/s40607-014-0009-9.

[15] English-Corpora.org, «Word and Phrase Search Help». 2023 г.