TECHNOLOGICAL BASIS OF FORMING UZBEK-ENGLISH PARALLEL CORPUS AND ITS ROLE IN SENTIMENT ANALYSIS

Amirkulov Ma’rufjon Alikulovich,
2nd year Doctoral Student
amirkulov.edu01@gmail.com
TSUULL

Abstract: This article provides a detailed analysis of the technological foundations and stages involved in building language corpora. The corpus creation process includes collecting texts, cleaning and formatting them, annotating with linguistic tags, organizing and indexing the database, and offering a user-friendly interface and analysis tools. Text collection utilizes diverse sources such as websites, digital documents, and audio speech transcriptions. Subsequently, texts are cleaned, standardized, tagged, and indexed. This process serves as a crucial basis for in-depth linguistic analysis and research in the field of corpus linguistics.

Keywords: *corpus creation, text collection, text cleaning, annotation, indexing.*

Annotatsiya: Ushbu maqolada til korpuslarini yaratishning texnologik asoslari va bosqichlari batafsil tahlil qilinadi. Korpus yaratish jarayoni matnlarni yig‘ish, ularni tozalash va formatlash, lingvistik teglar bilan belgilash (annotatsiya qilish), ma’lumotlar bazasini tashkil etish va indekslash, shuningdek, foydalanuvchiga qulay interfeys hamda tahlil vositalarini taqdim etishni o‘z ichiga oladi. Matnlarni yig‘ishda veb-saytlar, raqamli hujjatlar va og‘zaki nutqning transkripsiyalari kabi turli manbalardan foydalaniladi. Keyingi bosqichda matnlar tozalanadi, standartlashtiriladi, teglar bilan boyitiladi va indekslanadi. Mazkur jarayon korpus lingvistikasi sohasida chuqur lingvistik tahlil va ilmiy tadqiqotlar olib borish uchun muhim asos bo‘lib xizmat qiladi.

Kalit so‘zlar: *korpus yaratish, matn yig‘ish, matnni tozalash, annotatsiya, indekslash.*

INTRODUCTION

As a result of the integration of modern linguistics and computer sciences, the creation of corpora has become an effective tool for deep language analysis and acquiring new knowledge. The formation of corpora is a complex, multi-stage information-technological process, in which texts obtained from various sources are collected, cleaned, annotated, and structured. This process provides specific methodological foundations for linguists, linguistic analysts, and computer scientists.

The main stages of corpus creation include text collection, cleaning and formatting, annotation with tags, database formation, and indexing. Each stage requires a specific technological approach. In the process of text collection, various sources such as digital documents, web pages, and audio transcriptions can be used. Afterwards, the collected texts are cleaned and normalized into a unified structure and encoding. In the tagging (annotation) stage, the morphological, syntactic, and semantic aspects of the texts are marked. At the same time, the texts are indexed in special databases, and a user-friendly interface and analysis tools are provided.

As a result of the integration of modern linguistics and computer sciences, the creation of corpora has become an effective tool for deep language analysis and acquiring new knowledge. The formation of corpora is a complex, multi-stage information-technological process, in which texts obtained from various sources are collected, cleaned, annotated, and structured [5:1-5]. This process provides specific methodological foundations for linguists, linguistic analysts, and computer scientists and shapes new approaches in modern linguistics [6:558-564]. Furthermore, high-quality monolingual corpora process may be an appropriate basis for creating high-quality parallel corpora as well. For instance, based on worldwide knowledge, the theoretical foundations of the Uzbek-English parallel corpus have been developed [6:558-564], [7:388-391].

TECHNOLOGICAL BASIS OF CREATING CORPORA

Corpus creation is a multi-stage information-technological process. Several studies have been counducted on the processing of corpus texts worldwide. This is true for the Uzbek language as well [8:63-68], [9:57-62]. The following main stages can be listed for modern corpora:

Collecting texts (forming a database): At this stage, as many texts as possible in the target language are collected from various sources. These sources include digital documents (web pages, e-books, journals), digitized texts (for example, texts obtained through OCR from scanned books), and speech transcriptions from audio recordings. For instance, using special web crawler programs, thousands of pages from a specific domain can be downloaded from the internet [1:9]. Or books from a library’s collection are scanned, and their text is recognized using an OCR (Optical Character Recognition) program. In this process, the practice of web scraping—automatically extracting text from the web using programming languages such as Python—is widely used. For example, the BeautifulSoup package is used to extract text from HTML pages, pdfminer is used to convert PDF documents to text, and so on.

1. **Cleaning and formatting texts:** Once the raw material (texts) has been collected, the parts unnecessary for the corpus are removed. Unnecessary parts may include, for example, navigation menus on web pages, advertisements, script codes that are irrelevant to language, or errors and symbols resulting from OCR. Texts are cleaned using special scripts: extra spaces and symbols are removed, and encoding is standardized (for example, the Cyrillic–Latin alphabet issue is resolved). Each document in the texts is brought to a uniform structure. Titles, indents, and paragraphs are identified. If the corpus has multiple sources, each text is assigned a unique ID to identify its source. At this stage, it is important to convert the texts to a

single encoding standard: UTF-8 is an international standard that facilitates storing multiple languages in one file.

2. **Tagging (annotation) stage:** This is a very important step for the theoretical value of the corpus. Tagging refers to assigning linguistic tags and metadata to language units in the text. The simplest annotation is morphological tagging, which means assigning tags about the part of speech and grammatical form to each word (for example, *kitob/N* – noun, *yaxshi/JJ* – adjective, *o‘qi/VB* – verb). In English, POS tagging (Part-of-Speech tagging) is performed with high accuracy using automatic tools such as CLAWS, TreeTagger, and spaCy. In Uzbek, initial taggers are also being developed [2:57-62]. Tagging is not limited to morphology only; syntactic analysis (parsing) can also be performed, and sentence structures can be marked in the form of syntactic trees (a corpus of this kind is called a treebank) [3:1]. In addition, semantic tags [4:1] can be assigned to the corpus—for example, a specific domain or topic code (if the text’s topic is specified), stylistic style (formal, informal), and so on. All these annotations are embedded into the text as special codes. For this purpose, the international TEI (Text Encoding Initiative) standard is used: TEI defines an XML tag system for tagging texts, and these tags are added into the text code.

3. **Database and indexing:** The tagged texts form the central database of the corpus. This database can be stored as a simple collection of files, but for fast search, a special index is usually built. An index is a table that shows in which document and at which lines all word forms (or lemmas) occur in the text. In modern corpus platforms, indexing is carried out using the inverted index method, which is similar to the principle used in search engines. Thanks to the index, it becomes possible to search for any word or combination in the corpus within seconds. Otherwise, millions of words would have to be read from beginning to end for each query. In this process, corpus developers sometimes use database technologies as

well—for example, SQLite, SQLServer, or NoSQL databases. However, formats specially designed for corpora are more commonly used, such as CWB (Corpus WorkBench). CWB stores texts in .vrt (vertical format) and builds an index for it. Alternatively, Sketch Engine uses its own indexing mechanism called Manatee.

4. **User interface and analysis tools:** A convenient software environment is required to work with the finished corpus. This environment must allow the user to enter queries (for example, to search for a word or phrase) and to view and analyze the results. In corpus linguistics, the traditional tool is a concordancer. A concordance result is a list of sentences that contain the searched word, each showing the word in the context of 5–7 surrounding words (Key Word in Context, KWIC format). For example, if the word *ilm* ("science") is searched in the corpus, the output may include lines like: “ilm …… kuchdir” (“knowledge is power”), “..... ilm egallash .....” (“to acquire knowledge”), and so on. This reveals which terms and verbs the word commonly co-occurs with. The interface also offers filtering and sorting options for such concordances. For example, it can allow searching only within literary texts or only for specific grammatical forms of a word (e.g., only occurrences in the accusative case). A special search language called CQL (Corpus Query Language) is also used. In this language, for example, the query [word="til"] [] [pos="V"] finds constructions where “til” (language) is followed by any one word, and then a verb—such as *til o‘rganmoq* (“to learn a language”), *til o‘rgatmoq* (“to teach a language”), etc. The user interface must also be able to generate quick statistics based on search results: how many times a word occurs, in which years it is most frequently used (if the corpus is diachronic), with which words it most frequently collocates, and so on. Advanced systems such as Sketch Engine, for example, automatically generate a “Word Sketch” report for an input word. This report shows with which objects or modifiers the word appears in various syntactic roles, and its most typical collocations are presented in tables and graphs. Such

analytical tools accelerate the process of extracting new linguistic insights from the corpus. The technological foundations, stages, and goals of building language corpora are summarized in the following Table 1.

Table 1. Technological basis of creating corpora

Process stage / goal	Common technologies in English corpora	Available / experimental stages in Uzbek corpora	Comparative analysis & specific challenges
1. Data collection (web crawl, e-library, audio)	Common Crawl, wget, Scrapy, Wayback Machine, LDC catalogs	Python scripts (BeautifulSoup, Requests), state web-portal and media crawlers for uzbekcorpus.uz; National Library scans	In English, annual petabyte-scale global web scraping exists; in Uzbek, fewer domains and most content requires PDF → OCR.
2. OCR / transcription	ABBYY FineReader, Tesseract (eng.traineddata), Vosk-ASR (speech)	Tesseract (tos.traineddata, cyr.traineddata), Olam OCR; optional ASR models (“Silkroad ASR” prototypes)	OCR accuracy >99% in English; in Uzbek, due to Latin↔Cyrillic transliteration and Arabic historical texts, error rate is high, requiring post-correction.
3. Cleaning and normalization	Boilerpipe, Trafilatura, ICU-langid, Unicode NFC	Cyrillic–Latin converters, deduplicate.py, normalization of mixed-script characters to UTF-8	In English web texts, HTML clutter is the main issue; in Uzbek, priority is encoding (Windows-1251 → UTF-8) and simplifying diacritics in graphics.
4. Segmentation and tokenization	NLTK sent_tokenizer, spaCy tokenizer	UzbekTokenizer (rule-based), UDPipe-Uzbek model, UZNAT token module	Due to Uzbek’s agglutinative nature, token boundaries are more diverse; English scripts are ready-made, but Uzbek needs affix validation.
5. Sentence alignment (parallel sentences)	Gale–Church, Hunalign, Bleualign	Hunalign (with uz–en dictionary), Bleualign; experimental phase of “awesome-align” (mBERT)	English major languages perform well even without dictionaries; Uzbek requires bilingual dictionary, and freer word order means more manual validation.
6. Word alignment	GIZA++ (IBM 1–5), fast_align, awesome-align	GIZA++ (with morph-split on Uzbek side), fast_align; neural awesome-align (pilot)	With morph-split + lemma filters, Uzbek-English accuracy reaches ~85–88%; English models have ready-made parameter files.
7. Morphological/POS tagging	spaCy, Stanford CoreNLP, TreeTagger	UzbekTagger (rule-based); UDPipe-UZ, Stanza-UZ	English models reach 97–98% F1; in Uzbek, with affix-aware stemmer + 12-tag scheme, performance is 92–93% (domain-specific).
8. Syntactic parsing / UD treebank	UD-English-EWT (UDPipe, Stanza), spaCy-dep	UD-Uzbek treebank (1,700 sentences), UDPipe-UZ model	English has a 200K-sentence treebank; Uzbek’s size is small → fine-tuning neural parsers is resource-constrained.
9. Encoding standards & storage	TEI-P5 / XML, CoNLL-U, JSONL; “vertical” (.vrt) + CWB	TEI subset, CoNLL-U (UNC), JSON tokens; uzWaC SketchEngine format	Standards are unified; in Uzbek, TEI document headers (teiHeader) are often shortened.

10. Indexing & database	CWB/CQP, Sketch Engine–Manatee, ElasticSearch	CWB (UNC server), Sketch Engine (uzWaC), Postgres JSONB testing	English corpora have stable 2+ billion word indexes; Uzbek prototypes (gigaword) ~200 million words — CWB is sufficient.
11. User interface	BNCweb, COCA-web (ASP.NET), Sketch Engine SaaS, CQPweb	UzbekCorpus.uz, UNC web-ui (CQPweb fork), Sketch Engine-client	Functionally similar, but English interfaces have mature collocation/graph modules; Uzbek UI currently limited to KWIC + frequency.
12. Licensing / distribution	LDC (paid), OPUS (CC-BY-SA), COCA (commercial), C4 (CC-BY-NC)	UNC (open access + user registration), uzWaC (Sketch Engine → subscription), Alisher Navoi Corpus (CC-BY)	English resources offer a wide spectrum of licenses; in Uzbek, legal status is unclear, and no commercial/open licenses like Porter-style yet.

To illustrate the processes described above in an integrated manner, the following schematic workflow can be envisioned in Figure 1:

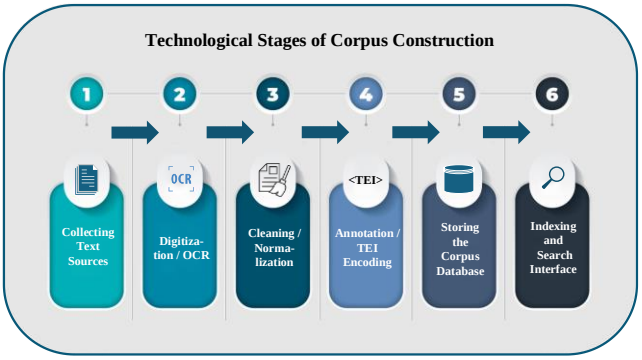

Figure 1. Technological Stages of Corpus Development

In real-world projects, these stages are typically carried out through a combination of automated scripts and manual labor. For example, there exists an open-source toolset known as the “Brno Pipeline,” which includes the SpiderLing web crawler (capable of downloading millions of words from the internet in 24 hours), language identification modules, boilerplate removal tools for filtering recurring code templates in HTML, and automatic components such as tokenizers and POS-taggers. With such ready-made tools, it has become possible to automatically compile multi-billion-word web corpora for specific languages within a few days. For instance, it is known that in the current Uzbek National Corpus project[10:1], over 150 million words of Uzbek text were collected from the internet in 2020 using SpiderLing, in cooperation with Czech researchers.

Of course, corpus creation cannot rely entirely on automation. The OCR stage, in particular, requires a significant amount of manual effort. When scanning and OCR-processing handwritten or printed texts, many errors occur, so filtering software and human supervision are needed for post-OCR editing. In Figure 2, you can see the process:

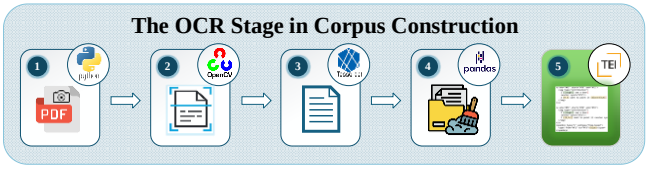

Figure 2. OCR Stage in Corpus Construction

As we can see, programming and language technologies are deeply integrated in corpus creation. Currently, efforts are underway to include historical texts (such as newspaper archives) into the corpus for the Uzbek language using OCR. In this regard, tools such as ABBYY FineReader and Tesseract are being tested.

THE UZBEK-ENGLISH PARALLEL CORPUS AND ITS ROLE IN SENTIMENT ANALYSIS

Sentiment analysis, often referred to as opinion mining, plays a central role in understanding public attitudes, emotions, and opinions expressed through text. While substantial progress has been achieved for high-resource languages such as English, Chinese, or Spanish, low-resource languages like Uzbek continue to face challenges due to the lack of linguistic data and computational tools. The construction of an Uzbek-English parallel corpus offers an effective solution to this problem by providing aligned data that bridges the gap between two languages with distinct grammatical structures and cultural contexts.

Parallel corpora have long been essential for various NLP tasks, including machine translation, semantic similarity measurement, and linguistic research. In recent years, researchers have recognized their growing significance in sentiment

analysis, where aligned bilingual texts can be used to transfer emotional and subjective features from one language to another. In this context, the Uzbek-English parallel corpus not only contributes to bilingual sentiment modeling but also supports broader goals of digital inclusion and linguistic resource development for the Uzbek language.

CONCLUSION

The process of corpus creation has become an essential tool for in-depth language analysis through the integration of linguistics and computer sciences. The technological foundations and stages outlined in the article encompass all key aspects of corpus construction, including text collection, cleaning, annotation with tags, indexing, and the development of a user interface. Each stage requires specific technologies and methodologies, which make the corpus creation process more efficient and accurate.

Corpora are not only significant for linguistic research but also hold great value in fields such as artificial intelligence, machine learning, and other technological domains. Properly constructed corpora enable the acquisition of comprehensive knowledge about various linguistic features. Furthermore, the advancement of corpus development technologies and the introduction of new methods in the future will create more effective opportunities for work in linguistics and other areas.

In conclusion, corpus creation is a fundamental tool necessary for studying not only the theoretical but also the practical aspects of linguistics. Corpora developed using modern technologies and methodologies elevate language analysis to a new level and become resources widely applicable across various fields.

REFERENCES

1. Van Lit, C., & Roorda, D. (2024). Neither Corpus Nor Edition: Building a Pipeline to Make Data Analysis Possible on Medieval Arabic Commentary Traditions. *Journal of Cultural Analytics*, *9*(3).

2. Boltayevich, E. B., Adalı, E., Mirdjonovna, K. S., Xolmo'Minovna, A. O., Yuldashevna, X. Z., & Uktamboy O'g'li, X. N. (2023, September). The Problem of Pos Tagging and Stemming for Agglutinative Languages (Turkish, Uyghur, Uzbek Languages). In *2023 8th International Conference on Computer Science and Engineering (UBMK)* (pp. 57-62). IEEE.

3. Elov, B., & Abdullayeva, O. (2025). Sintaktik parsing turlari. *Computer linguistics: problems, solutions, prospects*, *1*(1).

4. Elov, B., Hamroyeva, Sh., & Hamroqulova, M. (2024). NLPda semantik teglash usullari. *Uzbekistan: Language and Culture*, *1*(1).

5. Biber, D., Conrad, S., & Reppen, R. (1998). *Corpus Linguistics: Investigating Language Structure and Use*. Cambridge University Press,1-5.

6. McEnery, T., & Hardie, A. (2012). *Corpus Linguistics: Method, Theory and Practice* (2nd ed.). Cambridge University Press.

7. B.Mengliyev, S.Shahabitdinova, S.Khamroeva, S.Gulyamova, A.Botirova. (2021). The morphological analysis and synthesis of word forms in the linguistic analyzer. Journal of Language and Linguistic Studies, 17(1), 558-564.

8. K. R.Abdurasulovich, M. B. Rajabovich. (2019). The Role of the Parallel Corpus in Linguistics, the Importance and the Possibilities of Interpretation. International Journal of Engineering and Advanced Technology, 8(5), 388-391.

9. E.B.Boltayevich, S.S.Samariddinovich, K.S.Mirdjonovna, E.Adali, X.Z. Yuldashevna. Pos Taging of Uzbek Text Using Hidden Markov Model. UBMK 2023 - Proceedings: 8th International Conference on Computer Science and Engineering, 2023, P. 63–68.

10. E.B.Boltayevich, E.Adali, K.S.Mirdjonovna, X.Z.Yuldashevna, N.Uktamboy o'g'li, X.The Problem of Pos Tagging and Stemming for Agglutinative Languages (Turkish, Uyghur, Uzbek Languages). UBMK 2023 - Proceedings: 8th International Conference on Computer Science and Engineering, 2023, P. 57–62.

11. https://uzbekcorpus.uz/enParallel