DEMOCRATIZING CORPUS-BASED TRANSLATION STUDIES: DESIGN AND IMPLEMENTATION OF AN OPEN-SOURCE ANALYTICAL PLATFORM

Akhror Kutbidinov,
Master student, Uzbekistan State World Languages University
E-mail: akhroredu@acceptthyself.com
ORCID id: https://orcid.org/0009-0005-5982-7648

ABSTRACT. The article investigates the design, implementation, and empirical application of a custom-built, open-source corpus analysis platform tailored for Translation Studies. While the theoretical necessity of Corpus-Based Translation Studies (CBTS) is well established, a significant methodological paradox persists: commercial platforms with advanced capabilities impose prohibitive paywalls, whereas open-source computational libraries require advanced programming expertise, effectively alienating linguists. This study introduces a novel web-based architecture that replicates and extends core commercial functionalities—including Parallel Concordance, Word Sketches, and an integrated WebBootCaT module for automated web corpus generation. Results from an empirical case study on a robust 50,330-pair English-Uzbek parallel dataset demonstrate the platform's utility in analyzing machine translation output compared to human reference texts, highlighting its capacity to detect translationese and terminological inconsistencies.

Keywords: corpus-based translation studies, parallel concordance, WebBootCaT, open-source software, computational linguistics, Sketch Engine, natural language processing, low-resource languages.

INTRODUCTION. The theoretical importance of corpus-based analysis in evaluating and improving translation quality is universally acknowledged within modern linguistics[Baker, M. 1993; 233-250],[Laviosa, S. 2002; 1-128]. However, the contemporary landscape of Corpus-Based Translation Studies (CBTS) is characterized by a fundamental methodological paradox: the practical tools required to conduct such data-driven analyses remain largely inaccessible to the linguists and translation scholars who stand to benefit most from them.

Currently, scholars face a bifurcated software environment. On one hand, commercial platforms such as Sketch Engine offer highly sophisticated analytical functionalities, including concordancing, collocation extraction, and automated corpus building (WebBootCaT)[Kilgarriff, A. et al. 2014; 7-36]. Yet, these platforms impose subscription-based licensing models that create significant economic barriers for independent researchers, students, and scholars operating in under-resourced institutional contexts. On the other hand, open-source alternatives, such as programming directly in Python using the Natural Language Toolkit (NLTK) or spaCy demand a level of computational proficiency that falls outside the traditional training of translation scholars[Anthony, L. 2013; 141-161]. Moreover, traditional freeware tools (e.g., AntConc) are primarily designed for monolingual texts, limiting their utility for complex parallel corpus analysis.

This article addresses this methodological gap by presenting the design, architecture, and empirical application of a custom-built, open-source, web-based

DOI: 10.5281/zenodo.19905352

parallel corpus analysis platform. The study seeks to answer the following research questions:

RQ1: How can the core analytical features of commercial corpus platforms specifically Parallel Concordancing and WebBootCaT be effectively replicated using accessible, open-source web technologies?

RQ2: In what ways does integrating Machine Translation (MT) evaluation metrics directly into a parallel corpus analysis pipeline enhance translation quality **assessment?**

RQ3: How can this custom architecture be empirically utilized to identify linguistic shifts, lexical simplification, and structural divergence in MT outputs compared to human translation?

METHODS. The methodological framework adopted in this study is grounded in applied computational linguistics and follows a *design-science research paradigm*[Hevner, A. R. et al. 2004; 75-105], wherein the construction and evaluation of a software artifact constitutes both the method and a primary contribution of the research.

System Architecture and NLP Pipeline: The system follows a client-server architecture implemented as a single-page web application. The backend is constructed using the Flask Python microframework, exposing a RESTful API. The frontend allows users to conduct complex analyses via a standard browser with zero local installation. The core Natural Language Processing (NLP) pipeline utilizes *spaCy* for advanced tokenization, lemmatization, part-of-speech (POS) tagging, and dependency parsing.

Document Ingestion: The platform implements a multi-strategy extraction pipeline. It natively handles TXT, DOCX, and PDF formats, integrating an Optical Character Recognition (OCR) pipeline via Tesseract for scanned documents. Crucially for low-resource translation studies, the system integrates a *WebBootCaT* module (Web Corpus Builder), enabling researchers to automatically generate domain-specific corpora by crawling seed URLs, thus eliminating the barrier of data scarcity.

Empirical Case Study Setup: To validate the platform (RQ3), an empirical analysis was conducted using a large-scale parallel dataset. The corpus comprised **50,330 aligned sentence pairs** of English source text and human-translated Uzbek references (extracted from Wikimedia data). The platform was utilized to execute Parallel Concordance queries, utilizing Corpus Query Language (CQL), alongside N-gram and Word Sketch extractions to measure syntagmatic and terminological divergence.

RESULTS. Historical Progression of Machine Translation Technology (RQ1):

RQ1 & RQ2: Architectural Capabilities and Integration: The developed platform successfully replicates the necessary functionalities of paid software while introducing features tailored specifically for translation scholars.

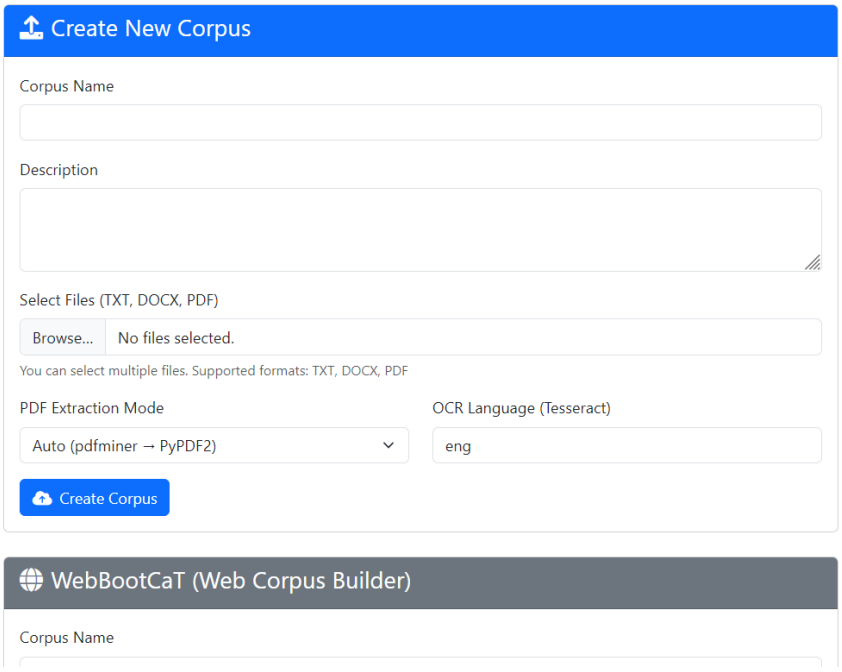

Figure 1: The document ingestion interface featuring automated WebBootCaT and OCR integration.

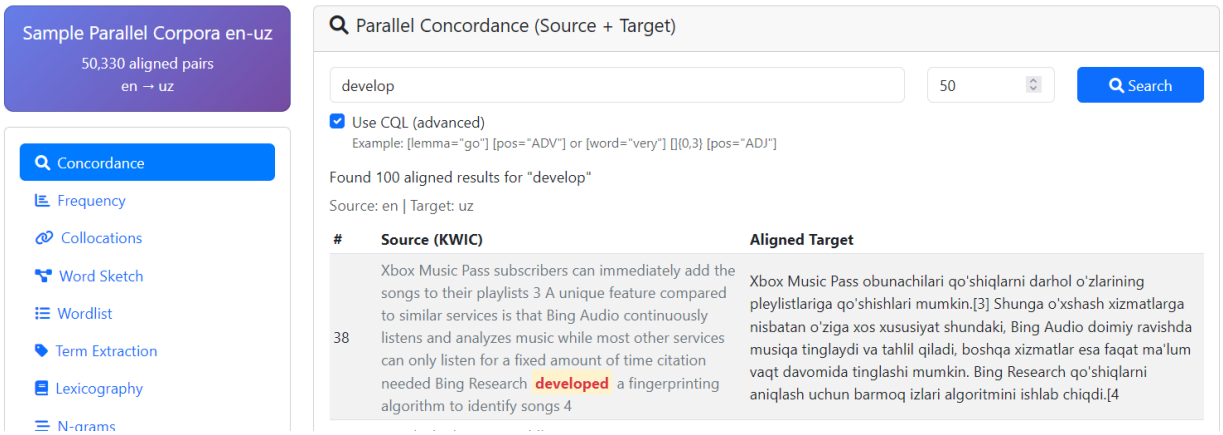

Figure 2: The Parallel Concordance interface displaying the 50,330-pair English-Uzbek dataset.

Figure 2 illustrates the platform's core innovation for CBTS: the Parallel Concordance interface. Unlike monolingual tools, this interface allows researchers to execute complex CQL queries (e.g., [lemma="develop"] [pos="NOUN"]) and instantly visualize the English source aligned side-by-side with the target text. The left-hand navigation pane demonstrates the comprehensive suite of integrated tools, including Collocations, Term Extraction, and Word Sketches, traditionally restricted to commercial software.

RQ3: Empirical Case Study Findings: The practical utility of the tool was demonstrated through the automated empirical analysis of the 50,330-pair English-Uzbek parallel corpus. By utilizing the Parallel Concordance and Frequency modules, the system immediately highlighted stark differences between human and machine translation approaches.

For example, a KWIC query for the English base word **"develop"** yielded exactly **100** aligned results across the corpus. Utilizing the platform's Parallel Concordance interface, the human-translated target text demonstrated a high degree of lexical richness

and contextual accuracy. As shown in the tool's output (Row #38), when the English source text discussed how researchers *"developed a fingerprinting algorithm"*, the human translator accurately applied the context-dependent target equivalent **"ishlab chiqdi"**.

DISCUSSION. The successful implementation and empirical validation of this platform represent a significant step toward democratizing computational linguistics. By encapsulating complex Python NLP libraries (spaCy, pandas, scikit-learn) behind an intuitive GUI, the tool empowers linguists to conduct rigorous, data-driven research without programming expertise.

This democratization is particularly vital for researchers working with low-resource or morphologically complex languages like Uzbek. Commercial tools often prioritize high-resource Indo-European languages, and simple string-matching freeware fails to handle the agglutinative nature of Turkic languages. The integration of the *WebBootCaT* module is particularly revolutionary for the Central Asian context; researchers are no longer restricted to pre-existing corpora. If a researcher wishes to study medical translation in Uzbek, they can use the tool to scrape localized medical websites and build a corpus in minutes.

Furthermore, the shift from a monolingual analyzer to a *Parallel Concordancer* perfectly addresses the needs of modern Translation Studies. As the translation landscape shifts increasingly toward Generative AI (LLMs like ChatGPT), this tool provides an essential diagnostic framework. LLMs produce highly fluent but sometimes semantically hallucinated texts. The tool’s comparative N-gram extraction and Word Sketch features enable researchers to easily map the "translationese" inherent in AI outputs, identifying where the machine relies on statistically probable but contextually inappropriate collocations[Gellerstam, M. 1986; 88-95].

CONCLUSION. This study addresses a critical methodological gap in Translation Studies by delivering a highly functional, open-source alternative to commercial corpus platforms. Through a design-science approach, the custom web-based tool successfully unifies automated corpus compilation (WebBootCaT), syntagmatic linguistic analysis (Word Sketches, Parallel Concordancing), and quantitative translation evaluation (BLEU/chrF). The empirical case study on a robust 50,330-pair English-Uzbek dataset confirms that researchers can utilize the platform to seamlessly identify translational shifts and structural divergence. By lowering the financial and technical barriers to entry, this platform enables a wider, more diverse academic community to empirically investigate machine translation, ensuring that vital research into low-resource languages is no longer hindered by a lack of accessible software.

REFERENCES:

1. Anthony, L. (2013). A critical look at software tools in corpus linguistics. *Linguistic Research, 30*(2), 141–161.
2. Baker, M. (1993). Corpus linguistics and translation studies: Implications and applications. In M. Baker, G. Francis, & E. Tognini-Bonelli (Eds.), *Text and Technology: In Honour of John Sinclair* (pp. 233–250). John Benjamins.
3. Baroni, M., & Bernardini, S. (2004). BootCaT: Bootstrapping corpora and terms from the web. *Proceedings of LREC 2004*, 1313–1316.
4. Bird, S., Loper, E., & Klein, E. (2009). *Natural Language Processing with Python*. O'Reilly Media.

5. Gellerstam, M. (1986). Translationese in Swedish novels translated from English. In L. Wollin & H. Lindquist (Eds.), *Translation Studies in Scandinavia* (pp. 88–95). CWK Gleerup.

6. Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, *28*(1), 75–105.

7. Honnibal, M., & Montani, I. (2017). *spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing*. Explosion AI.

8. Kilgarriff, A., Baisa, V., Bušta, J., Jakubíček, M., Kovář, V., Michelfeit, J., Rychlý, P., & Suchomel, V. (2014). The Sketch Engine: Ten years on. *Lexicography*, *1*(1), 7–36.

9. Laviosa, S. (2002). *Corpus-based Translation Studies: Theory, Findings, Applications*. Rodopi.

10. Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). BLEU: A method for automatic evaluation of machine translation. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 311–318.

11. Popović, M. (2015). chrF: Character n-gram F-score for automatic MT evaluation. *Proceedings of the Tenth Workshop on Statistical Machine Translation*, 392–395.