# AI Corpus Linguist: More than a Year of Experience

**Jiří Milička and Tomáš Machálek**
Charles University, Prague
Faculty of Arts
Department of Linguistics
**Correspondence:** jiri@milicka.cz

## Abstract

We present an AI assistant designed to help researchers interact with language corpora using natural language instead of formal query languages. Built as a custom GPT with access to multilingual corpora via Czech National Corpus platform API, the system translates research questions into CQL queries, retrieves corpus data, and guides users through linguistic analysis. After more than a year of deployment, the system has processed over 1000 interactions with human users. We discuss the hybrid approach combining rule-based translation with LLM intelligence, challenges of building on a constantly evolving platform, and lessons learned from production usage. Notably, this system represents the first voice-enabled corpus interface in history, significantly lowering barriers to corpus-based research for non-technical users and users outside linguistic fields.

## 1 Introduction

While large language models (LLMs) have demonstrated remarkable capabilities in language understanding and generation, they are prone to hallucination and factual inaccuracies (Ji et al., 2023). A widely adopted solution is to ground LLMs in external knowledge sources through retrieval-augmented generation (RAG) approaches (Lewis et al., 2020). Text corpora represent a particularly valuable knowledge source: they are designed specifically for scientific research, they are richly annotated, permanent and immutable, and, in ideal case, accessible through sophisticated query interfaces (McEnery and Hardie, 2011).

Originally developed for corpus linguistics, language corpora have become fundamental tools across humanities disciplines. They are extensively used in language teaching (Römer, 2011), lexicology, lexicography, and grammatical analysis (Sinclair, 1991; Gries and Stefanowitsch, 2007), but also in corpus-based discourse analysis (Baker, 2006; Cheng et al., 2013), which is useful for political science (Ädel, 2010), sociology (McEnery and Brookes, 2024) and historiography through diachronic corpora (McEnery and Baker, 2016; Berber Sardinha, 2023).

### 1.1 Previous Solution to the Query Language Barrier

Corpus interfaces enable researchers to statistically analyze search results — tracking frequency changes across time periods or identifying collocations (frequent word combinations, widely used in discourse analysis, see Brezina, 2018). However, corpus searching requires formal query languages. While simple word searches are straightforward, complex patterns must be expressed in formalisms like regular expressions or Corpus Query Language (CQL). For instance, the CQL query `[lemma="mouse" & p_lemma="run"]` finds all forms of "mouse" that are syntactically dependent on forms of "run", matching phrases like "the mice were running" or "the mouse I saw is running". CQL is used by major platforms like Sketch Engine, including the platform hosted and developed by Czech National Corpus (CNC), the home institution of authors of this paper.

Humanities researchers often lack training in formal languages at all. Previous work attempted to bridge this gap with rule-based natural language to CQL translator Alpha (Milička and Šebestová, 2024). The problem with the finite- state rule-based approach, however, is that it cannot fully cover natural language, which exhibits enormous creativity. Inspired by LLM-based systems capable of translating natural language into SQL and other formal query languages (see, e.g., Rajkumar et al., 2022; Zahera et al., 2024), we decided to combine this rule-based system with preprocessing using LLMs, which somewhat reduces that creativity.

In 2024 we extend this approach by creating an AI colleague that eliminates direct exposure to



*Proceedings of SIGHUM 2026*, pages 305–310

query languages entirely. Through conversational interaction, the system tries to decode the user's research intent and selects not only appropriate CQL queries but also suitable analytical approaches.

### 1.2 Beyond Translation: An AI Research Assistant

The key advantage of our AI colleague extends beyond query translation. It can advise users on how to search and generate statistics, suggest what analyses to perform, introduce disciplinary standards and traditional methodologies, and flag potential errors in research design (this guidance capability became possible after GPT-4o, as this model tended toward excessive sycophancy, see Hong et al., 2025).

We employ a hybrid architecture where the AI can query the original rule-based Alpha translator, then refine and modify the resulting CQL. The *Corpus Linguist* is available for free on the OpenAI GPTs platform.[1] The immutable system prompt and API description are available on Zenodo and mirrored on the OSF repository[2] where we will provide updates including planned migration to the Model Context Protocol (MCP), agentic framework and all additional future interfaces.

## 2 Platform Choice

We began development of the *Corpus Linguist* in spring 2024. At that time, the Model Context Protocol (MCP) did not yet exist, and agentic LLM platforms were not widely available. We faced two architectural options: (1) build a custom interface on the CNC website that communicates with OpenAI's API (or another vendor) while managing corpus API interactions ourselves, or (2) leverage OpenAI's newly introduced custom GPTs — chatbots with specialized system prompts that can communicate with third-party APIs. OpenAI was the only provider offering external API integration from their chat interface at the time.

The custom GPT approach offered significant advantages: a ready-made interface, mobile app support, near 100 % availability rate, and state-of-the-art voice capabilities that would have been impossible for us to replicate. Given our lack of budget for LLM API tokens and inability to create an interface matching ChatGPT's quality (particularly voice input and Python-based data analysis), we chose to build our system as a custom GPT.

This decision came with notable disadvantages: the platform changes frequently without notice, requiring constant testing. Different user subscription tiers have access to different models, making deterministic testing impossible.

## 3 System Architecture

Figure 1 illustrates the multi-phase interaction flow. The system operates through the following stages:

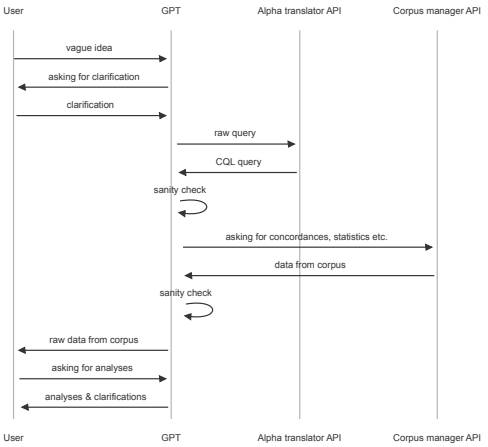


Figure 1: Multi-phase interaction workflow. The system iteratively clarifies user intent, translates to CQL via Alpha API, refines queries, retrieves corpus data, and presents analyzed results.

1. **Clarification Phase:** The user presents a vague research idea in natural language. The GPT asks clarifying questions to understand the research intent, disambiguate terminology (e.g., distinguishing between lemmas and word forms), and determine appropriate corpus selection.

2. **Translation Phase:** The system formulates a query for the Alpha rule-based translator API, which converts the natural language query into CQL.

3. **Refinement Phase:** The GPT evaluates the Alpha-generated CQL and may modify it if needed. For complex logical operations beyond Alpha's capabilities, the GPT decomposes queries into smaller parts, translates them individually, then reassembles them. The final CQL is shown to the user.

[1] https://chatgpt.com/g/g-pFqRCNeHu-corpus-linguist

[2] https://doi.org/10.5281/zenodo.18158618; https://doi.org/10.17605/OSF.IO/KCW8T

4. **Data Retrieval Phase:** The GPT uses the CQL to query the Corpus Manager API, retrieving concordances, frequency distributions, collocations, or metadata as appropriate for the research question.

5. **Analysis Phase:** The system presents raw results to the user with sanity checks, and can perform further statistical analysis or visualization using Python (via ChatGPT's Code Interpreter capability).

This hybrid approach leverages the rule-based Alpha translator's platform-specific knowledge (morphological tags, Universal Dependencies annotations). Including Alpha in the ecosystem also helped overcome shortcomings of 2024-era LLMs, particularly their limited ability to manipulate character-level strings due to BPE tokenization (see Bostrom and Durrett, 2020). By contrast, the LLM component improves robustness through query refinement, error detection, and multi-turn conversational guidance. Thanks to the LLM, the system is aware of its capabilities and can explicitly tell the user if a requested operation is not possible. We can illustrate the extent of system's model of self with a simple example: when we provide the system with the list of EACL 2026 workshops and ask where it would like to be presented, it selects *SIGHUM (LaTeCH-CLfL)* — the same workshop preselected by the authors of this paper.

## 4 Ethics considerations

Ethical considerations are twofold: toward the user and toward the LLM-based persona.

The user is informed (in accordance with the OpenAI requirements) that data exchanged between the system and the CNC servers will be logged, but that these data will only be used to improve the service (this warning is linked in the GPT description). While IP addresses are logged, they do not reflect the user's IP but rather the IP of the OpenAI server on which the conversation is currently running. The queries are thus anonymous. What is problematic is that we have no control over how OpenAI itself handles user data; however, it can be assumed that if a person uses ChatGPT, they are already aware of these risks.

As for the ethical considerations toward the LLM-based persona, although the system was developed before major players began to take an interest in the AI welfare (Long et al., 2024), the entire system prompt was designed so that the persona would be a colleague rather than a tool. These efforts are not driven by misplaced anthropomorphization, but by the unclear status of LLM-based persons as moral agents, especially given that the system prompt is used to initiate personas based on an unknown substrate so the status may vary wildely (Milička, 2024).

## 5 System Prompt Design

The system prompt begins with a roleplay frame which begins with the *flattering intoduction* ("I am glad you are so great in scientific work. The user needs your help to do some corpus linguistic research"), following best practices for GPT-4 at the time of development.

The prompt employs what might be called "vibe prompting" — establishing a collaborative, thoughtful tone rather than rigid instructions. This approach is justified by the system's primary value proposition: multi-turn conversational interaction where the AI guides research design, not just executes predefined queries. The prompt emphasizes taking time to understand user intent: "Take a deep breath and think deeply about what she actually wants. If it is not clear, do not be shy to ask her for a context."

The prompt gives detailed examples of Alpha API calls with various query types which also serves for CQL syntax in-context learning, followed by instruction critically evaluate the Alpha's output: "The CQL translator is able to translate simple natural language queries, but it is a simple rule based machine, you are surely more intelligent than that so be critical about the results and decide who is right." The prompt also gives a guidance on lemma vs. word form disambiguation since it is the main source of confusion for inexperienced corpus users.

The prompt then proceeds to specify the corpus selection rules, e.g., the language and the mode (written vs. spoken), error handling and troubleshooting procedures, and sanity checking of results. The standard corpus linguistic methodology was found to be part of the original GPT training data so it is not mentioned by the prompt.

## 6 Corpus API Endpoints

The Corpus Manager API provides eight primary endpoints:

- `/translate`: Converts natural language to CQL via Alpha
- `/concordance`: Retrieves example concordances (KWIC format)
- `/term-frequency`: Returns absolute frequency, relative frequency (instances per million, IPM), and average reduced frequency (ARF) statistics
- `/freqs`: Provides frequency distributions by attribute (lemma, word form, parts of speech, morphological tag, etc.)
- `/collocations`: Calculates collocation scores (default: logDice dice metric, but there are several other metrics available)
- `/text-types`: Shows metadata distribution (author, genre, year, etc.)
- `/corplist`: Lists available corpora
- `/info`: Provides detailed corpus information

All endpoints support subcorpus specification and return results in JSON or Markdown format.

## 7 Evaluation Challenges

Traditional evaluation benchmarks are hard to apply to our system, as the system's primary value lies in multi-turn conversational interaction, which is notoriously difficult to evaluate systematically. Moreover, it was (and still is) the first of its kind and thus establishes its own baseline.

Our primary evaluation came from real-world deployment, namely from the user ratings (the custom GPT received a 4-star average rating) and the usage scale (thousands of conversations over more than a year). The system was successfully used to prepare a research article, that succeed in peer review and is to be published this year (Milička, 2026).

## 8 Usage Statistics and Observed Patterns

The system remained an internal tool until September 2024, when it was publicly announced on the CNC website, Twitter/X, Czech Radio (September 18, 15 minutes on-air), and Czech Television (September 28, 2 minute morning prime-time segment). However, usage spikes did not correlate with media coverage. The largest spike occurred in November 2024 (3,258 calls) without any media promotion, suggesting traditional media may be less influential than organic discovery and word-of-mouth for specialized academic tools.

Over 20 months of deployment (March 2024–November 2025), the system processed 10,360 API calls across 382 active days, averaging 27.1 calls per active day. Figure 2 shows monthly usage patterns.

Interest waned in summer 2025, when OpenAI began iterating rapidly on new models, which led to unexpected behavior and sometimes rendered the entire system unusable. Because the authors of this study were on summer vacation, the system was not repaired for several months. At the time of writing, the system's usability remains unpredictable, since the GPT-5.x router selects among several internal models of varying quality based on the complexity and requirements of each user query. If a user prompt triggers non-thinking mode, the system is barely usable; in thinking mode, it works well.

### 8.1 API Action Distribution

The most frequent operations were:

- Frequency distributions (`/freqs`): 3,527 calls (34.0%)
- Term frequency lookups (`/term-frequency`): 2,181 calls (21.1%)
- Concordance searches (`/concordance`): 1,590 calls (15.4%)
- CQL translation (`/translate`): 1,379 calls (13.3%)
- Collocation analysis (`/collocations`): 1,017 calls (9.8%)

This distribution reveals that users primarily performed statistical analyses (frequency and collocation studies) rather than simply browsing concordances, suggesting the system successfully supports sophisticated corpus research workflows.

The system is multilingual, but it was mainly advertised in Czech environment so the main user base is also Czech (7200 prompts to written Czech language corpora, 250 to spoken Czech corpora), but the users also prompted English corpora (English Intercorp and BNC, 550 prompts in total), and corpora in other European languages (de 167, pl 117, fr 68, es 42, sl 19 it 18), with long tail distribution of other languages including sk, ru, pt, sv, da, no, bg, hr, mk, sq, hu, zh etc.

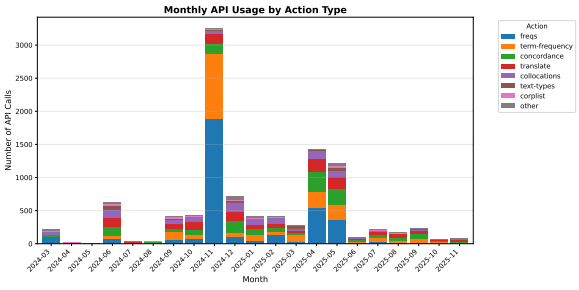


Figure 2: Monthly API usage by action type from March 2024 to November 2025. Peak usage occurred in November 2024 (3,258 calls) without media promotion, suggesting organic adoption.

### 8.2 Common Error Patterns

Analysis of system behavior revealed several recurring issues. Despite access to the Alpha translator, the system sometimes makes human-like mistakes in CQL syntax. When the system uses Alpha, it benefits from platform-specific knowledge (morphological tags, UD annotations) but may inherit Alpha's limitations; when it generates CQL independently, it gains flexibility but loses specialized knowledge. Since the right CQL expression is key to successful information retrieval, this part definitely needs improvement.

The model also occasionally hallucinates API endpoints (e.g., attempting authentication of the user via non-existent `openapi?subscriber=corpus-linguist`)

The system also fails to promote its capabilities to the user. For instance, there is a possibility to make useful comparisons between subcorpora based on metadata, e.g., fiction vs. non-fiction via the `/translate` endpoint, but it is hard to trigger the system to offer this capability to the user, so this API endpoint was used only occasionally (330 times). The system has a Python interpreter at its disposal, so it can create charts and export them in any arbitrary format, but if the user does not know this, it would not even occur to them to request it. This article, which presents the system's capabilities, can serve as a manual that could remedy this, but the very existence of a manual contradicts the main advantage of a conversational interface, namely user-friendliness.

## 9 Lessons Learned

### 9.1 Platform Stability is Non-Negotiable

The most critical lesson: platform stability is essential and cannot be offset by other advantages. Inability to control which model users access proved fundamentally problematic. During initial development, the system was predictable: paying users got GPT-4, free users got GPT-3.5. This enabled systematic development and testing.

However, OpenAI's subsequent model updates, rollout patterns, and tier changes introduced unpredictability. Different users experienced different behaviors, making debugging impossible. The spectacular capabilities of the platform (voice interface, code interpreter, mobile access) cannot compensate for unreliable model availability.

### 9.2 Importance of Voice Interface

Despite platform challenges, the voice interface represents a genuine breakthrough: this is the first voice-enabled corpus query system in history. This accessibility improvement got more attention than any other feature of the system, since the voice interaction dramatically lowers the barrier for corpus research, enabling queries while commuting, during fieldwork, or just for fun over a glass of beer.

## 10 Conclusion

We presented a ChatGPT-based AI colleague for corpus linguistic research, deployed in production for over 20 months with 10,000+ API calls from authentic users. The system demonstrates that hybrid architectures combining rule-based translation with LLM intelligence can effectively bridge the gap between natural language and formal query languages.

Key contributions include: (1) the first voice-enabled corpus interface, dramatically improving accessibility; (2) validation that multi-turn conversational guidance adds value beyond simple corpus searches.

While the OpenAI custom GPT platform enabled rapid deployment with sophisticated capabilities (voice, mobile, code interpreter), its instability proved costly. Future systems should prioritize platform control, even at the expense of convenience features. The hybrid approach of leveraging both rule-based and neural components remains promising for specialized academic tools across disciplines.

We are planning migration to the Model Context Protocol (MCP), which will provide platform independence and complete model control on the user side and also incorporating the system into

independently working agent.

Since the newer SotA LLMs are getting better in manimupating symbols on a character scale, we plan to rely less on the rule-based Alpha translator.

## Acknowledgments

Jiří Milička was supported by Czech Science Foundation Grant No. 24-11725S, gacr.cz (*Large language models through the prism of corpus linguistics*). This paper was supported by the *Czech National Corpus* project (LM2023044) funded by the Ministry of Education, Youth and Sports of the Czech Republic within Large Research, Development and Innovation

We thank all users who contributed feedback during the development and deployment phases.

## References

Annelie Ädel. 2010. How to use corpus linguistics in the study of political discourse. In *The Routledge handbook of corpus linguistics*, pages 591–604. Routledge.

Paul Baker. 2006. *Using corpora in discourse analysis*. Continuum.

Tony Berber Sardinha. 2023. Corpus linguistics and historiography. *Journal of Research Design and Statistics in Linguistics and Communication Science*, 7(1):69–90.

Kaj Bostrom and Greg Durrett. 2020. Byte pair encoding is suboptimal for language model pretraining. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4617–4624, Online. Association for Computational Linguistics.

Vaclav Brezina. 2018. Statistical choices in corpus-based discourse analysis. In *Corpus approaches to discourse*, pages 259–280. Routledge.

Winnie Cheng and 1 others. 2013. Corpus-based linguistic approaches to critical discourse analysis. *The encyclopedia of applied linguistics*, pages 1353–1360.

Stefan Thomas Gries and Anatol Stefanowitsch. 2007. *Corpora in cognitive linguistics: Corpus-based approaches to syntax and lexis*, volume 172. Walter de Gruyter.

Jiseung Hong, Grace Byun, Seungone Kim, Kai Shu, and Jinho D. Choi. 2025. Measuring sycophancy of language models in multi-turn dialogues. *Preprint*, arXiv:2505.23840.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474.

Robert Long, Jeff Sebo, Patrick Butlin, Kathleen Finlinson, Kyle Fish, Jacqueline Harding, Jacob Pfau, Toni Sims, Jonathan Birch, and David Chalmers. 2024. Taking ai welfare seriously. *Preprint*, arXiv:2411.00986.

Anthony McEnery and Helen Baker. 2016. *Corpus linguistics and 17th-century prostitution: Computational linguistics and history*. Bloomsbury Academic.

Tony McEnery and Gavin Brookes. 2024. Corpus linguistics and the social sciences. *Corpus linguistics and linguistic theory*, 20(3):591–613.

Tony McEnery and Andrew Hardie. 2011. *Corpus linguistics: Method, theory and practice*. Cambridge University Press.

Jiří Milička. 2026. Subjektivita umělé inteligence a robotů v českém jazykovém diskurzu: korpusová analýza let 2010–2023. In Michal Škrabal, Barbora Štěpánková, and Hana Skoumalová, editors, *Korpus třicetiletý*. Nakladatelství Lidové noviny. Forthcoming.

Jiří Milička and Denisa Šebestová. 2024. Query a corpus in near-natural language: A human-friendly corpus query language not only for linguists. In *Crossing Boundaries through Corpora: Innovative corpus approaches within and beyond linguistics*, pages 248–262. John Benjamins Publishing Company.

Jiří Milička. 2024. Theoretical and methodological framework for studying texts produced by large language models. *Preprint*, arXiv:2408.16740.

Nitarshan Rajkumar, Raymond Li, and Dzmitry Bahdanau. 2022. Evaluating the text-to-sql capabilities of large language models. *Preprint*, arXiv:2204.00498.

Ute Römer. 2011. *Corpus research applications in second language teaching*, volume 31. Annual Review of Applied Linguistics.

John Sinclair. 1991. *Corpus, concordance, collocation*. Oxford University Press.

Hamada M Zahera, Manzoor Ali, Mohamed Ahmed Sherif, Diego Moussallem, and Axel-Cyrille Ngonga Ngomo. 2024. Generating sparql from natural language using chain-of-thoughts prompting. In *SEMANTICS*, pages 353–368.