



# Ilja Barouski

MACHINE LEARNING/SOFTWARE ENGINEER

Žodzina, Belarus

+375-44-450-69-51 | [iljabarouski@gmail.com](mailto:iljabarouski@gmail.com) | [scurra.github.io](https://scurra.github.io) | [Scurra](#) | [iljabarouski](#) | [iljabarouski](#)

## Education

---

### BSU (Belarusian State University), Faculty of Mechanics and Mathematics, Computer Mathematics and System Analysis

BACHELOR OF MATHEMATICS AND SYSTEM ANALYSIS

Minsk, Belarus

Sep. 2019 - Jun. 2023

### BSU (Belarusian State University), Faculty of Mechanics and Mathematics, Computer Mathematics and System Analysis

MASTER OF MATHEMATICS AND COMPUTER SCIENCE

Minsk, Belarus

Sep. 2023 - Jun. 2025

## Skills

---

**Programming** Python, Julia, Rust, C, C++, Crystal, C#, Dart, Go, LaTeX, Wolfram

**Databases** PostgreSQL, MongoDB, SurrealDB

**Machine Learning** Data Visualization, Regression, Decision Trees, Neural Networks, Ensembles, Tokenization, NLP

**Languages** English (B2-C1), Bielarskaja (Native), Русский

## Pet Projects

---

### UBPE: Universal Byte-Pair Encoding Tokenizer

[github repo](#), [pypi](#), [google colab](#)

Jun. 2025 - Present

STACK: PYTHON, C++

A flexible and efficient tokenizer for general sequences, extending traditional BPE beyond text.

Key Features:

- *General-Purpose Tokenization*: Works with text, numerical sequences, or any custom alphabet.
- *Multiple Encodings*: Generates multiple encoding variants for a single sequence, allowing users to choose the best representation based on TF-IDF scores.
- *Optimized Encoding*: Supports batch substitution of token pairs and IDF-based token weighting for shorter, more efficient encodings.
- *High-Performance Backends*: Includes Python-native and Cython (C++20) backends for fast execution and low overhead.
- *PyPI Integration*: Easy to install and integrate into Python projects ('pip install ubpe[native]' or 'pip install ubpe[cython]').

### Wavelet Neural Networks

[github repo](#), [pypi](#)

Feb. 2025 - Present

STACK: PYTHON (PYTORCH)

A PyTorch library for wavelet transforms in neural networks.

Key Features:

- *Wavelet Integration*: Supports 1D and 2D wavelet transforms (e.g., Haar, Daubechies) directly within PyTorch tensors.
- *Seamless PyTorch Integration*: Compatible with PyTorch's autograd system, enabling end-to-end training of wavelet-augmented models.
- *Applications*: Ideal for signal processing, image analysis, and time-series data augmentation.
- *Efficiency*: Optimized for GPU acceleration, leveraging PyTorch's tensor operations.
- *Modular Design*: Easy to integrate into existing neural network architectures.

### Corgi.jl

[github repo](#)

Jun. 2022 - Present

STACK: JULIA

Machine Learning Library from scratch.

### Undergraduate practice in NLP

[github repo](#), [colab notebook](#)

Feb. 2023 - Mar. 2023

STACK: PYTHON

(Mostly) CNN News analysis, including Named Entity Recognition using flair, Sentiment Analysis using TimeLMs (RoBERTa, tuned on tweets), and Extractive (textrank algorithm) and Abstractive (PEGASUS and T5) Summarization. The results could be seen in the provided colab demo.

## Fossil

STACK: CRYSTAL

A FastAPI-inspired framework for building REST APIs in Crystal.

Key Features:

- *Macro-based routing*: Automatically generates endpoints from annotated methods, reducing boilerplate.
- *Parameter support*: Handles path, query, form, file, and body parameters with type safety.
- *Dependency injection*: Supports header, cookie, and singleton dependencies for modular and reusable code.
- *Content flexibility*: Built-in JSON serialization and custom content-type handling.
- *Lightweight wrapper*: Extends Crystal's HTTP::Server with minimal overhead.

[github repo, docs, guide](#)

Jul. 2025 - Present

## Hamilton

STACK: CRYSTAL, TELEGRAM BOT API

A lightweight framework for building Telegram bots in Crystal.

Key Features:

- Handles commands, text messages, and rich media (stickers, videos, GIFs, etc.).
- Supports context storage for stateful conversations.
- Enables asynchronous processing and signal-based task control for performance.
- Minimal boilerplate, extensible with custom handlers.

[github repo, docs, guide](#)

Sep. 2025 - Present

## tgsh: Telegram Shell Bot

STACK: CRYSTAL, HAMILTON.CR

A Telegram bot template for remote shell access to your server, built with Crystal and Hamilton.cr.

Key Features:

- Execute short and long-running commands with real-time output.
- Manage processes using POSIX signals.
- Upload/download files and run Python scripts directly from Telegram.
- Set custom log files for Python scripts and persistent logging.
- Async mode for better performance.

Use Case: Ideal for remote server administration, debugging, or running scripts on headless servers. Note: Avoid using on cloud VMs (violates most ToS).

[github repo](#)

Nov. 2025 - Present