

# NLP Assignment 1 – Text Processing Project Report

**Student:** Murad Valiyev

**Date:** 5 February 2026

## 1. Motivation

The goal of this project was to create a small Azerbaijani-language corpus from Wikipedia and apply standard text processing to it. Azerbaijani (Azeri) is a Turkic language with relatively limited publicly available NLP datasets and tools compared to high-resource languages like English. By building and processing our own corpus, we aimed to:

- Understand how tokenization, frequency statistics, subword methods, sentence segmentation, and spell-checking behave on real Azerbaijani encyclopedic text.
- Explore basic quantitative properties (e.g. Heaps' law) and practical challenges (e.g. diacritic-sensitive errors, common typos).

The corpus was collected from the Azerbaijani Wikipedia because it is freely accessible, high-quality, edited text in the target language. The setting is formal, edited, encyclopedic writing (monologue style, public knowledge-sharing). The project was individual, done for the course, with no external funding.

## 2. Datasheet / Data Statement

### Motivation

To create an Azerbaijani corpus for experimenting with tokenization, frequency analysis, Heaps' law, BPE, sentence splitting, and spell-checking algorithms as part of an NLP course assignment.

### Situation

Edited, formal encyclopedic articles written for public dissemination. Originally produced in Azerbaijani Wikipedia (2011–2025, most recent edits 2025).

### Language variety

Standard literary Azerbaijani (Latin script, Azerbaijan variety). Minimal dialectal variation.

### Speaker demographics

Authors are Wikipedia volunteers (mostly native Azerbaijani speakers from Azerbaijan and diaspora) and some bots like İşçiBot, Turkmenbot who also might be real people with a Botusername. No age/gender statistics available.

### Collection process

- Randomly sampled 3,000 articles via Wikipedia API

<https://az.wikipedia.org/w/api.php>

(action=query, list=random, rnnamespace=0).

- Extracted plain text + metadata (title, last edit timestamp, last editor).
- Final corpus (after cleaning): ≈5,423,130 characters, ≈617,554 tokens, ≈93,122 types.
- Pre-processing: removed section headings, wiki markup, templates, HTML tags, normalized multiple spaces/newlines.
- Data is public (CC BY-SA license), no consent required.
- Metadata preserved: title, last edit time, editor username.

## Annotation process

No manual annotations. All processing (tokenization, frequency counts, etc.) is done with mainly regex expressions.

## Distribution

Original Wikipedia content is CC BY-SA. The derived cleaned corpus is for educational/research use only.

## 3. Method

### Data collection

Used Python's requests library to fetch random articles from az.wikipedia.org API. Saved title, plain text, last edit timestamp and editor username into a CSV file.

### Tokenization (Task 1)

Custom rule-based regex tokenizer for Azerbaijani:

- Numbers with decimal/comma + suffix/currency/percentage (154.5\$, 50,5%, 2023-cü)
- Time formats (20:00)
- Words and hyphenated compounds (ayrı-ayrı, sərhəd-təhlükəsizlik)
- Case folding with Azerbaijani rules (I → İ, İ → i, then .lower()) Punctuation kept separate when not attached.

### Heaps' law (Task 2)

Computed vocabulary growth incrementally over tokens. Fitted  $V = k \times N^{\beta}$  using nonlinear least squares (scipy.optimize.curve\_fit).

### Byte-Pair Encoding (Task 3)

Simple character-level BPE implementation: started from characters + </w> end-of-word marker, merged top-frequency pairs iteratively (simple version, 25 merges shown).

### Sentence segmentation (Task 4)

Rule-based algorithm:

- Protected common abbreviations (Dr., Prof., Cən., və s., etc.) with placeholders
- Split on . ! ? followed by whitespace
- Recombined using heuristic (if next segment starts with lowercase → likely continuation)

### Spell checking (Task 5 + Extra)

- Baseline: uniform-cost Levenshtein distance (ins/del/sub cost = 1)
- Weighted version: custom Levenshtein with confusion matrix (lower costs for Azerbaijani-specific errors: a/ə=0.4, i/ɪ=0.4, o/ö=0.5, u/ü=0.5, ç/c=0.6, etc.) Tested on typical typos: azarbayan, qarabag, mesele, sehive, etc.

## 4. Experiments & Results

### Task 1 – Tokenization & frequencies

Corpus size after cleaning: 5,423,130 characters, 617,554 tokens, 93,122 types.

Type-Token Ratio ≈ 0.151 (low, expected for large corpus with repetition).

Top 25 most frequent tokens (clean, meaningful):

və (19,651), ilə (5,408), bu (4,683), bir (4,447), olan (2,950), üçün (2,728), də (2,519), azərbaycan (2,479), sonra (2,468), ildə (2,410), kimi (2,300), isə (2,070), o (2,019), edir (1,803), tərəfindən (1,771), da (1,765), onun (1,738), görə (1,714), çox (1,513), idi (1,509), ki (1,507), olaraq (1,408), böyük (1,393), ilk (1,291), the (1,290)

(Note: "the" appears due to English special terms in mainly citations of Wikipedia articles.)

### Task 2 – Heaps' law

Fitted parameters:

$k \approx 10.40$

$\beta \approx 0.683$

$\beta = 0.683$  is within the typical range for natural languages (0.4–0.8). It indicates reasonable vocabulary growth for a medium-sized, topically diverse corpus.

### Task 3 – BPE

First 25 merges produced frequent pairs and syllables: n</w>, ə</w>, i</w>, r</w>, ər, və</w>, də</w>, etc.

The implementation behaved correctly for the limited merges and corpus size.

### **Task 4 – Sentence segmentation**

The abbreviation-protected splitter handled Wikipedia text reasonably well.

Correctly avoided splitting on Dr., Prof., Cən., və s., etc.

Some limitations exist: There may be some over-splitting on numbers, decimals, and ellipses, though these are acceptable for a rule-based approach, there may be rare edge cases may not have been fully accounted for.

### **Task 5 & Extra – Spell checking**

Test set: common typos (azarbaycan, qarabag, mesele, sehife, etc.).

- Baseline Levenshtein: good candidates, but uniform cost sometimes ranked non-phonetic matches higher. Examples: azarbayan → azərbayan ( $d=1$ ), qarabag → qarabağ ( $d=1$ ), sehife → səhifə ( $d=1$ ) (after fixes).
- Weighted model: clearly better for Azerbaijani-specific errors due to confusion matrix. Examples: azarbayan → azərbayan ( $w\text{-dist}=0.40$ ) ranked highest, qarabag → qarabağ ( $w\text{-dist}=0.70$ ), səhifə → səhifə ( $w\text{-dist}=0.80$ ).

### **Error analysis / baseline comparison**

- Weighted version outperformed uniform Levenshtein on diacritic/phonetic errors (ə/a, ı/i, ö/o, ğ/g).
- Limitation: no language-model prior or context → ranking purely by distance + frequency.
- Negative result: BPE with only 25 merges did not produce linguistically meaningful subwords (too few iterations for real utility).

## **5. Contributions**

Individual project.

All tasks (data collection, cleaning, algorithms, implementation, report) completed by Murad.

## **References**

- Gebru et al. (2020). Datasheets for Datasets.
- Bender & Friedman (2018). Data Statements for Natural Language Processing.
- Wikipedia API ([az.wikipedia.org/w/api.php](https://az.wikipedia.org/w/api.php))
- Levenshtein distance (1965), BPE (Sennrich et al., 2016)