Spring 2026 Natural Language Processing (CSCI-6515)

Azerbaijani Wikipedia Corpus: Tokenization, Heaps' Law, BPE, Sentence Segmentation, and Spell Checking

Kamal Ahmadov
kahmadov24700@ada.edu.az; kamal.ahmadov@gwu.edu
Rufat Guliyev
rguliyev24988@ada.edu.az; rufat.guliyev@gwu.edu

ADA University & George Washington University

February 5, 2026

Motivation

- Azerbaijani is a low-resource language for NLP research
- Wikipedia provides freely licensed, diverse text data (CC BY-SA)
- Goal: build a clean corpus and implement core preprocessing modules
- Outputs are fully reproducible via `bash scripts/run_all.sh`

Dataset Snapshot

- Source: Azerbaijani Wikipedia (MediaWiki API)
- Cleaned corpus stored as CSV: `data/raw/corpus.csv`
- Final size:
    - 31,842 documents
    - 11,905,937 tokens
    - 586,674 unique token types

Cleaning Challenges

- Wikipedia markup noise:
    - templates, categories, navigation text, references
- Mixed-language artifacts (English-heavy references)
- Broken punctuation / formatting residue
- Short or empty pages removed to improve quality

Tokenization (Unicode-aware)

We tokenize using Unicode letter categories and preserve Azerbaijani characters.

$$\text{token} \in \{\text{letters}\}^{+} \quad \text{with optional internal } [\,'-]$$

Example:

‘‘S.Rustamov 2.5 df artırsaq’’ ⇒ [S.Rustamov, 2.5, df, artırsaq]

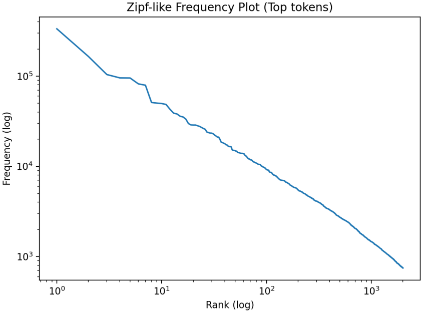
- Handles diacritics: , ğ, ö, ş, ü, ı, ç
- Keeps decimals to avoid splitting numeric expressions
- Lowercasing used for vocabulary normalization

Zipf's Law (Rank–Frequency)

- Empirical law for word frequencies
- Frequency decays with rank

$$f(r) \propto \frac{1}{r^s} \quad \Rightarrow \quad \log f(r) = c - s \log r$$

- Near-linear mid section on log–log plot
- Confirms heavy-tailed behavior in our corpus



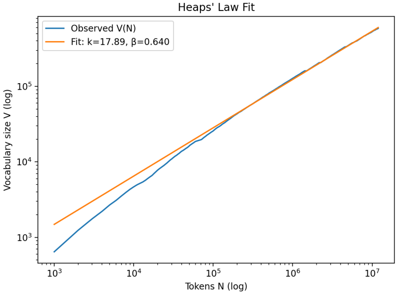
Heaps' Law (Vocabulary Growth)

$$V(N) = kN^{\beta} \quad \Rightarrow \quad \log V = \log k + \beta \log N$$

- Sample (N, V) every 1000 tokens
- Fit $\log V$ vs. $\log N$ by linear regression

$$k = 17.89, \qquad \beta = 0.640$$

- $\beta > 0.5$ indicates fast vocabulary growth
- Encyclopedic topics + named entities increase diversity



Byte Pair Encoding (BPE)

BPE learns subword units by repeatedly merging the most frequent symbol pairs.

$$(a, b) = \arg\max_{(x,y)} \mathrm{freq}(x\,y) \quad\Rightarrow\quad xy \to \langle xy \rangle$$

- Merges: 5000 → Subword vocab: 6946
- Reduces OOV rate and helps with morphology

Example:

$$\texttt{azrbaycanlılardan} \Rightarrow \texttt{azrbaycan} + \texttt{lı} + \texttt{lardan}$$

Enhanced Rule-Based Sentence Segmentation

Sentence boundaries are detected using punctuation

$$p \in \{., !, ?\}$$

only when contextual constraints are satisfied.

We block sentence boundaries if:

Non-space on both sides of $p \Rightarrow$ no split

Examples: `3.14`, `50.5`, `S.Rustamov`

Uppercase initial + period $\Rightarrow$ no split

Example: `A. Mlikli`

Segmentation Challenges and Heuristics

- Abbreviations: block splits after dr., prof., mr., etc.
- Decimals / numbers: no split inside 3.14, 50,5
- Initials / acronyms: no split inside S.Rustamov, A.M.
- Quotes: closing quote + space + lowercase ⇒ no split; + uppercase ⇒ split
- Lowercase continuation after period: keep kv. verst unsplit

Lightweight rules reduce false boundaries in encyclopedic text while remaining fast and interpretable.

Quotation-Aware and Context-Sensitive Splitting

Special handling is applied to quoted text and citations.

Quotation rule:

$$\text{closing quote} + \text{space} + \text{Uppercase} \Rightarrow \text{sentence boundary}$$

$$\text{closing quote} + \text{space} + \text{lowercase} \Rightarrow \text{no split}$$

This prevents over-splitting in:

- Embedded citations
- Lowercase continuations (e.g., `kv.   verst`)
- Azerbaijani quotation styles (", , etc.)

Segmentation Evaluation

- Metrics: Precision, Recall, F1 (sklearn) and BDER $= (FP + FN)/|gold|$
- Inputs: gold + predicted boundary indices (JSON or newline list)
- Tooling: `python -m src.evaluate_segmentation ...`; wrapper `bash scripts/eval_sentseg.sh <gold_sentences> <limit>`
- Status: pipeline ready; awaiting manually curated gold set (auto gold used only for scaffolding)

Why These Rules Matter

- Azerbaijani Wikipedia contains dense punctuation and abbreviations
- Naive punctuation splitting produces many false sentence boundaries
- Our rules significantly reduce:
  - False splits in names and numbers
  - Errors in quoted material
- Lightweight, interpretable, and fast
- Strong baseline for low-resource languages

Spell Checking: Levenshtein Distance

Baseline: Levenshtein distance via dynamic programming.

$$D(i,j) = \min \begin{cases} D(i-1,j)+1 & \text{delete} \\ D(i,j-1)+1 & \text{insert} \\ D(i-1,j-1)+[x_i \neq y_j] & \text{substitute} \end{cases}$$

- Given misspelled token w, search candidates $c \in V$
- Rank by smallest $D(w,c)$ (with pruning / max distance)

Spell Checking: Candidate Generation

Before querying the model, we generate candidates by focusing on common letter substitutions in Azerbaijani:

- Problematic letters: { a, o, u, c, s, ş, ç, e, ğ }
- Substituting these letters gives multiple possible candidates
- For each candidate, we retrieve 5 suggestions from the model

Spell Checking: Ranking and Pruning Candidates

After generating multiple candidates:

- Concatenate all candidate lists
- Rank candidates by Levenshtein distance and frequency in the dataset
- Prune irrelevant candidates based on character length

Final suggestion:

- Top-5 words with shortest distance and highest frequency

Spell Checking: Impact on Accuracy

- Precision and recall greatly improved due to:
  - Candidate generation via substitution
  - Higher quality model suggestions
- Top-5 suggestions highly reliable, reducing false positives

Spell Checker: Final Recap

Candidate Generation:

- Letter substitution for common spelling errors
- Generate candidates by substituting frequent miswritten characters

Ranking Final Suggestions:

- Top-5 ranked suggestions from model, based on Levenshtein distance and frequency

Weighted Edit Distance (From Confusions)

We replace uniform substitution costs with learned costs from character confusions.

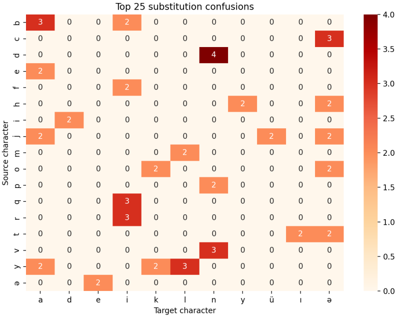
$$\text{cost}(a \to b) = \frac{1}{\epsilon + P(b|a)}$$

Weighted recurrence:

$$D(i,j) = \min \begin{cases} D(i-1,j) + w_{\text{del}}(x_i) \\ D(i,j-1) + w_{\text{ins}}(y_j) \\ D(i-1,j-1) + w_{\text{sub}}(x_i, y_j) \end{cases}$$

Weighted Edit Distance (Interpretation)

- Confusions learned from synthetic misspellings
- Common substitutions get lower penalty
- Diacritic and keyboard-adjacent errors favored
- Improves ranking of realistic corrections
- Candidate pruning: length filter ($||w| - |c|| \leq$ max_dist) before scoring; ranking by (distance, freq)


Top 25 substitution confusions

Evaluation Metrics

We evaluate spellchecker ranking using Accuracy@k.

$$\text{Acc@k} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}\{y_i \in \text{TopK}(x_i)\}$$

Results on 1000 synthetic misspellings:

$$\text{Acc@1} = 0.489, \qquad \text{Acc@5} = 0.726$$

- Acc@1 = strict correction success
- Acc@5 = practical usefulness of suggestions

Spellcheck Pipeline Recap

- Vocab filtering: Azerbaijani alphabet only, min_freq (2), min_len (3)
- Candidate pruning: length difference $\leq$ max_dist (2) before distance
- Distance: Levenshtein; weighted edits if confusion weights provided
- Ranking: (distance asc, frequency desc), top-5 suggestions
- Rare-token scan: report suggestions for 200 rarest tokens to spot likely typos

Key Outputs (Reproducibility)

- Run everything:
  - `bash scripts/run_all.sh`
- Plots:
  - `outputs/plots/zipf.png`, `outputs/plots/heaps.png`
- BPE:
  - `outputs/bpe/merges.txt`, `bpe_summary.json`
- Sentence segmentation:
  - `outputs/sentences.txt`, optional eval `outputs/sentseg_eval.json`
- Spellcheck:
  - `spell_eval.json`, `confusion_heatmap.png`

Conclusion

- Built a clean Azerbaijani Wikipedia corpus (31,842 docs)
- Verified Zipf-like distribution and fitted Heaps' law
- Trained BPE subword model for robust tokenization
- Implemented rule-based sentence segmentation with edge-case handling
- Developed spellchecker with weighted edit distance; Acc@5 $= 0.726$

Future Work

- Improve cleaning and Azerbaijani-only filtering earlier
- Replace synthetic gold data with manual annotations
- Context-aware spell correction using language model scoring
- Explore neural sentence segmentation as stronger baseline

Questions?