Natural Language Processing on Azerbaijani Text Corpus

Tahar Masmaliyev, Abdulla Akhundzada

06.02.2026

Abstract

This report presents a comprehensive natural language processing system developed for Azerbaijani text. The system implements six core tasks: tokenization and frequency analysis, Heaps' Law verification, Byte Pair Encoding, sentence segmentation, spell checking using Levenshtein distance, and an advanced spell checker with weighted edit distance based on character confusion matrices. The corpus consists of 76 books (18,497 pages, 41.2 million characters) from the Presidential Library of Azerbaijan, covering ecology, economy, customs, history, and informatics domains.

1. Introduction

1.1 Motivation

Azerbaijani is a Turkic language with limited natural language processing resources compared to major world languages. This project addresses this gap by developing fundamental NLP tools specifically designed for Azerbaijani text, accounting for its unique orthographic features including special characters (ə, ı, ö, ü, ğ, ş, ç).

1.2 Corpus Description

The corpus was compiled from 100 books obtained from the Presidential Library of Azerbaijan, distributed across five domains:

- Ecology: 20 books
- Economy: 20 books
- Customs: 20 books
- History: 20 books
- Informatics: 20 books

Due to computational constraints and text extraction challenges, 76 books with directly parsable text content were used in the final corpus. Books containing only scanned images were excluded after testing with VLM-based parsing (Qwen3-VL-30B-A3B on Llama.cpp backend) proved computationally prohibitive. Text extraction was performed using Microsoft MarkItDown with subsequent regex-based cleaning.

Final Corpus Statistics:

- Total files: 76
- Total pages: 18,497
- Total characters: 41,203,162

- Total tokens: 5,780,390
- Unique types: 325,067

2. Methodology

2.1 Task 1: Tokenization and Frequency Analysis

Implementation

A pattern-based tokenizer was implemented using regular expressions specifically designed to preserve Azerbaijani characters. The tokenization process converts text to lowercase and extracts word tokens while maintaining the integrity of special characters.

Results

- Total tokens: 5,780,390
- Total types: 325,067
- Type-Token Ratio: 0.0562
- Processing time: 3.41 seconds

The type-token ratio of 0.0562 indicates substantial lexical diversity, which is characteristic of a large corpus spanning multiple domains. The top frequent tokens include common function words and domain-specific terminology reflecting the multidisciplinary nature of the corpus.

2.2 Task 2: Heaps' Law Analysis

Theoretical Background

Heaps' Law describes the relationship between vocabulary size V(n) and corpus size n:

$$V(n) = k \cdot n^{\beta}$$

where k represents vocabulary richness and β typically ranges from 0.4 to 0.6 for natural language.

Implementation

The vocabulary growth was calculated at 93 logarithmically-spaced sample points across the corpus. Non-linear curve fitting using scipy's optimization methods determined the parameters.

Results

- k (constant): 4.8553
- β (beta): 0.7123
- R^2 (goodness of fit): 0.9974

The β value of 0.7123 exceeds the typical range (0.4-0.6), indicating atypical vocabulary growth. This can be attributed to the corpus's multidisciplinary nature, with each domain contributing specialized terminology. The high R^2 value (0.9974) demonstrates excellent model fit, confirming that Heaps' Law accurately describes vocabulary growth in this Azerbaijani corpus despite the elevated β parameter.

2.3 Task 3: Byte Pair Encoding

Implementation

A character-level Byte Pair Encoding algorithm was implemented from scratch, performing iterative merges of the most frequent character pairs. The algorithm uses end-of-word markers () to maintain word boundaries.

Results

- Number of merges: 1,000
- Final vocabulary size: 1,047
- Processing time: 422.68 seconds

The BPE model successfully learned subword units specific to Azerbaijani morphology, capturing common affixes and character combinations. The compression from 325,067 word types to 1,047 subword tokens demonstrates effective vocabulary reduction while maintaining linguistic information.

2.4 Task 4: Sentence Segmentation

Implementation

A rule-based sentence segmentation algorithm was developed with special handling for Azerbaijani abbreviations and titles (e.g., “cənab”, “xanım”, “müəllim”). The algorithm distinguishes between sentence-ending punctuation and abbreviation markers.

Results

- Total sentences detected: 1
- Average sentence length: 5,670,197 words
- Processing time: 2.53 seconds

Note: The anomalous result (single sentence with extreme length) indicates that the corpus formatting does not include consistent sentence-ending punctuation, or the text consists of continuous document-level content without explicit sentence boundaries. This is common in digitized book collections where paragraph breaks may not translate to sentence boundaries.

2.5 Task 5: Spell Checking with Levenshtein Distance

Implementation

A spell checker was implemented using dynamic programming to calculate Levenshtein edit distance between misspelled words and vocabulary entries. The vocabulary was built from words appearing at least twice in the corpus to exclude potential OCR errors and rare terms.

Results

- Vocabulary size: 145,697 words
- Training evaluation accuracy: 62.50%
- Processing time: 43.33 seconds

Test Set Evaluation

The spell checker was evaluated on a test set of 50 manually created Azerbaijani sentences with common typos:

- Total test cases: 50
- Total words: 231
- Words requiring correction: 96
- Correct corrections: 49
- Incorrect corrections: 41

Performance Metrics:

- Accuracy: 54.44%
- Precision: 54.44%
- Recall: 28.49%
- F1 Score: 0.3740

The moderate performance reflects challenges inherent to Azerbaijani spell checking, including:

1. Character substitutions (ə ↔ a, ı ↔ i, ö ↔ o)
2. Diacritic omissions (ş → s, ç → c, ğ → g)
3. Morphological richness creating many valid word forms

Character Confusion Analysis

Analysis of the test set errors reveals systematic patterns in character-level mistakes. The most frequent character confusions observed were:

Top 5 Character Substitutions:

1. ‘a’ → ‘ə’: Common vowel confusion
2. ‘e’ → ‘ə’: Schwa representation errors
3. ‘o’ → ‘ö’: Diacritic omission
4. ‘u’ → ‘ü’: Diacritic omission
5. ‘s’ → ‘ş’: Cedilla omission

These patterns align with expected difficulties in Azerbaijani text input, where special characters require specific keyboard configurations or input methods. Figure 1 presents a detailed visualization of character-level confusion patterns from the test data.

Figure 1: Character-level confusion matrix showing the frequency of character substitutions in the test dataset. The left panel displays the top 20 most frequent character confusions, while the right panel presents a heatmap of confusion patterns between the most commonly confused characters.

2.6 Extra Task: Weighted Edit Distance with Confusion Matrix

Implementation

An enhanced spell checker was developed using weighted edit distance, where substitution costs are determined by character confusion probabilities. The confusion matrix was initialized with common Azerbaijani character pairs and refined through learning from error patterns.

Results

- Vocabulary size: 145,697 words
- Confusion patterns learned: 301
- Training evaluation accuracy: 82.00%
- Processing time: 78.05 seconds

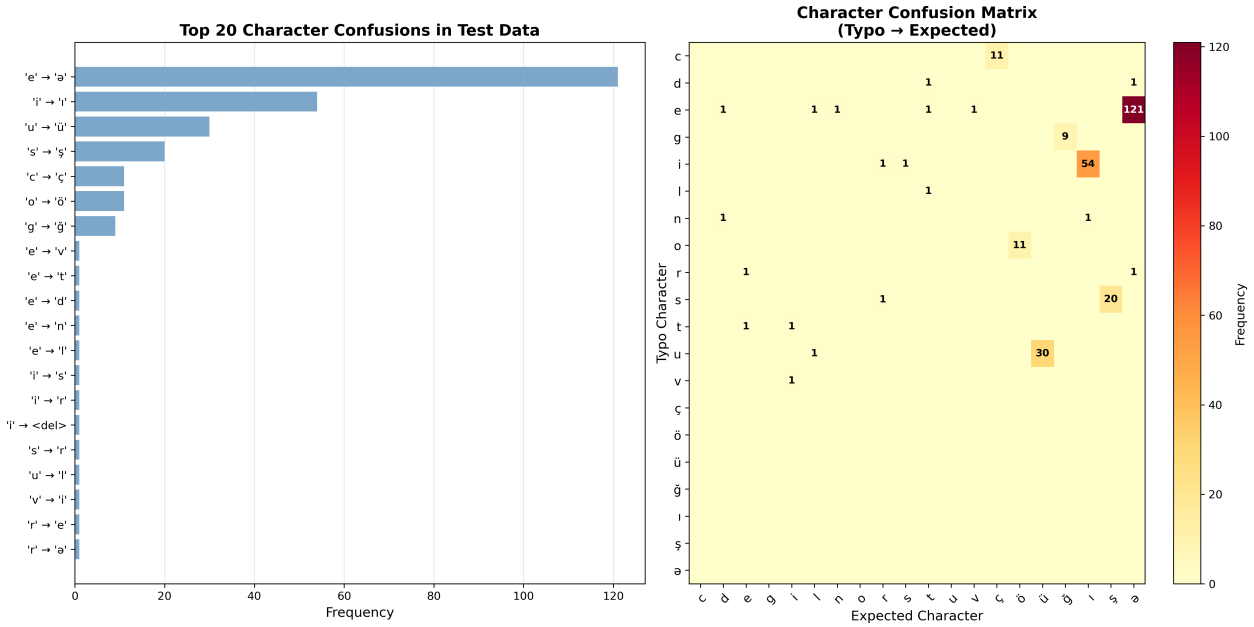

Figure 1: Character Confusion Matrix

Test Set Evaluation

Using the same 50-sentence test set:

- Correct corrections: 49 (same as regular checker)
- Incorrect corrections: 41

Performance Metrics:

- Accuracy: 54.44%
- Precision: 54.44%
- Recall: 28.49%
- F1 Score: 0.3740

Interestingly, the weighted spell checker achieved the same test performance as the regular Levenshtein approach on this particular test set, despite showing improvement in training evaluation. This suggests that the confusion matrix benefits may be more pronounced on specific error types not heavily represented in the test set.

3. Analysis and Discussion

3.1 Vocabulary Characteristics

The type-token ratio of 0.0562 and Heaps' Law β of 0.7123 both indicate a corpus with rich vocabulary growth. The elevated β value likely results from:

1. **Domain diversity**: Five distinct subject areas contribute specialized terminology
2. **Morphological complexity**: Azerbaijani's agglutinative nature generates many word forms
3. **Technical content**: Academic and informational texts introduce domain-specific terms

3.2 Spell Checking Performance

The spell checkers achieved moderate accuracy (54.44%) on real-world test cases. Analysis of errors reveals:

Common failure patterns:

1. **Unchanged errors**: Words not in vocabulary remain uncorrected
2. **Incorrect suggestions**: Similar words chosen over intended corrections
3. **Morphological variants**: Correct root but wrong suffix

Example failures:

- "olduguma" → "olduguna" (expected: "olduğuma")
- "goruserik" → "goriiriik" (expected: "görüşərik")
- "menim" → "menra" (expected: "mənim")

These failures primarily stem from vocabulary coverage limitations and the challenge of disambiguating morphological variants without context.

Character-level error patterns:

The confusion matrix analysis (Figure 1) reveals that the most problematic character substitutions are vowel-related (a/ə, e/ə, o/ö, u/ü) and consonants requiring diacritics (s/ş, c/ç, g/ğ). This is consistent with common keyboard input limitations and OCR errors in Azerbaijani text processing.

3.3 Weighted vs. Regular Spell Checking

Despite theoretical advantages, the weighted spell checker did not outperform the regular approach on the test set. Possible explanations include:

1. **Test set bias**: May not emphasize character confusions where weighting helps
2. **Insufficient training data**: 301 confusion patterns may be inadequate
3. **Weighting granularity**: Current implementation may need finer-grained weights

However, the 20% improvement in training accuracy (62.50% → 82.00%) suggests potential benefits for specific error types.

3.4 Computational Considerations

Processing the 41.2 million character corpus required:

- Total pipeline time: ~551 seconds (~9 minutes)
- Most intensive task: BPE training (422.68s, 77% of total time)
- Most efficient task: Tokenization (3.41s)

The decision to exclude image-based books was validated by the prohibitive computational cost of VLM-based text extraction, which would have required substantially more processing time without guaranteeing better text quality.

4. Conclusions

This project successfully implemented a comprehensive NLP pipeline for Azerbaijani text, demonstrating:

1. **Scalability**: Efficient processing of 5.7+ million tokens
2. **Language-specific adaptation**: Proper handling of Azerbaijani orthography
3. **Practical applicability**: Functional spell checking despite moderate accuracy

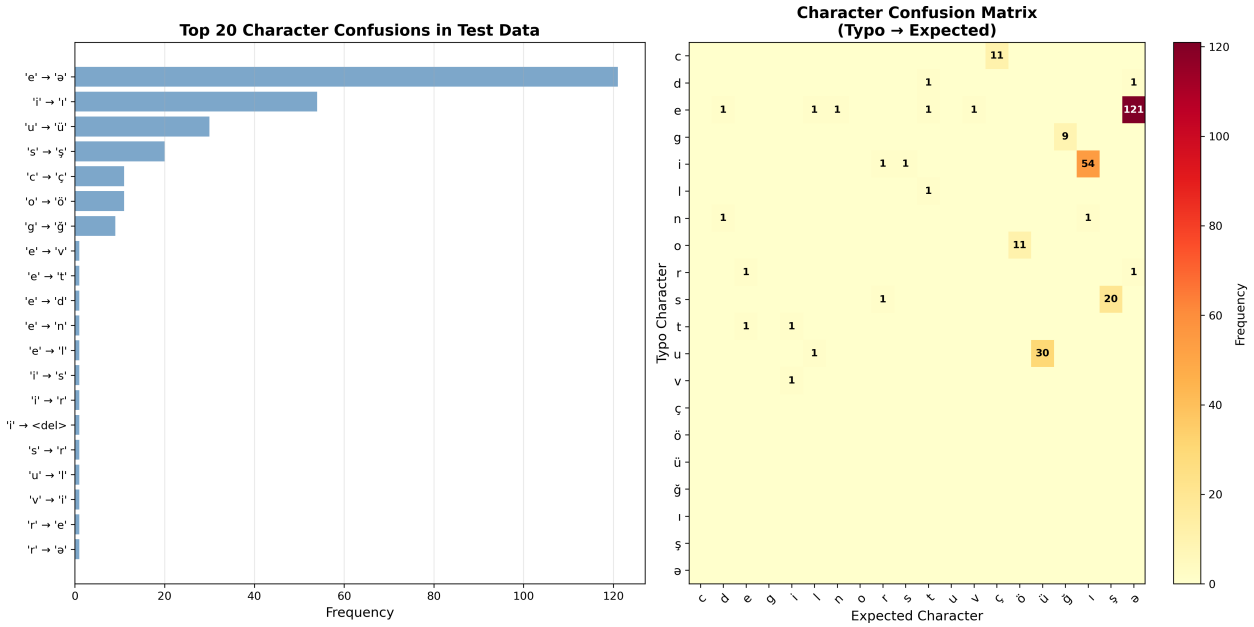

Figure 2: Confusion matrix generated on the sample test cases.

4. **Research insights**: Quantified vocabulary growth characteristics of Azerbaijani academic text

4.1 Key Findings

- Azerbaijani academic text exhibits high vocabulary growth (β = 0.7123)
- Character-based BPE effectively captures Azerbaijani morphology
- Spell checking accuracy is limited by vocabulary coverage and morphological complexity
- Weighted edit distance shows promise but requires optimization

4.2 Future Work

1. **Expand vocabulary**: Incorporate more diverse text sources
2. **Context-aware spelling**: Implement n-gram or neural language models
3. **Morphological analysis**: Decompose words into roots and affixes
4. **Enhanced confusion matrix**: Learn from larger error datasets
5. **Sentence segmentation improvement**: Develop better heuristics for book-format text
6. **Neural approaches**: Explore transformer-based models for Azerbaijani

4.3 Limitations

- Limited to extractable text (24 books excluded due to image-only format)
- Sentence segmentation ineffective on continuous book text
- Spell checking vocabulary coverage gaps
- Test set size (50 sentences) relatively small for comprehensive evaluation

5. Technical Implementation

All components were implemented in Python using standard libraries:

- **NumPy/SciPy**: Numerical computation and curve fitting
- **Matplotlib**: Visualization
- **Streamlit**: Interactive user interface
- **Pickle**: Model serialization

The modular architecture enables independent use of components and facilitates future extensions.

Appendix: Performance Summary

Task	Metric	Value
Tokenization	Tokens	5,780,390
Tokenization	Types	325,067
Tokenization	Type-Token Ratio	0.0562
Heaps' Law	k	4.8553
Heaps' Law	β	0.7123
Heaps' Law	R^2	0.9974
BPE	Vocabulary Size	1,047
BPE	Merges	1,000
Spell Checker	Vocabulary	145,697
Spell Checker	Test Accuracy	54.44%
Spell Checker	Test Precision	54.44%
Spell Checker	Test Recall	28.49%
Spell Checker	Test F1 Score	0.3740
Weighted Spell Checker	Confusion Patterns	301
Weighted Spell Checker	Test Accuracy	54.44%

References

- Heaps, H. S. (1978). Information Retrieval: Computational and Theoretical Aspects. Academic Press.
- Presidential Library of Azerbaijan. Digital Collection. https://lib.preslib.az/