# Extending zxcvbn Password Strength Estimator with Slovenian Language Support

**Filip Merkan**

*Faculty of Computer and Information Science*
*Večna pot 113*
*1000 Ljubljana*

**Abstract.** Password strength estimation is crucial for cybersecurity. The zxcvbn library, developed by Dropbox, provides accurate password strength estimation by analyzing patterns, dictionary matches, and keyboard layouts. However, it primarily focuses on English language patterns, limiting its effectiveness for non-English passwords. This project extends zxcvbn with comprehensive Slovenian language support, including Slovenian word frequency lists, names, surnames, common passwords, and QWERTZ keyboard layout recognition. Data was collected from official Slovenian sources (SURS), text collections (CLARIN), and real password breach databases. The implementation adds Slovenian dictionaries, keyboard pattern recognition, and localized feedback messages. A demo application demonstrates the enhanced accuracy for Slovenian passwords, showing significant improvements in detecting weak passwords containing Slovenian words, names, and patterns that were previously undetected.

**Key words:** password strength, zxcvbn, Slovenian language, cybersecurity, dictionary attack

## 1 INTRODUCTION

Password security remains a critical concern in modern cybersecurity. Weak passwords are a primary vector for unauthorized access, with dictionary attacks and pattern-based guessing being among the most common attack methods. The zxcvbn library [3] provides a sophisticated approach to password strength estimation by analyzing multiple factors: dictionary matches, keyboard patterns, sequences, and common substitutions.

However, zxcvbn's original implementation primarily targets English-language passwords. It includes English word frequency lists, common English names and surnames, and QWERTY keyboard layout patterns. This limitation significantly reduces its effectiveness for users creating passwords in other languages, particularly those using non-Latin scripts or different keyboard layouts.

Slovenian presents unique challenges for password strength estimation. The language uses a QWERTZ keyboard layout (distinct from QWERTY), includes special characters (č, š, ž), and has its own corpus of common words, names, and cultural references that may appear in passwords. Without language-specific support, passwords like "geslo123" (Slovenian for "password123"), "ljubljana2024" (Slovenia's capital city), or "filipmerkan" (the full legal name of the author) would be incorrectly assessed as stronger than they actually are.

This project extends zxcvbn with comprehensive Slovenian language support, enabling accurate strength estimation for passwords containing Slovenian linguistic elements. The contributions include:

- Slovenian word frequency lists derived from text collections
- Slovenian names and surnames from official statistical sources
- Common Slovenian password patterns from real breach data
- QWERTZ keyboard layout pattern recognition
- Localized feedback messages in Slovenian
- Date format recognition for Slovenian conventions

## 2 RELATED WORK

Password strength estimation has been extensively studied. Wheeler [3] introduced zxcvbn as a practical alternative to traditional password policies, using pattern matching and entropy estimation rather than arbitrary complexity requirements. The library's approach has been adopted widely due to its accuracy and user-friendly feedback.

Language-specific password analysis has received less attention. Most password strength estimators assume English-language input, despite evidence that users worldwide create passwords in their native languages [1]. Research on non-English password patterns is limited, with few tools providing comprehensive multilingual support.

Keyboard layout analysis has been explored in the context of password security [2]. Different layouts (QWERTY, QWERTZ, AZERTY) produce distinct patterns that attackers can exploit. zxcvbn includes support for QWERTY and AZERTY layouts, but lacked QWERTZ support until this work.

## 3 METHODOLOGY

### *3.1 Data Collection*

Slovenian language data was collected from multiple authoritative sources to ensure comprehensive coverage and accuracy.

*3.1.1 Word Frequency Lists:* Slovenian word frequencies were extracted from multiple text collections:

- **CLARIN Repository**: The Trendi Corpus 2021 and Slovenian Wikipedia corpus provided word frequency statistics from contemporary Slovenian texts, including news articles and encyclopedic content.
- **SLED Corpus**: Historical word frequency data comparing 2021 usage against 1991-2020 trends, ensuring coverage of both contemporary and traditional vocabulary.

Words were filtered to remove very short tokens (less than 4 characters) and normalized to lowercase. The final list contains approximately 30,000 most frequent Slovenian words, ranked by frequency.

*3.1.2 Names and Surnames:* Official data from the Statistical Office of the Republic of Slovenia (SURS) provided comprehensive name and surname statistics:

- **Male names**: Complete list of 4,902 names with frequency rankings (e.g., Franc, Janez, Marko, Andrej)
- **Female names**: Complete list of 4,861 names with frequency rankings (e.g., Marija, Ana, Petra, Mojca)
- **Surnames**: Complete list of 33,493 surnames, alphabetically organized and divided into four files, with Novak being the most common

The data was processed from CSV format, handling Windows-1250 encoding and extracting name/surname frequencies from the appropriate columns. All names were normalized to lowercase for consistent matching.

*3.1.3 Password Lists:* Real password data was collected from multiple sources to capture authentic Slovenian password patterns. Finding Slovenian-specific password lists proved challenging, as most publicly available password databases focus on English-language patterns. Additionally, legal and ethical considerations limit access to password data, as modern security best practices advocate for storing only password hashes rather than plaintext passwords, making it difficult to obtain authentic Slovenian password patterns from legitimate sources.

The following sources were utilized:

- **Have I Been Pwned**: Processed the ordered-by-count password list, filtering for Slovenian-relevant patterns and removing hash-like entries. This database contains millions of real passwords from data breaches, providing authentic patterns used by actual users.
- **weakpass.com**: Downloaded and processed Slovenian-specific password lists including "mega_slovar.txt" (320MB) and "Slovene.udic" (13MB). These lists contain hundreds of thousands of passwords collected from various sources, with particular focus on Central European language patterns.

Password lists underwent rigorous filtering to ensure quality:

- Removed cryptographic hashes (MD5, SHA1, SHA256 patterns) that could skew frequency analysis
- Filtered out very short passwords (less than 4 characters) and excessively long ones (more than 50 characters)
- Prioritized text-based passwords over purely numeric sequences to focus on dictionary-attackable patterns
- Normalized all passwords to lowercase for consistent matching
- Deduplicated entries while preserving frequency information

The final list contains 30,000 most common passwords, sorted by frequency. This size was chosen to balance coverage with performance, ensuring comprehensive pattern detection without excessive memory overhead.

### *3.2 Data Processing*

All data sources required preprocessing to match zxcvbn's expected format. The library expects frequency lists as plain text files, one token per line, sorted by frequency (most common first). Python scripts were developed to:

1) Parse various input formats (CSV, TSV, plain text)
2) Handle encoding issues (UTF-8, Windows-1250, UTF-16)
3) Filter and normalize tokens
4) Sort by frequency
5) Generate output in zxcvbn-compatible format

The build process integrates these lists using `build_frequency_lists.py`, which:

- Parses all frequency list files
- Filters short, rare tokens (rank $> 10^{\text{length}}$)
- Deduplicates tokens across dictionaries (keeping the lowest rank)

- Applies dictionary-specific limits (30,000 words for Wikipedia, 30,000 passwords)
- Generates `frequency_lists.coffee` for compilation

The following Slovenian datasets were added to the existing zxcvbn dictionaries:

- 30,000 most frequent Slovenian words from text collections (added to existing English Wikipedia word list)
- 4,902 Slovenian male names from SURS (added to existing English male names)
- 4,861 Slovenian female names from SURS (added to existing English female names)
- 33,493 Slovenian surnames from SURS (added to existing English surnames)
- 30,000 most common Slovenian passwords from breach databases (added to existing English password list)

## 4 Implementation

### *4.1 Keyboard Layout Support*

Slovenian uses the QWERTZ keyboard layout, which differs from QWERTY in the placement of the Z and Y keys, and includes additional characters (č, š, ž) accessible via dead keys or AltGr combinations. This layout difference is significant for password security, as users often create passwords by following keyboard patterns, and these patterns differ between layouts.

zxcvbn's adjacency graph system was extended to recognize QWERTZ patterns. The original zxcvbn supported QWERTY (US layout) and AZERTY (French layout), but lacked QWERTZ support despite its widespread use in Central and Eastern Europe.

The implementation adds a new adjacency graph definition in `matching.coffee` that maps:

- **Horizontal adjacencies**: Characters on the same row, such as q-w-e-r-t-z (the top row in QWERTZ)
- **Vertical adjacencies**: Characters in the same column across rows, considering the shifted positions in QWERTZ
- **Diagonal adjacencies**: Characters diagonally adjacent, accounting for the different key positions
- **Special character positions**: Mapping for č, š, ž which are commonly used in Slovenian and accessible via AltGr or dead keys

The adjacency graph is represented as a directed graph where each node (key) has edges to its adjacent keys. This allows zxcvbn to detect sequences of adjacent keys, which are weak because they're easy to type and remember, but also easy for attackers to guess.

This enables detection of weak passwords like "qwertz", "asdfgh", or "yxcvbn" that follow Slovenian keyboard patterns. Without QWERTZ support, these patterns would be missed or incorrectly assessed, as the original QWERTY adjacency graph doesn't account for the Z-Y swap and different key positions.

### *4.2 Feedback Localization*

User feedback is crucial for password strength estimators. A multilingual feedback system supporting both English and Slovenian was implemented. The system:

- Detects language preference via function parameter or global setting
- Provides localized warnings and suggestions
- Recognizes Slovenian dictionary matches in feedback
- Includes Slovenian-specific date format recognition

Example feedback translations include:

- "Straight rows of keys are easy to guess" → "Ravne vrstice tipk je enostavno uganiti"
- "This is a top-10 common password" → "To je eno izmed 10 najpogostejših gesel"
- "Slovenian keyboard patterns are easy to guess" → "Slovenski vzorci tipkovnice je enostavno uganiti"

### *4.3 Date Pattern Recognition*

Date pattern recognition was enhanced to support Slovenian date formats (DD.MM.YYYY) in addition to existing formats. The implementation recognizes DDMMYYYY, DD.MM.YYYY, DDMMYY, and YYYYMMDD patterns, enabling detection of weak passwords containing dates in Slovenian format. Without this support, such passwords would be analyzed as numeric sequences rather than dates, potentially underestimating their weakness.

### *4.4 Integration Process*

The integration required modifications to multiple components of the zxcvbn codebase:

1) **Build script** (`build_frequency_lists.py`): Added Slovenian dictionaries to the `DICTIONARIES` configuration with appropriate limits:
   - `slovenian_wikipedia`: Limited to 30,000 most frequent words
   - `slovenian_passwords`: Limited to 30,000 most common passwords
   - `slovenian_male_names`: No limit (all names included)
   - `slovenian_female_names`: No limit (all names included)
   - `slovenian_surnames`: No limit (all surnames included)
2) **Frequency lists**: Generated new `frequency_lists.coffee` containing all Slovenian data. This file is automatically

generated by the build script and includes all dictionaries in a format optimized for JavaScript execution. The script handles deduplication, ensuring each token appears only once in the most appropriate dictionary.

3) **Matching module** (`matching.coffee`): Added QWERTZ adjacency graph definition. The module now includes a complete mapping of QWERTZ keyboard layout, allowing detection of keyboard pattern matches specific to Slovenian keyboards.
4) **Feedback module** (`feedback.coffee`): Added comprehensive Slovenian translations for all feedback messages. The module now supports language detection via function parameter or global setting, and provides culturally appropriate translations that maintain the meaning and urgency of security warnings.
5) **Main module** (`main.coffee`): Added language parameter support to the main `zxcvbn()` function. The function signature now accepts an optional third parameter for language selection, allowing runtime language switching without requiring separate library instances.
6) **Build process**: Recompiled CoffeeScript to JavaScript using browserify, maintaining source maps for debugging. The build process bundles all modules into a single file while preserving the modular structure for maintainability.

The final build produces a single JavaScript file (`zxcvbn.js`) that includes all Slovenian language support without requiring separate language packs. This design choice ensures backward compatibility with existing code while adding new functionality. The library can be used exactly like the original zxcvbn, with the added capability of Slovenian language support when needed.

The integration maintains the original zxcvbn's performance characteristics while extending its capabilities. All Slovenian dictionaries are included in the build, but the matching algorithm efficiently searches only relevant dictionaries based on the password being analyzed, ensuring minimal performance overhead.

## 5 Results and Evaluation

### *5.1 Demo Application*

A comprehensive demo application was developed that allows side-by-side comparison of the original zxcvbn (English-only) and the Slovenian-enhanced version. The demo loads both versions simultaneously, provides real-time password strength analysis, displays dictionary matches to show which dictionaries detected patterns, and includes a language selector for feedback messages. This enables users to directly observe the improvements in password strength estimation for Slovenian passwords.

### *5.2 Test Results*

A comparison test was conducted with 73 representative Slovenian passwords covering various patterns: Slovenian words, names, surnames, common password patterns, dates, and leet speak variations. Both the original zxcvbn (English-only) and the Slovenian-enhanced version were tested on the same password set.

The results demonstrate significant improvement:

- **Original zxcvbn**: Correctly identified 47 out of 73 passwords as weak (64%)
- **Slovenian-enhanced**: Correctly identified all 73 passwords as weak (100%)
- **Improvement**: 26 passwords (36%) that were incorrectly assessed as strong by the original version are now correctly identified as weak

Key examples where the enhanced version improved detection:

- **Slovenian words**: ”slovenija” - Original score 3 (incorrectly strong, likely due to detecting the English word ”love” within it), Enhanced score 0 (correctly weak) via `slovenian_wikipedia`
- **Names**: ”filipmerkan” - Original score 4 (incorrectly strong), Enhanced score 2 (correctly weak) via `slovenian_surnames`
- **Common passwords**: ”slovenija2024”, ”geslo2024”, ”ljubljana2024” - Original scores 3-4, Enhanced score 0 via `slovenian_passwords`
- **Leet speak**: ”sl0v3n1j4” - Original score 3, Enhanced score 0 via dictionary matching

The enhanced version provides accurate strength estimates for Slovenian passwords, with dictionary matches correctly identifying patterns that would be missed by English-only analysis. Performance remains acceptable with analysis completing in under 10ms for typical passwords, despite the addition of approximately 2MB of Slovenian dictionary data.

## 6 Conclusion

This project successfully extends zxcvbn with comprehensive Slovenian language support, addressing a significant gap in password strength estimation for non-English languages. The implementation includes:

- Comprehensive Slovenian word, name, and surname dictionaries derived from authoritative sources
- Real password data from breach databases, filtered and validated for quality
- QWERTZ keyboard layout pattern recognition for Slovenian keyboards

- Localized feedback in Slovenian with culturally appropriate translations
- Enhanced date format recognition supporting Slovenian conventions

The enhanced library provides significantly more accurate password strength estimation for Slovenian users, correctly identifying weak passwords that would have been missed by the English-only version. The evaluation showed a dramatic improvement: from 64% detection rate with the original version to 100% with the Slovenian-enhanced version for a representative set of 73 Slovenian passwords.

The demo application demonstrates these improvements clearly, showing dictionary matches and providing localized feedback. Users can see side-by-side comparisons and understand why their passwords are weak, encouraging better password practices.

### *6.1 Known Limitations of zxcvbn*

This implementation extends zxcvbn with Slovenian language support, but inherits several fundamental limitations from the original zxcvbn library architecture. These limitations affect password strength estimation accuracy in certain scenarios:

- **Multi-word phrase matching**: zxcvbn does not perform multi-word phrase matching, but only searches for the presence of individual words in dictionaries. For example, the password ”sirni burek” would be analyzed by finding the word ”sir” in the dictionary, but the remaining 8 characters (” burek”) would not be recognized as a pattern, causing the password to be incorrectly assessed as secure.
- **Morphological analysis**: The library does not perform morphological analysis and does not understand that word forms like ”čokoladno” (chocolate, adjective form) are related to their base forms like ”čokolada” (chocolate, noun). Since ”čokoladno” is not found in the dictionary, the library concludes that it is a secure password, despite its relationship to a common word.
- **Entropy estimation**: zxcvbn does not calculate true entropy, but rather estimates it, or more precisely, estimates the number of guesses required. The most significant limitation is the fixed constant `BRUTEFORCE_CARDINALITY`, which in the original library is set to 10, thus considering only 10 possible characters regardless of the actual number of characters used in the password. This leads to incorrect entropy estimates for random passwords, particularly those with larger character sets.

These limitations stem from the library’s design philosophy of providing fast, practical password strength estimation rather than comprehensive linguistic analysis. While they affect accuracy in specific cases, they were not within the scope of this project to address, which focused on extending zxcvbn with Slovenian language data rather than modifying its core matching algorithms.

### *6.2 Future Work*

While this implementation provides comprehensive Slovenian support, several areas could be enhanced:

- **Additional languages**: The same approach could be applied to Croatian, Serbian, and other South Slavic languages, which share similar characteristics
- **Machine learning**: Pattern detection could be improved using machine learning to identify novel password patterns not captured by dictionaries
- **Integration**: The enhanced library could be integrated into popular password managers and web frameworks
- **User studies**: Empirical studies with Slovenian users could validate the effectiveness of the feedback and measure actual password strength improvements
- **Dynamic updates**: Password lists could be updated dynamically from breach databases to stay current with emerging patterns
- **Regional variations**: Support for regional Slovenian dialects and variations could further improve accuracy
- **Hash-based password lists**: Currently, zxcvbn requires plaintext passwords for analysis, which presents a significant limitation. Modern security guidelines advocate for storing only password hashes (e.g., bcrypt, Argon2) rather than plaintext, making it difficult to obtain authentic password patterns from legitimate sources. Future work could extend zxcvbn to support hash-based password frequency lists, allowing analysis of hashed passwords from breach databases without requiring plaintext access. This would enable more comprehensive password pattern analysis while maintaining security best practices and compliance with data protection regulations.

### *6.3 Impact and Contributions*

This work contributes to cybersecurity by:

- Improving password security for Slovenian-speaking users
- Providing a template for extending password strength estimation to other languages
- Demonstrating the importance of language-specific security tools
- Making password strength estimation more accessible to non-English speakers

The code, data, and documentation are available for further research and can serve as a foundation for multilingual password strength estimation. The methodology developed here can be adapted for other languages, potentially improving security for millions of users worldwide.

This work demonstrates significant potential for improvement in password strength estimation for non-English languages. However, a fundamental limitation remains: zxcvbn requires plaintext passwords for analysis, which conflicts with modern security best practices that mandate storing only password hashes. This limitation makes it challenging to obtain authentic password patterns from legitimate sources, as organizations following security guidelines cannot provide plaintext password data. Future enhancements could address this by extending zxcvbn to support hash-based password frequency analysis, enabling the use of hashed password lists from breach databases while maintaining security compliance. Such an extension would allow more comprehensive pattern analysis and better alignment with current security standards.

As password-based authentication remains prevalent despite advances in alternative authentication methods, accurate password strength estimation remains crucial. Language-specific tools like this extension help bridge the gap between security best practices and real-world password creation patterns across different linguistic and cultural contexts.

## REFERENCES

[1] Florencio, Dinei and Cormac Herley: *A large-scale study of web password habits*. ResearchGate, 2007. Accessed: 2025-01.

[2] Wang, Ding *et al.*: *Studies of keyboard patterns in passwords: Recognition, characteristics and strength evolution*. ResearchGate, 2021. Accessed: 2025-01.

[3] Wheeler, Dan: *zxcvbn: realistic password strength estimation*. https://github.com/dropbox/zxcvbn, 2012. Accessed: 2025-01.