PROJECT ABSTRACT

Readability Scorer using NLP

An NLP readability analyzer computing six classic formulas — Flesch Reading Ease, Flesch-Kincaid, Gunning Fog, SMOG, Coleman-Liau and ARI — from real linguistic counts, plus a softmax classifier assigning Elementary/Intermediate/Advanced reading level. A single-file demo analyzes pasted text live. Delivered built-to-order with code, notebook, report, PPT and viva kit.

01. Title

Readability Scorer using NLP

02. Introduction / Background

Teachers, editors and content teams share one practical problem: is this text too hard for its audience? A school textbook, a patient leaflet or a government form that overshoots its readers fails at its job — but 'too hard' is subjective until it is measured. Readability formulas turn text into numbers: sentence length, word length and syllable counts combine into grade-level scores validated against comprehension tests for decades. Modern NLP goes one step further, training classifiers on corpora labeled by reading level. This project builds both: six classic formulas computed from genuine linguistic analysis, plus a readability-level classifier, all demonstrated in a browser app that analyzes any pasted text live and shows exactly how each score was derived.

03. Problem Statement

Content is constantly written for audiences it never reaches: study material pitched above students' level, public notices the public cannot parse, documentation that assumes too much. Writers have no quick, objective way to check difficulty before publishing, and 'read it and judge' does not scale and is inconsistent between reviewers. The project addresses this by building an analyzer that measures the surface features decades of reading research link to difficulty — words per sentence, syllables per word, polysyllabic density — reports six standard scores with interpretation bands, trains a classifier to assign a reading level, and flags the specific sentences where difficulty concentrates.

04. Objectives

- Build an NLP preprocessing pipeline: sentence/word tokenization and a rule-based syllable counter handling silent-e, diphthongs and affixes.

- Implement six readability formulas from their published definitions: Flesch Reading Ease, Flesch-Kincaid Grade Level, Gunning Fog, SMOG, Coleman-Liau and ARI.
- Train a softmax classifier on linguistic features to assign Elementary, Intermediate or Advanced reading level, evaluated with accuracy and a confusion matrix.
- Build a single-file browser demo: live analysis of pasted text, metric dashboard with explanations, reference-text comparison and sentence-level highlighting.
- Ground the work in real corpora: the OneStopEnglish design (189 texts x 3 levels, Vajjala & Lucic 2018) and WeeBit (625 articles x 5 age classes) as the extension path.
- Deliver the complete student kit: source code, notebook, report, PPT and viva Q&A.

05. Proposed Methodology

The build has three parts. (1) Linguistic analysis: text is tokenized into sentences and words; a rule-based syllable counter measures syllables, and character and polysyllabic-word counts are derived. (2) Formula layer: the six formulas are computed exactly from these counts — e.g. Flesch Reading Ease = 206.835 - 1.015*(words/sentences) - 84.6*(syllables/words) — and mapped to their published interpretation bands. (3) Classifier: the same feature vector trains a softmax (multinomial logistic regression) classifier on reference texts labeled by level; train/test splits and the confusion matrix are reported in the notebook. The browser demo reimplements the whole chain in JavaScript, including in-browser training of the softmax classifier on the built-in reference texts.

06. System Architecture / Working

Pasted text enters the tokenizer, which emits sentence and word lists; the syllable counter and character counter produce the feature vector (word count, sentence count, syllable count, character count, polysyllabic count). The formula engine computes all six scores and their interpretation bands, while the softmax classifier maps the same vector to level probabilities. The demo's metric view shows every score with its formula and what drives it up or down; the analyze view highlights sentences exceeding complexity thresholds; the reference view compares built-in Elementary/Intermediate/Advanced passages side by side. The notebook documents each formula's derivation, the classifier training and the evaluation.

07. Hardware / Software Requirements

- Python 3.10 with NLTK/regex tokenization, NumPy, scikit-learn, Matplotlib (pipeline and notebook)
- Jupyter notebook for the formula walkthrough and classifier training
- Reference texts aligned with the OneStopEnglish corpus design (public academic corpus); WeeBit discussed as extension

- Any modern desktop browser for the single-file HTML/CSS/JS demo app — no server, works offline
- No special hardware required

08. Expected Outcomes

- Correct implementations of all six readability formulas with interpretation bands.
- Rule-based syllable counter accurate on ordinary English prose.
- Softmax level classifier with accuracy and confusion matrix from the buyer's own build on held-out reference texts.
- Browser demo: live analysis, metric explanations, sentence highlighting, reference comparison.
- Complete documentation: report, PPT, formula derivations and viva Q&A.

09. Applications

- Checking study material, notices and documentation against audience reading level.
- Teaching platform for NLP preprocessing: tokenization, feature engineering, classification.
- Editorial aid: pinpointing the sentences that make a draft hard to read.
- Reference build for other text-difficulty tasks (simplification, leveling).

10. Future Scope

- WeeBit-based 5-class classifier for finer age-level prediction.
- Vocabulary-rarity features using word-frequency lists for deeper difficulty modeling.
- Multilingual readability with language-specific syllable rules.
- Integration with text-simplification suggestions for flagged sentences.

11. References

- Flesch, R. — A New Readability Yardstick (Journal of Applied Psychology, 1948).
- Gunning, R. — The Technique of Clear Writing (1952); McLaughlin, G. H. — SMOG Grading (1969); Coleman, M. and Liau, T. L. — A Computer Readability Formula (1975).
- Vajjala, S. and Lucic, I. — OneStopEnglish corpus: A new corpus for automatic readability assessment (LREC, 2018).
- Xia, M. et al. — WeeBit corpus for readability assessment (labeled by age group).