BSA – Team Google

Papers – Findings

COLOR CODE:

*Yellow Highlighting = Features that Team Google considers computing*

*Blue Highlighting = Algorithms that Team Google considers training with*

*Nota Bene : Texts 4 and 9 are the two main inspiration for this project. The others are useful to correlate information and provide additional insight.*

# Comparing Machine Learning Classification Approaches for Predicting Expository Text Difficulty (2018)

## Objective

Comparing different ML approaches for automated assessment of text difficulty.

## Training data

* iStart dataset

## Methodology

* NLP tools used to identify linguistic features predictive of text difficulty
* Feature submitted to ML

## Features

* Flesch-Kincaid Grade Level (FKGL)
  + Computed using average number of syllables per word and average number of words per sentence.
  + Number of words in a sentence correlates with the effort required to read the sentence.
  + Number of syllables in a word is inversely related with word frequency and affects reading difficulty.
* Syntactic complexity
  + Mean number of words before the main verb.
* L2 Readability
* Lexical Divesity
* Word Familiarity
* Word Imagability
* Age of Acquisition

## Algorithms

* Hierarchal classification performed the best (Neural Networks or kNN)
* SVM
* Suggest trying multiple approaches because results vary with datasets

## Source

*Balyan, R., McCarthy, K. S., & McNamara, D. S. (2018). Comparing Machine Learning Classification Approaches for Predicting Expository Text Difficulty. In Grantee Submission. https://eric.ed.gov/?id=ED585216*

# Applying Natural Language Processing and Hierarchical Machine Learning Approaches to Text Difficulty Classification. (2020)

## Objective

Applying advanced AI methods to the problem of assessing text difficulty.

## Training data

* Human raters estimated text difficulty level of 262 texts across two text sets.
* iStart dataset

## Methodology

* NLP tools used to identify linguistic features predictive of text difficulty
* Features submitted to both flat and hierarchical machine learning algorithms.

## Features

* Flesch-Kincaid Grade Level (FKGL)
  + Computed using average number of syllables per word and average number of words per sentence.
  + Number of words in a sentence correlates with the effort required to read the sentence.
  + Number of syllables in a word is inversely related with word frequency and affects reading difficulty.
* Syntactic complexity
  + Mean number of words before the main verb.
* L2 Readability
* Uncommon or rare words count
* Word familiarity
* Word imageability
* Age of Acquisition

## Algorithms

* Results indicated that « hierarchical » outperformed « non-hierarchical (flat) machine learning classification ».
* Logistic Regression
* Support Vector Machine - (SVM)

*SVM constructs a hyperplane that separates the data into classes. SVMs are efficient for high-dimensional feature spaces and are among the best supervised learning algorithms*

* Linear Discriminant Analysis - (LDA)

*Method used to find a linear combination of features that characterizes or separates two or more classes of objects or events.*

* AdaBoost
* Naïve Bayes

*Naïve Bayes is based on the Bayes’ theorem of posterior probability. It is a probabilistic learning method, which assumes that the effect of an attribute value on a given class is independent of other attributes values*

## *Source*

*Balyan, R., McCarthy, K. S., & McNamara, D. S. (2020). Applying Natural Language Processing and Hierarchical Machine Learning Approaches to Text Difficulty Classification. International Journal of Artificial Intelligence in Education, 30(3), 337‑370.* [*https://doi.org/10.1007/s40593-020-00201-7*](https://doi.org/10.1007/s40593-020-00201-7)

# Evolutionary Data Measures: Understanding the Difficulty of Text Classification Tasks (2018)

## Objective

* Analyses exactly which characteristics of a dataset best determine how difficult that dataset is for the task of text classification.
* Proposes an intuitive measure of difficulty for text classification datasets which is simple and fast to calculate.

## Training data

78 text classification datasets

## Methodology

78 text classification datasets and trained 12 different ML algorithms on each of the datasets for a total of 936 models trained.

## Features

*Distinct Unigrams: Total Unigrams + Class Imbalance + Class Diversity + Maximum Unigram Hellinger Similarity + Unigram Mutual Info*

## Algorithms

* Classic algorithms + TF-IDF
* NN + Word Embeddings
* CNN + Characters

## Source

Collins, E., Rozanov, N., & Zhang, B. (2018). Evolutionary Data Measures : Understanding the Difficulty of Text Classification Tasks. arXiv:1811.01910 [cs]. http://arxiv.org/abs/1811.01910

# Automatic Text Difficulty Classifier (2015)

## Objective

System to assist the selection of adequate reading materials to support European Portuguese teaching while highlighting the key challenges on the selection of linguistic features for text difficulty.

## Training data

Texts on 5 levels scale A1 to C1 and same texts dataset on 3 level scale (A, B and C).

## Methodology

* NLP tools used to identify linguistic features predictive of text difficulty
* Feature submitted to ML

## Features

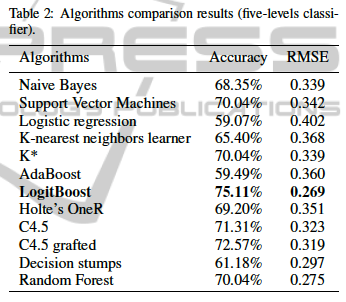
Une image contenant texte

Description générée automatiquement

Une image contenant table

Description générée automatiquement

## Algorithms



## Source

*Curto, P., Mamede, N., & Baptista, J. (2015). Automatic Text Difficulty Classifier—Assisting the Selection Of Adequate Reading Materials For European Portuguese Teaching. 36‑44.* [*https://doi.org/10.5220/0005428300360044*](https://doi.org/10.5220/0005428300360044)

# Automatic Text Difficulty Classifier (2008)

## Objective

Tool that searches the web and performs in real-time a) html-free text extraction, b) classification for thematic content and c) evaluation of expected reading difficulty.

## Training data

Hand collected dataset with labels

## Methodology

## Features

* Lix readability formula:

*The Lix readability algorithm distinguishes between five levels of readability: very easy, easy, standard, difficult, or very difficult. If W is the number of words, LW is the number of long words (7 or more characters), and S is the number of sentences, them the Lix index is LIX = W/S + (100 \* LW) / W. An index of 0-24 corresponds to a very easy text, 25-34 is easy, 35-44 standard, 45-54 difficult, and 55 or more is considered very difficult.*

* Rix readability formula:

*The Rix readability formula consists of the ratio of long words to sentences, where long words are defined as 7 or more characters. The ratio is translated into a grade level as indicated in Table (1).*

* Coleman-Liau readability formula:

*The Coleman-Liau readability formula is similar to the Rix formula in that it gives the approximate grade level of the text. Unlike the Lix and Rix formulas, the Coleman-Liau formula requires the random selection of a 100-word excerpt from the text.*

## Algorithms

* Naïve Bayes
* Maximum Entropy classifier
* MIRA

## Source

*Miltsakaki, E., & Troutt, A. (2008). Real Time Web Text Classification and Analysis of Reading Difficulty. Proceedings of the Third Workshop on Innovative Use of NLP for Building Educational Applications, 89‑97.* [*https://www.aclweb.org/anthology/W08-0911*](https://www.aclweb.org/anthology/W08-0911)

# A Language Modeling Approach to Predicting Reading Difficulty

## Objective

## Predict the reading difficulty of a text

## Training Data

## Taken online from different language teaching institutions, already labeled

## Methodology

## Smooth frequencies and submit the features to ML

## Features

* Keep stopwords
* remove words with frequency less than 3 times in the corpus
* remove words appearing only in a range of 3 grades (but here it is for a scale on 12 grades total)

## Algorithms

## Naïve Bayes (smoothed unigrams model) with BOW

## Simple Good-Turing

## Source

# CEFR-based Lexical Simplification Dataset

## Objective

## Construct a dataset of words in order to replace complicated words in a text, thus simplifying it.

## Training Data

## Randomly selected introductory chapters of university textbook from OpenStax website by Rice University

## Methodology

## Classify candidates based on 3 stages: grammatical reformation, definition and context, then apply:

## - Language Model (LM) using Google N-gram

## - Word2Vec with cosine similarity of candidates against targets vectors

## Features

## Level of a work

## Similarity of words based on the 3 stages

## Algorithms

## Creation of their own classification algorithm based on LM and W2V

## Source

*Uchida, Satoru, Shohei Takada, et Yuki Arase. « CEFR-Based Lexical Simplification Dataset », s. d., 5.* [*https://www.aclweb.org/anthology/L18-1514.pdf*](https://www.aclweb.org/anthology/L18-1514.pdf)

# SW4ALL: a CEFR-Classified and Aligned Corpus for Language LearningO

## Objective

## Classify texts according to their complexity

## Training Data

## EFCAMDAT (Cambridge) as training corpus. 9000 texts for each level

## Methodology

## Gather two types of Wikipedia text corpus, then build a classifier using machine learning.

## Features

## annotate text in four categories: length-based, morphological, syntactic, and readability.

## results suggest that length-based features have great relevance for the level classification

## Algorithms

## Simple Logistic Regression

## Random Forest

## Source

# Automatic Classification of Text Complexity

## Objective

## Classify texts according to their complexity

## Training Data

## Corpus of Italian texts already labelled by CVCL (linguistic institution)

## Methodology

## Filter corpus by number of texts, types (unique tokens through the dataset) and tokens for each level.

## NLP pipeline library (UDPipe), then ML with 10 different classification models

## 

## 

## Features

## Standardization of the features

## cross validated Recursive Feature Selection algorithm that to remove the weakest features (more than half)

## Algorithms

## 

## Source