Automatic Readability Checker

Muskan Arora[1] Pranav Kanungo[2] Mohti Jatav[3] Dr. Santosh Varshney[4] Prof Kavita Namdev[5]

[1,2,3]Student [4]Project Guide [5]Project Coordinator

[1,2,3,4,5]Department of Computer Science and Engineering

[1,2,3,4,5]Acropolis Institute of Technology and Research, Indore, Madhya Pradesh, India

Abstract— Readability is a measure of how easy a piece of text is to read. It can includes complexity, familiarity, legibility, and typography. The written word is used to communicate a whole host of ideas and information. But, what if without even being aware of it, your writing was stopping people engaging with your content, readability scores measure whether the content is likely to be understood by your intended reader. When text is too difficult or awkward to read, messages may not be engaged with or understood. On the flip side, when writing is too simplistic, your audience might feel patronized or just plain bored. Either way, the readability of a given text influences the extent to which people engage with and take on a message.

Keywords: Readability, Understandability, Natural Language Processing

I. INTRODUCTION

The readability report scores your document for readability, cohesion and information density. These scores provide the author with an indication of how well their intended audience will understand their text. Many writers, professionals, and amateurs face the dilemma of writing materials above the reading level of their readers. The best way to write for your target audience is to learn how to make your writing more readable by removing unnecessary words, using more familiar words, and sticking to a shorter sentence length. If automatic readability checkers could be built, they could be integrated into development tool-chains, and thus continually inform developers about the readability level of the code. Unfortunately, readability is a subjective code property, and not amenable to direct automated measurement. In a recently published study, Buse et al. asked 100 participants to rate code snippets by readability, yielding arguably reliable mean readability scores of each snippet; they then built a fairly complex predictive model for these mean scores using a large, diverse set of directly measurable source code properties.

A readability score can tell you what level of education someone will need to be able to read a piece of text easily. The score identifies a grade level approximate to the number of years of education a person has had.

Our project is based on the rules which are defined universally for checking a particular document. The application will help us to determine what is the level of a particular article/story and which age group can read that article/story. It will also help writers to check whether they are targeting the right audience or not and whether the piece is readable by that audience or not. The application will be providing grading from 1-10 based on your text and with that user can get a fair idea who can read the text.

II. LITERATURE SURVEY

Readability is the ease with which a reader can understand a written text. In natural language, the readability of text depends on its content (the complexity of its vocabulary and syntax) and its presentation (such as typographic aspects like font size, line height, and line length). Researchers have used various factors to measure readability, such as

1) Speed of perception
2) Perceptibility at a distance
3) Perceptibility in peripheral vision
4) Visibility
5) Reflex blink technique
6) Rate of work (reading speed)
7) Eye movements
8) Fatigue in reading

Readability is more than simply legibility—which is a measure of how easily a reader can distinguish individual letters or characters from each other.

Higher readability eases reading effort and speed for any reader, but it is especially important for those who do not have high reading comprehension. In readers with average or poor reading comprehension, raising the readability level of a text from mediocre to good can make the difference between success and failure of its communication goals.

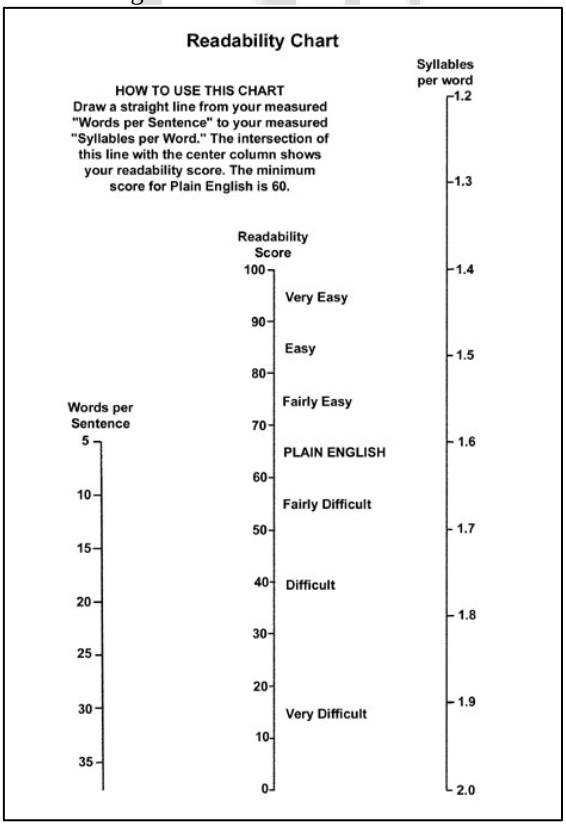

Fig. 1: Showing How Grading is done

Readability is the ease with which a reader can understand a written text. In natural language, the readability of text depends on its content (the complexity of its vocabulary and syntax) and its presentation.

That readability formulas stem are strictly text-based does not reflect the interactive nature of the reading process. Popular formulas employ only a syntactic (sentence length) and a semantic factor (vocabulary diversity). Formulas do not directly address factors to communicate of meaning, nor can they discriminate between written discourse and nonsensical combinations of words. Moreover, formulas cannot address reader-related factors — interest, experience, knowledge, and motivation.

Readability formulas are objective, quantitative tools for estimating the difficulty of written material without requiring to test the reader. You can assess texts involving a wide range of content and prose styles through readability formulas. Formulas stem from interest in matching reader ability and text difficulty.

In the existing system, readability is defined by a formula based approach in which total words and sentences are used to determine who can read the particular text.

$$206.835 - 1.015\left(\frac{\text{total words}}{\text{total sentences}}\right) - 84.6\left(\frac{\text{total syllables}}{\text{total words}}\right)$$ [7]

Scores can be interpreted as shown in the table below.[8]

Score	School Level	Notes
100.00-90.00	5th grade	Very easy to read. Easily understood by an average 11-year-old student.
90.0–80.0	6th grade	Easy to read. Conversational English for consumers.
80.0–70.0	7th grade	Fairly easy to read.
70.0–60.0	8th & 9th grade	Plain English. Easily understood by 13- to 15-year-old students.
60.0–50.0	10th to 12th grade	Fairly difficult to read.
50.0–30.0	College	Difficult to read.
30.0–0.0	College Graduate	Very difficult to read. Best understood by university graduates.

Fig. 2: Process Flow of the Existing System

The limitations of Existing System are that the system was unable to give any physiological explanation of how the formula was defined. The reading ability is also based on how words are phrased n sentence and not only just by word count. Use of different vocabulary word also affect the readability, this was not taken into consideration.

III. RESEARCH METHODOLOGY

As we saw readability, focuses on textual content such as lexical, semantical, syntactical and discourse cohesion analysis. It is usually computed in a very approximate manner, using average sentence length (in characters or words) and average word length (in characters or syllables) in sentences.

We applied supervised learning to get the desired outputs. As we know Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. It infers a function from labelled training data consisting of a set of training examples. The begin working with any supervised machine learning algorithm first most step is to gather data and give it to the machine as an input. For our project our sources were database at of approx. 100 english books which include both ‘easy’ to read books and ‘difficult’ to read books. A dataset of identifiers, graded on the Lexile text complexity scale which is predefined scale.

Our dataset assumes that the Lexile score is close to the actual readability perception of an average person, which might not be, because here we are using mainly two features: sentence length (number of words in sentence, long sentences are automatically considered to be more complex and difficult) and words frequency.

A. Tokenization and Standardization

We first began with cleaning tokenizing the text into sentences and words. Then we represented the text as an array of numbers to make it as feature vector.

Each book was represented by a vector of 50 float numbers, each of them being a text feature:

1) Mean number of syllables per word
2) Mean number of words per sentence
3) Mean number of words considered ‘difficult’ in a sentence (a word is ‘difficult’ if it is not part of an ‘easy’ words reference list)
4) Part-of-Speech (POS) tags count per book
5) Most common Readability formulas such as Flesch-Kincaid and SMOG
6) Polysyllable count (more than 3 syllables).

These features are all on different scales, therefore to have a similar scale from -1 to 1 we follow a Standardization process. In this we remove the mean and divide by the standard deviation of the dataset. This is done because some of the algorithms we use during modelling assume that the data given as input follows a Gaussian distribution.

B. Feature selection

The data features that we use to train our machine learning models have a huge influence on the performance we can achieve. Feature Selection is the process where we automatically or manually select those features which contribute most to our prediction variable or output in which we are interested in.

After building a set of features representing a text, we reduce the vector to only the most salient features; the ones that discriminate the most our annotations. Using features that do not carry information related to the readability score is a computation time burden to the model, and may also give our inconsistent results. This is might happen if the inference is done with more features than necessary.

To perform this feature selection step, we use the LASSO method (scikit-learn implementation) with cross-validation or CV. CV is the process of training and testing models with different data splits to avoid a bias from a specific dataset order.

Here we train model on 70% of labelled data and evaluate the trained model on the remaining 30%

K-fold cross-validation improves on this by letting you do this multiple times so you can see whether the test performance varies based on which samples you used to train / test.

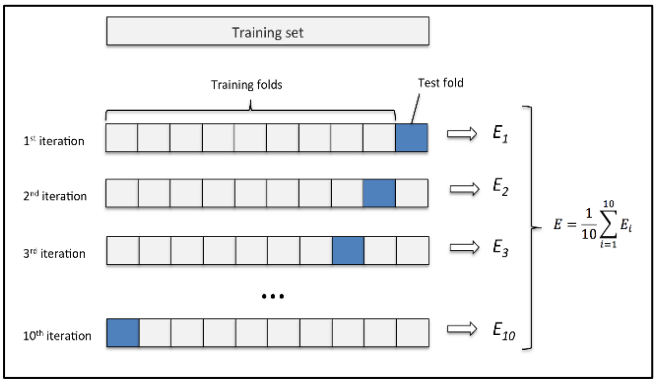

Fig. 3: K-fold Cross-Validation

The LASSO method is performed by creating multiple subsets of our feature set. For each feature set a regression function is fitted using our training data. Then a correlation is computed (using a metric such as Person, Kendall-Tau or Chi-Square) between each set's regression function and the readability score. Feature sets are ranked by correlation performance and the best one is selected.

C. Choosing the right model

Our output variable is numerical and continuous which narrows the spectrum of machine learning models applicable to our dataset (regression task). To select an appropriate model, there is several indicators that may guide one's choice, such as the number of features or the number of samples available.

In the case of constrained bayesian algorithms such as Naive Bayes variants (simple or tree augmented), performances are likely to decrease with large number of features. This is due to their inability to build large variable dependencies between an output variable and an explanatory variable. Naive Bayes is built under the assumption that variables are independent, which is less likely the case with longer feature vectors. Tree Augmented Naive Bayes (TAN) allows only one explanatory variable as a dependency of another to predict an output variable. This lack of feature intrication makes these algorithms bad candidates for our feature vector length.

However, Decision Tree (DT) based algorithms cope very well with high dimensional data (more features) but need lots of data samples (varies as a function of algorithm hyperparameters). DTs build rules (for example: average number of words per sentence > 5) and these rules are split when a given amount of data samples fit them. In decision tree algorithms, the number of data samples is a function of model granularity, by handling overfitting correctly, the more data and features there is, the better a DT based model is.

Another approach to model selection that we choose to use is Grid Search, this technique is a training and testing brute force over a set of models and a set of hyper parameters for each model.

In our Grid Search, three algorithms compete: a Random Forest Regressor (4 hyper parameters), a Linear Regression and a Support Vector Regressor (2 hyper parameters), the best model is generated through Random Forest regression.

We now have a production grade model that takes a book's feature vector as input and gives a readability score as output.

We will use the metric R^2, also known as coefficient of determination, is the metric we will use to test our regression algorithm. The resulting value we get from it ranges from 0 to 1 and Random Forest is optimised to converge to 1. This value is the explained variance accounted by our model: the higher it is, the less test data samples we find outside of our model's prediction error range.

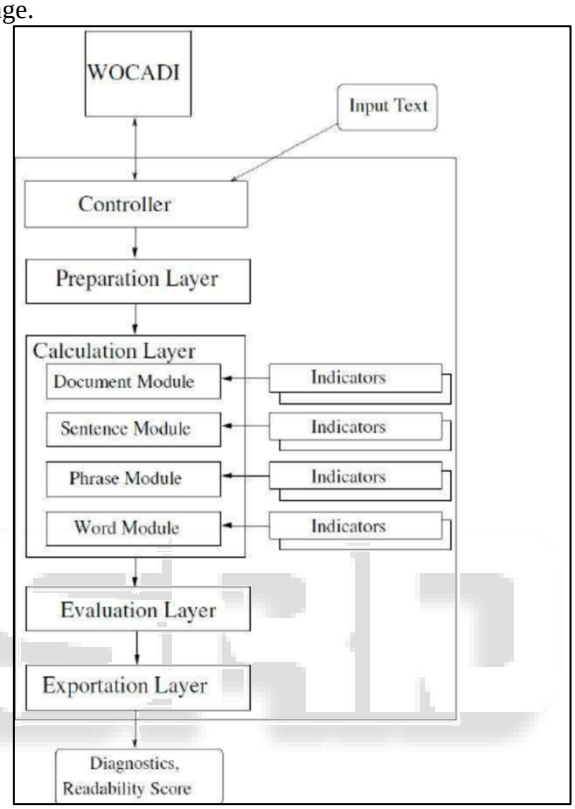

Fig. 4: Working

IV. CONCLUSION

Determining text complexity can be a valuable tool for us as it has many applications. Right from designing content for brochure to choosing right content for children's books. After trying various models we concluded that Random Forest Model gave the best results. Feature and model selection can be varied further to improve the results. It is a key step to get the right output. Selecting different features can also change the functionality of the project.

REFERENCES

[1] https://pdfs.semanticscholar.org/ac3a/db8d2fe1c175e039ceeb49c5f723a6a86ee7.pdf
[2] https://readable.com/blog/what-is-readability/
[3] https://en.wikipedia.org/wiki/Readability
[4] https://libraries.io/pypi/textstat
[5] https://pdfs.semanticscholar.org/ac3a/db8d2fe1c175e039ceeb49c5f723a6a86ee7.pdf
[6] http://www.codeguru.com
[7] https://medium.com/glose-team/how-to-evaluate-text-readability-with-nlp-9c04bd3f46a2

[8] https://hub.lexile.com/find-a-book/search
[9] http://karlrosaen.com/ml/learning-log/2016-06-20/
[10] http://languageartsreading.dadeschools.net/pdf/FAIR/LexileConversionChart.pdf
[11] https://www.wizenoze.com/language/en/readability-classification/
[12] https://extensions.openoffice.org/en/project/readability-report
[13] https://www.readabilityformulas.com/
[14] Alawad, Duaa. (2018). An Empirical Study of the Relationships between Code Readability and Software Complexity.
[15] Lin, Jin-Cherng & Wu, Kuo-Chiang. (2006). A Model for Measuring Software Understandability. 192 - 192. 10.1109/CIT.2006.13.
[16] Ryser-Welch, Patricia. (2015). Discovering human-readable algorithms for the Traveling Salesman Problem using Cartesian Genetic Programming. 10.13140/RG.2.1.3766.1527.
[17] https://www.readabilityformulas.com/practical-purposes-of-using-readability-formulas.php
[18] Mohammad Sajid. "Machine Learning In Python – Sklearn Feature Selection." Mar 2, 2017, https://stepupanalytics.com/feature-selection-for-machine-learning-in-python/
[19] Rajat Sharma. "Can we predict Stock Price using Deep Learning?." Jun 22, 2019, https://medium.com/dataseries/can-we-predict-stock-price-using-deep-learning-54e26df8e50b
[20] Marc Benzahra. "How to Evaluate Text Readability with NLP." Jun 20, 2019, https://medium.com/glose-team/how-to-evaluate-text-readability-with-nlp-9c04bd3f46a2

