Parameter Learning for a Readability Checking Tool

Tim vor der Brück and Johannes Leveling
Intelligent Information and Communication Systems (IICS)
FernUniversität in Hagen (University of Hagen)
58084 Hagen, Germany
{tim.vorderbrueck, johannes.leveling}@fernuni-hagen.de

Abstract

This paper describes the application of machine learning methods to determine parameters for DeLite, a readability checking tool. DeLite pinpoints text segments that are difficult to understand and computes for a given text a global readability score, which is a weighted sum of normalized indicator values. Indicator values are numeric properties derived from linguistic units in the text, such as the distance between a verb and its complements or the number of possible antecedents for a pronoun. Indicators are normalized by means of a derivation of the Fermi function with two parameters. DeLite requires individual parameters for this normalization function and a weight for each indicator to compute the global readability score.

Several experiments to determine these parameters were conducted, using different machine learning approaches. The training data consists of more than 300 user ratings of texts from the municipality domain. The weights for the indicators are learned using two approaches: i) robust regression with linear optimization and ii) an approximative iterative linear regression algorithm.

For evaluation, the computed readability scores are compared to user ratings. The evaluation showed that iterative linear regression yields a smaller square error than robust regression although this method is only approximative. Both methods yield results outperforming a first manual setting, and for both methods, basically the same set of non-zero weights remain.

1 Introduction

Cognitive difficulties for readers are often approximated by a readability function returning a text readability score. The calculation of such a function is typically done in two steps [Flesch, 1948; Chall and Dale, 1995]:

- Determine several indicators for reading difficulty from the surface structure of the text (usually including indicators such as average sentence length and word average length).
- Compute a linear combination of weighted indicator values.

Readability scores have a long history and tradition, especially in English-speaking countries.

The parameters of a readability function may be derived automatically as follows. Given a set of user ratings for a certain text corpus, linear regression can be applied to derive the parameters, minimizing the square difference between the user ratings and the readability score.

A well-known example of a readability function following this schema is the Flesch Reading Ease Score [Flesch, 1948] for English texts, given in equation 1. It is based on computing two indicators from the surface structure, namely the average sentence length (ASL) and the average word length (AWL). For German, similar formulas exist to test the readability of texts (e.g. Amstad [1978]).

$$R_{\text{Flesch}} = 206.835 \quad - \quad (1.015 \cdot \text{ASL}) \quad - \quad (0.846 \cdot \text{AWL}) \tag{1}$$

Readability functions of this type have several drawbacks. First, the weights have no intuitive meaning. Therefore, they are difficult to interpret and would be difficult to adjust manually. Second, a large number of indicators in such a formula can easily lead to overfitting, which means that additional work is required to reduce the number of indicators to an optimal set.

For DeLite, our readability checking tool for German texts, a different approach is employed. Before the indicator values are combined they are mapped into the interval [0, 1], which avoids the drawbacks described above and allows a comparison of weights, e.g. for different types of readers (for a detailed description see Section 4).

2 Readability Score and Indicators

DeLite is a readability checking tool for German texts. Its graphical user interface is shown in Figure 2. The readability checking in DeLite relies on a linguistic analysis of text documents with the syntactico-semantic parser WOCADI [Hartrumpf, 2003]. The experiments described here were performed largely on German texts, because the natural language processing tools rely on German resources, e.g. a large German semantic lexicon. WOCADI parses texts and returns their semantic representation, including analysis results corresponding to the morphologic, lexical, syntactic, semantic, and discourse level of linguistic units such as words, phrases, or sentences. Natural language processing results are represented as semantic networks based on the MultiNet paradigm [Helbig, 2006]. These analysis results serve as a basis to derive 47 readability indicators, which represent measurable properties of linguistic units. Indicators are associated with one of the different levels of linguistic analysis given above. Table 1 shows some typical examples of indicators. The readability indicators and

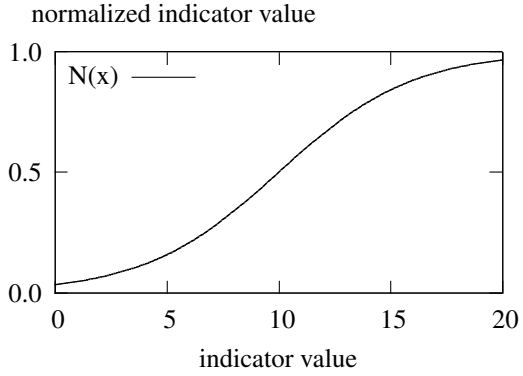

Figure 1: Normalizing function $N(x)$ derived from the Fermi function for $\mu = 10$, $\delta = 3$.

their computation from natural language processing results of the parser are described in more detail by Hartrumpf *et al.* [2006] and Jenge *et al.* [2006].

3 Data Normalization

Unnormalized indicators have vast differences in their value distribution, mean value and variance, e.g. the number of concepts in a compound usually varies between two and five while the number of nodes in a semantic network, representing a sentence, can easily exceed 20. Therefore, indicator values are normalized, mapping them into the interval [0, 1]. A simple method to normalize indicator values would be to employ a linear transformation based on the maximum and minimum values. However, this may not be a reliable solution for several reasons: In new texts, indicator values may exceed the known extreme values. Usually such values are mapped to either zero or one. But in this case, the normalization function will be no longer differentiable on the whole value range, which makes it difficult to apply non-linear optimization techniques like least-squares estimation [Greene, 1993]. Furthermore, this approach becomes very sensitive to outliers.

Many of these difficulties are avoided by the function $N(x)$ used in DeLite (see equation 2), a derivation of the Fermi function. Figure 1 shows the graph of this function for the parameter values $\mu = 10$ and $\delta = 3$.

$$N(x) = \frac{1}{1 + e^{-\frac{x-\mu}{\delta}}} \tag{2}$$

The parameter μ is the location of the 0.5-intercept ($N(\mu) = 0.5$) and δ specifies the incline of the function. For simplicity, its is presumed that indicator values are non-negative and that high unnormalized indicator values correspond to less readability.

One approach to determine the parameters consists of applying a nonlinear optimization to all constituents of the weighted sum and compute both weights and parameters simultaneously. Several (in)equality constraints have to be defined because all weights are expected to be non-negative and normalized to sum up to one. However, estimating more than 140 parameters (three for each indicator) with a constrained nonlinear optimization algorithm is quite difficult and also rather slow.[1] For DeLite, a more efficient approach is chosen, which is also guaranteed to converge.

[1] Weight learning may have to be repeated several times, e.g. for user groups with different cognitive impairments.

The parameter estimates derived by DeLite could be employed as an initial parameter guess for a nonlinear optimization problem as described above.

The parameter μ_j of the normalization function for a given indicator I_j determines the 0.5-intercept. It usually corresponds to some point near the center of the distribution of the indicator values. Several methods to calculate the parameters μ_j and δ_j of the individual normalization functions were tested, including techniques based on analyzing conditional probabilities, utilizing quantiles, the median, and mean value. Selecting the *mean value* of the distribution for μ_j yielded the smallest error and proved to be quite robust to outliers. The parameter δ_j was obtained by computing the arithmetic mean for solutions of $N_j(x)$ for given values of μ and maximum and minimum values of the indicator value I_j under consideration.

4 Data Combination

In DeLite, a readability score R for a text is calculated as a weighted sum, combining all normalized indicator values v_j. Equation 3 shows the general structure of this function.

$$R = \sum_{j=1}^{m} w_j v_j \tag{3}$$

In the remainder of this paper, normalized weights are assumed, i.e. $w_1 + \ldots + w_m = 1$ and $w_1 \geq 0, \ldots, w_m \geq 0$.

To compute the readability score R, the weight w_j and the normalization parameters μ_j and δ_j have to be determined for each indicator I_j individually. Several machine learning approaches to accomplish this are described and evaluated in Section 5 and Section 6.1.

Note that the indicator weights reflect the importance of each indicator with respect to global readability of a text. If necessary, it would be easy to support manual adjustments, i.e. changing user preferences via the user interface.

There is no need to determine the best set of indicators. After the training period, all indicators with a weight of zero are automatically eliminated from the readability function, i.e. they do not contribute to the readability score. However, all readability indicators – regardless of their weight – are utilized to identify and pinpoint text passages that are difficult to read.

5 Weight Learning

5.1 Problem Description

The parameters of the normalization function are determined as described in Section 3. Thus, to compute the text readability score R, only the indicator weights (w_j) in the weighted sum remain to be found. Basically, two types of machine learning algorithm have been applied to solve such types of problems. These are on the one hand algorithms that depend on a specific probability distribution and on the other hand algorithms which make no such assumption. A method of the first type is for instance the Expectation Maximization algorithm (EM, see Dempster *et al.* [1977]). This algorithm cannot be applied on data where the indicators are highly correlated among each other. A transformation technique like Principal Component Analysis (PCA, see [Jolliffe, 1986]) is necessary in this case to create a new data set with independent indicators.

Since different indicators also have varying probability distributions, an approach of the second type is preferred, which includes regression techniques. Regression can also

Table 1: Linguistic levels of analysis and corresponding indicators.

Linguistic level	Indicator (German example/English translation, value)
Morphologic	Number of concepts in a compound ('*Mehrwertsteuererhöhungsdiskussion*'/'*discussion to increase value added taxes*', 4)
Lexical	Word frequency class ('*Stadtverwaltungen*'/'*municipal administration*', 36)
Syntactic	Number of syntactic readings of a sentence ('*Polizei erschoss Mann mit Gewehr*'/'*Police shot man with gun*', 2)
Semantic	Number of propositions per sentence ('*Die Familie besuchte die Tante und übernachtete dort*'/'*The family visited the aunt and spent the night there*', 2)
Discourse	Number of reference candidates for a pronoun ('*Jutta und Maria trafen sich in ihrem Haus*'/'*Jutta and Maria met in her/their house*', ≥ 2 for the pronoun '*ihrem*')

be used on highly correlated indicator values without the necessity of any data transformation. However, for most types of regression algorithms the indicator values still have to be linearly independent of each other.

In common optimization algorithms, the optimal weights are determined by minimizing the square error (see equation 4).

$$w_{\text{opt}} = arg\min_{w}(\sum_{i=1}^{n}(y_i - X_i w)^2) \tag{4}$$

The variables given above have the following meanings:

- n: The number of indicators.
- m: The number of rated texts.
- y_i: The average user rating for text i. This value is determined from the global readability ratings by the users. Values of the discrete seven-point Likert scale (Likert [1932], see Section 6.1) are converted into a numeric value between zero and one by a linear transformation. A value of one represents optimal, a value of zero the worst readability.
- X_i : Vector notation for $(x_{i1}, \ldots, x_{im})$. x_{ij} is an indicator value between zero and one for indicator I_j and text i.
- w : Vector notation for $(w_1, \ldots, w_m)$. w_j is the weight for the indicator I_j.

Because all weights are required to be non-negative, simple linear regression cannot be employed. Two alternative approaches are investigated: robust regression with linear optimization (see Section 5.2), and an approximative iterative linear regression based method (see Section 5.3).

5.2 Robust Regression with Linear Optimization

Robust regression leads to estimating parameters by minimizing the sum of the absolute error instead of the square error. The minimization can be achieved via linear optimization, usually applying the Simplex algorithm [Bertsimas and Tsitsiklis, 1997]. This kind of regression is called robust, since it is not as sensitive to outliers as linear regression.

The minimization problem for determining the weights of our readability function can be defined as follows:

$$w_{\text{opt}} = arg\min_{w}(\sum_{i=1}^{n}(|y_i - X_i w|)) \tag{5}$$

In equation 5, $|y_i - X_i w|$ can be replaced by variables z_i, if the constraints $z_i \geq |y_i - X_i w|$ are added (see Bertsimas and Tsitsiklis [1997]). Using the equivalence in equation 6, the optimization problem can be rewritten as shown in equation 7.

$$z_i \geq |y_i - X_i w| \Leftrightarrow z_i \geq (y_i - X_i w) \wedge z_i \geq -(y_i - X_i w) \tag{6}$$

$$\begin{array}{rcl} & & arg\min_w z_1 + \ldots + z_m \text{ , with} \\ z_i & \geq & x_{i1}w_1 + \ldots + x_{im}w_m - y_i \text{ ,} \\ z_i & \geq & y_i - x_{i1}w_1 - \ldots - x_{im}w_m \end{array} \tag{7}$$

and $i = 1, ..., n$.

This problem consists of linear equations only and can therefore be solved by traditional linear optimization algorithms.

5.3 Iterative Linear Regression

In this section, an approximative solution by using a restricted linear regression problem is discussed. A general restricted linear regression problem is given by equation 8. L contains the coefficients of one or several linear equality restrictions. In addition, the restriction that all weights sum up to one must be represented. Thus, L is set to the vector $(1, ..., 1)$. q represents the values of Lw, in our case $q = (1)$. The regression can be solved by equation 9 (see Greene [1993]).

$$W = \begin{bmatrix} X'X & L' \\ L & 0 \end{bmatrix} \quad u = \begin{bmatrix} X^T y \\ q \end{bmatrix}$$

$$W \begin{bmatrix} w \\ \lambda \end{bmatrix} = u \tag{8}$$

$$\begin{bmatrix} w \\ \lambda \end{bmatrix} = W^{-1}u \tag{9}$$

Note that the resulting weights might be negative. Negative weights may have one of the following reasons: First, they might occur if some of the indicators are not correlated with our output (the readability score). Second, they may result if some indicators are strongly correlated among each other. The first problem is avoided by setting indicator weights to zero for indicators which are not correlated with the readability rating R, effectively eliminating the corresponding indicators. The regression described above only has to be applied on the remaining indicators.

The following iterative algorithm is proposed to solve the second problem:

1. Execute the restricted regression as described above.
2. Determine all negative weights and remove the corresponding indicators from the regression model.

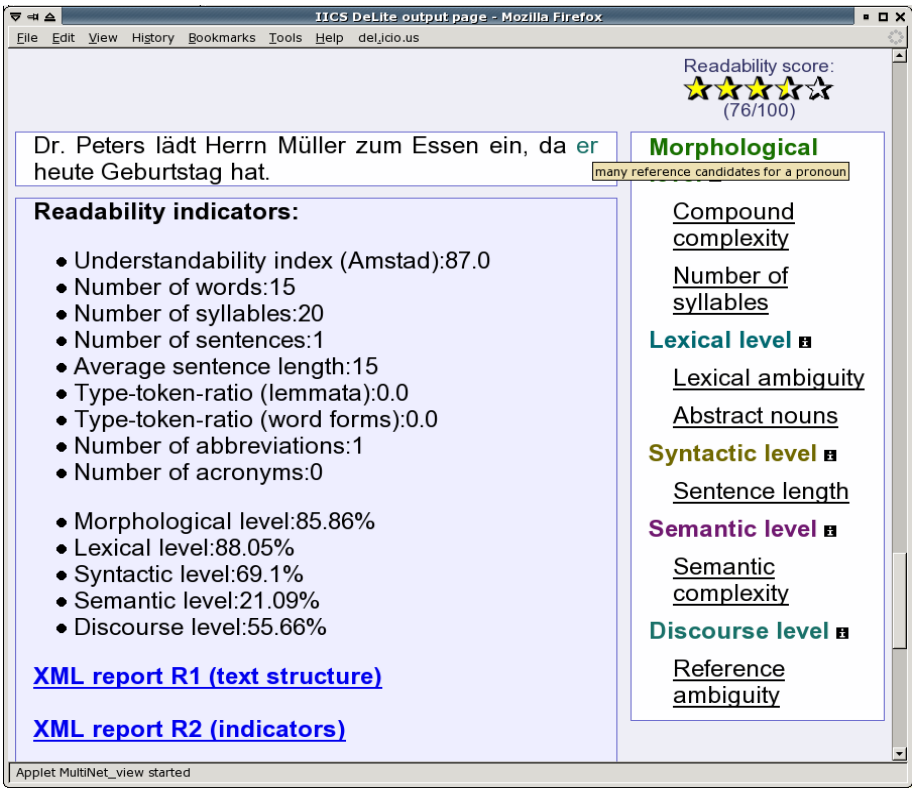

Figure 2: Graphical user interface of the DeLite readability checking tool.

3. If any negative weights are found, continue with step 1.
4. Return the set of computed weights.

A further improvement of this method may be to remove indicators which are most correlated to indicators with negative weights at every iteration, since very highly correlated (normalized) indicators are nearly exchangeable. However, in the worst case the performance becomes exponential to the number of indicators, since in every iteration several solution paths have to be followed.

6 Evaluation and Implementation

6.1 Evaluation of Parameter Learning

Training data was collected via a web experiment in which participants were asked to answer questions on the readability of given short texts. The participants in the web experiment were asked to judge the global readability of a text. Answers were given on a seven-point Likert scale labelled '*I strongly agree*', '*I agree*', '*I agree somewhat*', '*Undecided*', '*I disagree somewhat*', '*I disagree*', and '*I strongly disagree*'.

The training data for the weight learning approaches consists of user ratings of 500 texts, primarily originating from the municipality domain. The user ratings were obtained from more than 300 participants in a web experiment. The data contains more than 2800 readability ratings.

The evaluation consists of measuring the absolute and square error between the user ratings and the readability scores calculated with weights learned by either iterative linear regression or robust regression. The approximative iterative linear regression method leads to very good results in practice: It always yields a smaller square error than computing scores with the weights found by the robust regression algorithm. Table 2 shows absolute and square error for both methods together with the weights for the remaining indicators. Note that only a small number of the 47 indicators remain for computing the readability score. There are several reasons for this effect, including data sparseness and missing robustness for the semantic analysis of the texts, which causes some indicators to be available for a subset of textual units only. The table shows results for a three-fold cross-validation (CV) as well.

Additionally, the user ratings were compared to the scores obtained from a German variant of the Flesch Reading Ease Score, the Amstad understandability index (Equation 10, see Amstad [1978]).

$$R_{\text{Amstad}} = 180 - \text{ASL} - \text{ASW} \cdot 58.5 \qquad (10)$$

The relative and absolute errors for the Amstad index are 0.203 and 0.245, respectively. The correlation between user ratings and the Amstad index amounts to 0.165. This relatively low correlation may reflect that the Amstad index is not an adequate measure of text understandability, especially concerning texts of our selected municipal domain. DeLite's readability scores have a higher correlation with user ratings, and in comparison, the absolute and square errors are considerably lower (also shown in Table 2). These improvements are mainly due to a larger number of indicators and to indicators resulting from deep natural language processing methods, i.e. indicators on the semantic and discourse level.

In summary, if applied on the training data, the robust regression algorithm yields a lower absolute error than iterative linear regression, while iterative linear regression yields a lower square error. Since the differences between errors from both methods are very small, this assertion cannot necessarily be made if those methods are applied on new data which is also shown by the cross-validation.

Table 2: Weights learned by robust and iterative linear regression.

	Learning algorithm	
Normalized weight	Robust regression	Iterative linear regression
w_1	0.130	0.084
w_2	0.153	0.176
w_3	0.035	0.020
w_4	0.032	0.031
w_5	0.026	0.068
w_6	0.169	0.143
w_7	0.181	0.133
w_8	0.065	0.058
w_9	0.138	0.159
w_{10}	0.010	0.013
w_{11}	0.029	0.086
w_{12}	0.029	0.029
w_{13}	0.003	0.000
Absolute error	0.126	0.127
Square error	0.159	0.157
Absolute error (CV)	0.142	0.141
Square error (CV)	0.177	0.176

Starting with all 47 indicators, only 13 indicators remain as factors of the readability function when using robust regression (twelve if using iterative linear regression). In the DeLite implementation, the iterative linear regression algorithm is more than ten times faster than the robust regression.

6.2 The Readability Checking Tool DeLite

The readability formula as described above is used in the readability checking tool DeLite. DeLite calculates the global readability score and highlights text passages for which the indicator value exceeds a certain threshold. Figure 2 shows the graphical user interface of the readability checking tool. In the upper right corner, the readability score is displayed as a sequence of stars as well as a numerical value. On the top left, the input text is shown, which consists of a relatively simple text with a single sentence (‘*Dr. Peters lädt Herrn Müller zum Essen ein, da er heute Geburtstag hat*’/‘*Dr. Peters invites Mr. Müller to diner because he has birthday today.*’). Below the input text, several readability scores and indicator values are shown, including the Amstad readability index. On the right side, a number of indicators is aligned under the corresponding linguistic level. If selected, the text passages violating readability are highlighted. In the example, the pronoun ‘*er*’ is highlighted, because there are two reference candidates (‘*Dr. Peters*’ and ‘*Mr. Müller*’), which affects a reader’s cognitive ability to understand this text.

7 Conclusion and Outlook

In this paper, novel approaches to determine weights for a readability function were investigated. When using normalization, the importance of each indicator is denoted by its weight, which allows to adapt settings manually. Furthermore, a manual selection of a subset of readability indicators to avoid overfitting is no longer necessary.

Two methods to determine parameters and weights for the readability function were evaluated. The iterative linear regression technique outperforms the linear optimization at the minimization of the average square error. Using the linear regression method, only twelve of a total of 47 indicators remain to be computed (i.e. with a non-zero weight), with linear optimization, 13 indicators have a non-zero weight, including all twelve indicators with non-zero weights determined by linear regression.

For future work, we need to perform significance tests to see if one method performs significantly better than the other. We also intend to integrate nonlinear optimization techniques. Finally, we plan to perform experiments with different user groups sharing the same type of cognitive impairments to see which indicators are affected, i.e. which readability indicators are weighted differently compared to settings for a group of typical users and correspond to the type of cognitive impairment.

Acknowledgments

We wish to thank our colleagues at the FernUniversität in Hagen for their support, especially Sven Hartrumpf, Hermann Helbig and Rainer Osswald. The work on DeLite has been funded by the EU project ‘*Benchmarking Tools and Methods for the Web*’ (BenToWeb, FP6-004275).

References

Toni Amstad. *Wie verständlich sind unsere Zeitungen?* PhD thesis, Universität Zürich, 1978.

Dimitris Bertsimas and John Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, Belmont, Massachusetts, USA, 1997.

Jeanne Chall and Edgar Dale. *Readability Revisited: The New Dale-Chall Readability Formula*. Brookline Books, Brookline, Massachusetts, USA, 1995.

Arthur Dempster, Nan Laird, and Donald Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1), 1977.

Rudolf Flesch. A new readability yardstick. *Journal of Applied Psychology*, 32:221–233, 1948.

William Greene. *Econometric Analysis*. Prentice Hall, Englewood Cliffs, New York, USA, 1993.

Sven Hartrumpf, Hermann Helbig, Johannes Leveling, and Rainer Osswald. An architecture for controlling simple language in web pages. *eMinds: International Journal on Human-Computer Interaction*, 1(2):93–112, 2006.

Sven Hartrumpf. *Hybrid Disambiguation in Natural Language Analysis*. Der Andere Verlag, Osnabrück, Germany, 2003.

Hermann Helbig. *Knowledge Representation and the Semantics of Natural Language*. Springer, Berlin, Germany, 2006.

Constantin Jenge, Sven Hartrumpf, Hermann Helbig, and Rainer Osswald. Automatic control of simple language in web pages. In Klaus Miesenberger, Joachim Klaus, Wolfgang Zagler, and Arthur Karshmer, editors, *Proceedings of the 10th International Conference on Computers Helping People with Special Needs (ICCHP 2006)*, volume 4061 of *Lecture Notes in Computer Science*, pages 207–214, Berlin, Germany, 2006. Springer.

Ian T. Jolliffe. *Principle Component Analysis*. Springer, Berlin, Germany, 1986.

Rensis Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 140:1–55, 1932.