Identifying Critical Features for Formative Essay Feedback with Artificial Neural Networks and Backward Elimination

Mohsin Abbas[1,4(✉)], Peter van Rosmalen[2(✉)], and Marco Kalz[1,3(✉)]

[1] UNESCO Chair of Open Education, Faculty Management, Science and Technologies, Open University of the Netherlands, Heerlen, The Netherlands
{mohsin.abbas,marco.kalz}@ou.nl
[2] Faculty of Health, Medicine and Life Sciences, Maastricht University, Maastricht, The Netherlands
p.vanrosmalen@maastrichtuniversity.nl
[3] Heidelberg University of Education, Heidelberg, Germany
kalz@ph-heidelberg.de
[4] Faculty of Information Technology, University of Central Punjab, Lahore, Pakistan
mohsin.a@ucp.edu.pk

Abstract. For predicting and improving the quality of essays, text analytic metrics (surface, syntactic, morphological and semantic features) can be used to provide formative feedback to the students. In this study, the intent was to find a small number of features that exhibit a fair proxy of the scores given by the human raters. Using an existing corpus and a text analysis tool for the Dutch language, a large number of features were extracted. Artificial neural networks, Levenberg Marquardt algorithm and backward elimination were used to reduce the number of extracted features automatically. Irrelevant features were eliminated based on the inter-rater agreement between predicted and human scores calculated using Cohen's Kappa (κ). By using our algorithm, the number of features in this study was reduced from 457 to 23. The selected features were grouped into six different categories. Of these categories, we believe that the features present in the groups "Word Difficulty" and "Lexical Diversity" are most useful for providing automated formative feedback to the students. The approach presented in this research paper is the first step towards our ultimate goal of providing meaningful formative feedback to the students for enhancing their writing skills and capabilities.

Keywords: Formative feedback · Natural Language Processing · Neural Networks · Backward Elimination · Dimensionality reduction · Feature selection

M. Scheffel et al. (Eds.): EC-TEL 2019, LNCS 11722, pp. 396–408, 2019.
https://doi.org/10.1007/978-3-030-29736-7_30

1 Introduction

Providing meaningful formative feedback to students about the quality of their written assignments and texts is a time-consuming task [1,2]. Giving it immediately is sometimes not possible for teachers due to the large number of students [3] and the time required to grade an individual written assignment. Providing it automatically is possible using Natural Language Processing and Machine learning techniques [4–7]. Several systems have been implemented to provide feedback on essays.

Ellis Page, an English teacher proposed in the 1960s to use computers for assessments tasks [8]. PEG (Project Essay Grade) was his system that automatically graded essays. The scores given by PEG were comparable with the scores given by human judges with a correlation scores varying between 0.65 to 0.71. The focus of using PEG was to reduce the workload of the teachers which is one of the motivations of our work. The current version of PEG [9] provides automated essay scoring along with immediate feedback on texts through recommendations on how to improve the scores. IntelliMetric [10], another early AES system used artificial intelligence to score essays. IntelliMetric calculated more than 300 discourse, semantic and syntactic features to give a final score based on coherence, organization, elaboration, sentence structure and overall mechanics of the essay [11]. Educational Testing Services (ETS) uses E-rater [12] to automatically score GMAT essays. In order to provide scores, E-rater uses a huge corpus of graded responses to train its system. The first version of E-rater used approximately 50 features and with an agreement of 0.87 to 0.94 between the system and expert readers' scores on GMAT essay prompts [13]. In the newer version of E-rater (version 2.0), 12 more features were added with a kappa (κ) value of 0.58 [12]. Despite the existence of these systems, there is still a need to develop these types of feedback systems for languages other than English.

For the development of these questions, one of the critical questions is, which textual features are most important for automated feedback and how these features can be identified. The textual features (surface, syntactic, morphological and semantic features) that contribute the most in predicting the quality of students' texts can be extracted using machine learning techniques to provide formative feedback to the students. These metrics may be used to provide formative feedback to the students to improve their learning with an intent to calculate a small number of features that are required to provide meaningful feedback.

Several approaches for feature selection exist. In a study [14], an automatic linguistic and textual feature extraction tool Coh-Metrix [15] was used to select the features required to predict the essay quality; this selection was based on the highest values of Pearson correlation of features compared to scores given by human raters. Statistical techniques (discriminant analysis and stepwise regression) were used in a similar study [16] to select Coh-Metrix features significant in predicting the quality of high and low scoring essays. The feature classes related to lexical diversity, word frequency and syntactic complexity were reported to be the most predictive ones in determining the essay proficiency. Writing-Pal [17], an Intelligent Tutoring System, also uses features selected from Coh-Metrix

using statistical procedures [18]. Features were selected in another study [19] using Principal Component Analysis and the effectiveness of chosen features was analyzed for providing formative feedback to the writers. 211 features used in the study were extracted from 3 different tools: Coh-Metrix, Linguistic Inquiry and Word Count [20], and the Writing Assessment Tool [18]. Feature Selection techniques in text mining using deep learning have been reviewed in [21].

Several existing text analysis tools can calculate a huge number of textual features against input texts. ReaderBench [22] is an open source multilingual framework that makes use of natural language processing techniques to provide text analysis tools. The framework is multilingual [23] – text analysis tools are available in Dutch, French, Romanian and English. Readerbench provides more than 200 textual complexity indices related to linguistic features of the text including surface, syntactic, morphological, semantic, and discourse features. Using ReaderBench, a research to choose features that contribute the most towards the scores given by human raters has already been conducted for the French language [24]. That research uses a different approach, namely Discriminant Function Analysis. T-scan [25,26] is a Dutch language analysis tool that calculates more than 400 text features which can be used for lexical and syntactic analysis. Experiments in this research have been conducted using T-scan that heavily relies on the Alpino parser [27] while calculating its features.

The current study explores a data-driven approach to identify textual features and metrics for an essay feedback system for the Dutch language. Machine learning algorithms such as Neural Networks can be used to create models using a corpus of scored texts. In this study it was investigated whether features that may be used to provide formative feedback on essays written in Dutch can be identified using artificial neural networks and backward elimination. The analysis was done by calculating more than 400 features against a scored corpus of Dutch texts extracted using T-Scan. To understand and comprehend the meaning behind all these features is time-consuming task. These features are meant for technical experts, therefore, not all the features are useful in providing meaningful formative feedback to the students. In this study, as a first step, we reduce the number of features using machine learning techniques. This paper is divided into four sections - the algorithm used in the research is described in the following section. Next we present the outcomes and the findings of our experiment. Finally we discuss the significance of our findings and discuss limitations of the research and conclude implications for future research that can be conducted using our algorithm.

2 Methods

We regard Automatic Essay Scoring as a subfield of Natural Language Processing where the prediction of scores against input texts is done automatically. The input of these models are features that are calculated from the corpus. The features are used as an input and the scores given by the human raters are used as output of machine learning algorithms to create the learned models. These can then be used to predict the scores against unknown texts. The performance of

applications involving machine predicted scores is done by finding the inter-rater reliability between the predicted score and the scores given by human raters. For this purpose, a value of Cohen's Kappa (κ) [28] is calculated. This value lies between -1 to 1. A value less than zero means that there is no agreement between the predicted and the human scores. For the values of Cohen' Kappa (κ), the interpretation of inter-rater agreement is presented in Table 1.

Table 1. Inter-rater agreement for different values of Cohen's Kappa (κ).

Sr.no.	Kappa value (κ)	Inter-rater agreement
1	$\kappa \leq 0$	None
2	$0 < \kappa \leq 0.20$	Slight
3	$0.20 < \kappa \leq 0.40$	Fair
4	$0.40 < \kappa \leq 0.60$	Moderate
5	$0.60 < \kappa \leq 0.80$	Substantial
6	$0.80 < \kappa \leq 1$	Perfect

Existing research [9–13] focused on increasing the value of Kappa (κ) so that agreement between human raters and machine predicted scores is impeccable. In our research, the goal was to reduce the number of input features until the value of Kappa (κ) remains greater than zero. We used a corpus of scored Dutch texts and extracted different features from them using T-Scan. For our experiment, features extracted from Readerbench could have been used, however, we went for T-Scan since the number of features calculated by T-Scan is greater than the ones calculated by Readerbench. The input text features were used to train a machine learning model and an agreement between the scores given by the human raters and the predicted scores was found by calculating the value of Cohen's Kappa (κ). Then, the number of input features was reduced using Neural Networks Backward Elimination Technique [29,30]. This process (involving the training of the machine learning models and applying the Neural Networks Backward elimination technique) was repeated while the value of Kappa (κ) at the end of each feature elimination remained greater than zero.

2.1 Instruments

A corpus of scored texts was used to train a machine learning model to predict scores against texts. In this research, quality of Dutch texts is correlated with the scores obtained in these texts using the CLiPS Stylometry Investigation (CSI) [31]. This Dutch language corpus of scored texts was used to train models using a Neural Networks algorithm after extracting features from T-Scan. The corpus provides 517 essays of which 436 essays are graded. For each of the 436 scored Dutch essays, there exists a single score that lies between 0 to 20. The minimum score given of a text in this corpus is 5 and the maximum score is 18. A histogram of scores present in the corpus is shown in Fig. 1.

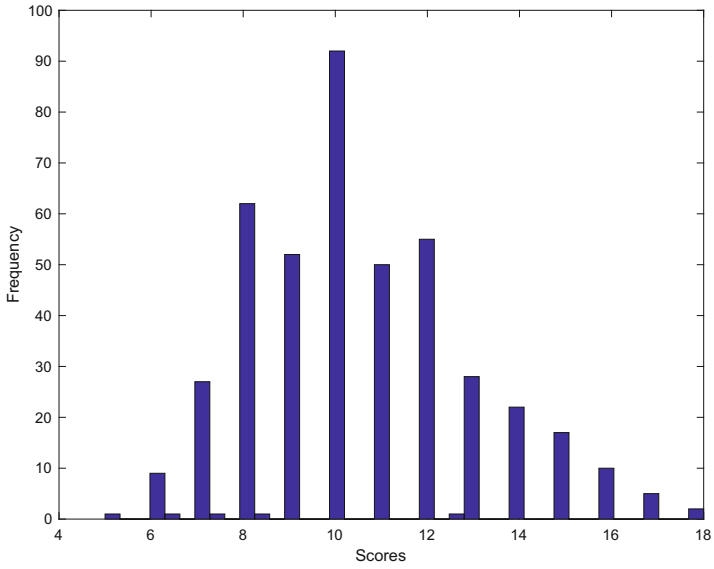

Fig. 1. Histogram of scores in the CLiPS corpus

T-Scan is an analysis tool for Dutch texts that provides text complexity features for input texts. This analysis tool was used to extract features from the texts present in the CLiPS corpus. For the texts, the number of features calculated by T-Scan is 457. However, not all these features can be shown to the students to provide formative feedback, therefore, the number of features was reduced. Textual features against each of the 436 texts were extracted using the T-scan online tool [32]. These extracted features were then used to train a neural networks prediction model to predict scores against unknown texts.

The neural networks algorithm used in our experiments was Levenberg-Marquardt algorithm [33–35]. The texts in the corpus were divided into two parts - one part for training and another one for testing the Neural Networks prediction model. MATLAB was used to create these models using the Levenberg Marquardt algorithm. For dimensionality reduction, the technique that was used was backward elimination. The Backward Elimination technique is a greedy algorithm that starts with n input features with a target to eliminate one out of these n features. In our research, for eliminating a single input feature, using backward elimination, n machine learning models were trained leaving each of the *n-1* features at a time. The models were created using Levenberg Marquardt algorithm and the value of kappa (κ) was calculated after leaving out each of the feature. After n models were trained, that feature was eliminated without which the value of kappa (κ) remained the maximum. The fact that the value of kappa (κ) stayed maximum was an indication that the inter-rater reliability between the human and predicted scores was still the best without the eliminated feature.

2.2 Procedure

For all of the 457 features extracted using T-Scan, one feature was eliminated at a time using Backward Elimination until the value of Kappa (κ) remained greater than zero. The procedure followed to achieve our goal is shown in Fig. 2 and is described below:

1. Extract features from Dutch texts using T-Scan
2. Start the experiment with all the 457 extracted features
3. Train the model using Neural Networks with chosen features
4. Test the model trained in Step 3 and calculate the value of kappa (κ)
5. If kappa (κ) is greater than zero, go to step 6, otherwise, go to step 7
6. Use the backward elimination technique to eliminate one feature and then repeat steps 3 to 5
7. Stop the experiment

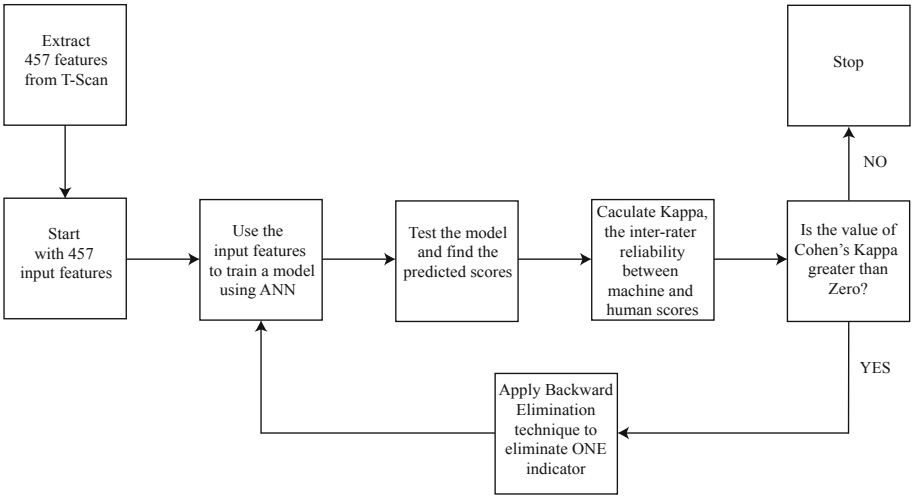

Fig. 2. The procedure followed to reduce the number of features

3 Results

The experiment was run on MATLAB R2017b on an iMac with MacOS version 10.14.4 having an Intel core i7, 4 GHz processor with 32 GB of RAM. The experiment ran for 13 days after which the stopping criteria was reached. The total neural network learning models trained during the experiment were 104,440. The value of Cohen's Kappa (κ) varied between 0.05 to 0.52. The variation in the value of Cohen's Kappa (κ) against different number of features is shown in Fig. 3. At the end of the experiment, we were left with 23 features; these features are given in Table 2. A brief description of each of the feature category is given below:

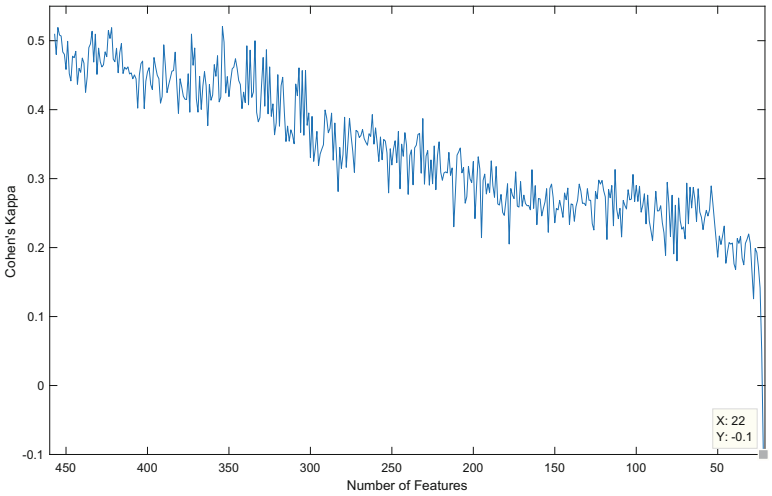

Fig. 3. The variation in the value of Cohen's Kappa (κ) against different number of features

Word Difficulty: The first seven features in Table 2 are related to the difficulty of words used in the texts. One of the features calculates the number of words per morpheme where a morpheme is a unit of the language that cannot be subdivided. Remaining features of this category compute the frequency of the words used in the texts. Four of these features quantify the proportion of:

1. the words that belong to most frequent 2000 words,
2. content words associated with the most frequent 1000 words,
3. nouns associated with the most frequent 20000 words,
4. words pertaining to the most frequent 1000 words.

The remaining two features related to word difficulty are the logarithm of frequency of words and the logarithm of frequency of nominal compositions. Nominal composition is the process of forming words that include lexemes that have more than one stem.

Sentence Complexity: There is only one feature in Table 2 associated with the sentence complexity. This feature provides the average of the number of words present in each sentence.

Lexical Diversity: Six features in the list of are related to lexical diversity and can be used to determine the richness of vocabulary used in a text. One of the features measures the lexical diversity of words and represent the uniqueness of words used in a text. Features such as the type token ratio (TTR) for words, density of content words and the number of arguments that occur in the previous sentence per sentence are also present in this category. TTR is defined as a ratio

Table 2. A description of the 23 reduced features after completion of the experiment.

Sr.no.	Feature name	Explanation	Category
1	Wrd_per_morf_zn	Words per morpheme, without names	Word Difficulty
2	Wrd_freq_log_sam_nw	Logarithm of word frequency of the nominal compositions	Word Difficulty
3	Wrd_freq_zn_log	Logarithm of word frequency without names	Word Difficulty
4	Freq2000	The word that belongs to the most frequent 2000 words	Word Difficulty
5	Freq1000_inhwrd	The proportion of content words associated with the most frequent 1000 words	Word Difficulty
6	Freq20000_nw	Proportion nouns associated with the most frequent 20000 words	Word Difficulty
7	Freq1000_corr	Corrected proportion of words pertaining to the most frequent 1000 words	Word Difficulty
8	Wrd_per_zin	Words per sentences	Sentence Complexity
9	MTLD_wrd	Measure of Lexical Diversity for words	Lexical Diversity
10	TTR_wrd	Type token ratio for words	Lexical Diversity
11	Inhwrd_d	Density of content words	Lexical Diversity
12	Arg_over_vzin_dz	Number of arguments that occur in the previous sentence per part sentence	Lexical Diversity
13	Tijd_d	Density of time words	Lexical Diversity
14	Tijd_MTLD	Measure of Lexical Diversity in text for time words	Lexical Diversity
15	Concr_ov_nw_p	Proportion of other specific nouns	Semantic Classes
16	Gedekte_nw_p	Proportion of nouns and names in the list	Semantic Classes
17	Alg_nw_p	Proportion of general nomina to all nomina	Semantic Classes
18	Ep_ev_bvnw_p	Proportion of nouns that evaluate epistemically	Semantic Classes
19	Conc_ww_p	Proportion of concrete verbs	Semantic Classes
20	Alg_ww_rel_sit_p	Proportion of general verbs around relationships between situations on all verbs	Semantic Classes
21	Spec_bijw_p	Proportion of specific adverbs to adverbs	Semantic Classes
22	Procesww_p	Proportion of process verbs to verbs	Verb Characteristics
23	Perplexiteit_bwd	Perplexity, backwards	Probability Measures

between the total number of unique words (type) to the total number of words (token) in a text [36]. Content words are the words in the texts that carry meaning. The remaining two features in this class are the density of time words and the measure of lexical diversity in text for time words.

Semantic Classes: Semantic features represent the meaning of lexical components in the text. There are seven features in this class of features. These features measure the proportion of:

1. specific nouns - these nouns specify a particular thing
2. nouns and names in the list (provided by T-Scan)
3. general nomina to all nomina
4. nouns that evaluate epistemically
5. concrete verbs - in the verbs of motion, these represent unidirectional aspect of the verb
6. general verbs around relationships between situations on all verbs
7. specific adverbs to adverbs

Verb Characteristics: One feature is related to the verb characteristics in the text. This feature delineates the proportion of process verbs to all the verbs used in the text.

Probability Measures: Lastly, a feature calculates the logarithm of the backward perplexity. In Natural Language Processing, "perplexity" is a way to evaluate the language model [37] and has an inverse relation with the probability. A lower value of perplexity refers to a higher value of probability.

4 Discussion and Conclusions

In this research, the goal was to reduce the number of features calculated against input texts written in Dutch language via a data-driven approach. The results of our research present the features for which there remained a slight agreement between machine predicted scores and human ratings by the end of our experiment. The number of features in this research was reduced from 457 to 23 by using a combination of machine learning and feature reduction technique. These 23 features were grouped into different categories based on their description given in the T-Scan documentation [25]. Of these features, we believe that the features present in the categories "Word Difficulty" and "Lexical Diversity" are most useful for providing automated formative feedback to the students. Informing the students immediately about the richness in the vocabulary, the fraction of words that carry meaning, the type token ratio, the proportion of words that belong to a specific set of words (such as words or content words associated with the most frequent 1000 words) or the frequency of certain words used in their text may help them in improving the quality of their writing.

The features present in the categories "Sentence Complexity", "Semantic Classes" and "Verb Characteristics" need to be explored further. The results obtained from these categories serve as a starting point for our future research where the experts of Dutch language will analyze if these features can be used to provide meaningful formative feedback. The only feature present in the category "Probability measures" that calculates the logarithm of the backward perplexity is too technical and may not be helpful in providing meaningful feedback to the students.

The results in our study are restrained by the corpus used in the experiments - there are a lot of texts in the corpus having an average score, however, the texts having a high score, or the ones having a low score are not sufficient. The machine learning algorithms therefore sometimes tend to overfit on those texts that have an average score. This problem can be solved by using such a corpus that includes texts having scores that are uniformly distributed. In these experiments, the corpus that was used had a normal distribution of scores given by the human raters. Secondly, the corpus used in this work does not have texts that belong to the same subject or topic. There could be certain features that correspond to higher values for certain domains and lower values for others - using a domain specific corpus may improve the results further. Lastly, the texts in the corpus used in our experiments have been written by people having different backgrounds, age groups and levels of education. The type of writing may have different features that distinguish the type of writer (such as their age, gender etc...). Conducting the experiment with texts written by people having same age group, same level of education and similar background also needs to be investigated.

In future, the same experiment can be repeated using machine learning algorithms other than neural networks, or, by using different neural network algorithms such as gradient descent [38] or quasi-Newton [39] methods to explore whether there is an improvement in results by using a different algorithm. Finally, applying the algorithm on features extracted from texts using a different tool such as ReaderBench may add to the existing set of our chosen features. The approach presented is in this research paper is the first step of the three-step approach. In the first step, dimensionality of the input features was reduced automatically - as presented in this paper. The future work will include feedback on the usefulness of these features by humans (teachers/experts) and then by students. The ultimate goal is to provide meaningful formative feedback to the learners for improving the quality of their texts.

References

1. Irons, A.: An Investigation into the Impact of Formative Feedback on the Student Learning Experience (2010)
2. Shute, V.J.: Focus on formative feedback. Rev. Educ. Res. **78**(1), 153–189 (2008). https://doi.org/10.3102/0034654307313795

3. Irons, A.: Enhancing Learning through Formative Assessment and Feedback. Routledge, Taylor and Francis, London (2007)
4. Mehmood, A., On, B.W., Lee, I., Choi, G.S.: Prognosis essay scoring and article relevancy using multi-text features and machine learning. Symmetry **9**(1), 1–16 (2017). https://doi.org/10.3390/sym9010011
5. Nguyen, H., Xiong, W., Litman, D.: Iterative design and classroom evaluation of automated formative feedback for improving peer feedback localization. Int. J. Artif. Intell. Educ. **27**(3), 582–622 (2017). https://doi.org/10.1007/s40593-016-0136-6
6. Ramachandran, L., Gehringer, E.F., Yadav, R.K.: Automated assessment of the quality of peer reviews using natural language processing techniques. Int. J. Artif. Intell. Educ. **27**(3), 534–581 (2017). https://doi.org/10.1007/s40593-016-0132-x
7. Taghipour, K., Ng, H.T.: A neural approach to automated essay scoring. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, Texas, pp. 1882–1891 (2016). https://doi.org/10.18653/v1/d16-1193
8. Page, E.B.: The imminence of... grading essays by computer. Phi Delta Kappa Int. **47**(5), 238–243 (1966)
9. PEG Writing. https://pegwriting.com. Accessed 4 Dec 2018
10. Rudner, L.M., Garcia, V., Welch, C.: An evaluation of the IntelliMetric essay scoring system. J. Technol. Learn. Assess. **4**(4), 1–22 (2006)
11. Shermis, M., Burstein, J.: Automated Essay Scoring: A Cross-Disciplinary Perspective (2003)
12. Attali, Y., Burstein, J.: Automated essay scoring with E-Rater®V.2.0. J. Technol. Learn. Assess. **4**(3), 1–21 (2006)
13. Burstein, J., Kukich, K., Wolff, S., Lu, C., Chodorow, M.: Computer analysis of essays. In: Proceedings of the NCME Symposium on Automated Scoring, pp. 1–13 (1998)
14. Crossley, S.A., Roscoe, R., McNamara, D.S.: Predicting human scores of essay quality using computational indices of linguistic and textual features. In: International Conference on Artificial Intelligence in Education (AIED 2011), pp. 438–440 (2011). https://doi.org/10.1007/978-3-642-21869-9_62
15. Graesser, A.C., McNamara, D.S., Louwerse, M.M., Cai, Z.: Coh-Metrix: analysis of text on cohesion and language. Behav. Res. Methods Instrum. Comput. **36**(2), 193–202 (2004). https://doi.org/10.3758/BF03195564
16. McNamara, D.S., Crossley, S.A., McCarthy, P.M.: Linguistic features of writing quality. Written Commun. **27**(1), 57–86 (2010). https://doi.org/10.1177/0741088309351547
17. Roscoe, R.D., Allen, L.K., Weston, J.L., Crossley, S.A., McNamara, D.S.: The writing pal intelligent tutoring system: usability testing and development. Comput. Compos. **34**, 39–59 (2014). https://doi.org/10.1016/j.compcom.2014.09.002
18. McNamara, D.S., Crossley, S.A., Roscoe, R.: Natural language processing in an intelligent writing strategy tutoring system. Behav. Res. Methods **45**(2), 499–515 (2013). https://doi.org/10.3758/s13428-012-0258-1
19. Crossley, S.A., Kyle, K., Mcnamara, D.S.: To aggregate or not? linguistic features in automatic essay scoring and feedback systems. J. Writ. Assess. **8**(1), 1–16 (2015)
20. LIWC - Linguistic Inquiry and Word Count. https://liwc.wpengine.com. Accessed 23 Mar 2019

21. Liang, H., Sun, X., Sun, Y., Gao, Y.: Text feature extraction based on deep learning: a review. EURASIP J. Wirel. Commun. Networking **2017**(1), 1–12 (2017). https://doi.org/10.1186/s13638-017-0993-1
22. Dascalu, M., Westera, W., Ruseti, S., Trausan-Matu, S., Kurvers, H.: *ReaderBench* learns dutch: building a comprehensive automated essay scoring system for Dutch language. In: André, E., Baker, R., Hu, X., Rodrigo, M.M.T., du Boulay, B. (eds.) AIED 2017. LNCS (LNAI), vol. 10331, pp. 52–63. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-61425-0_5
23. Dascalu, M., et al.: *ReaderBench*: a multi-lingual framework for analyzing text complexity. In: Lavoué, É., Drachsler, H., Verbert, K., Broisin, J., Pérez-Sanagustín, M. (eds.) EC-TEL 2017. LNCS, vol. 10474, pp. 495–499. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66610-5_48
24. Dascalu, M., Dessus, P., Thuez, L., Trausan-Matu, S.: How well do student nurses write case studies? a cohesion-centered textual complexity analysis. In: Lavoué, É., Drachsler, H., Verbert, K., Broisin, J., Pérez-Sanagustín, M. (eds.) EC-TEL 2017. LNCS, vol. 10474, pp. 43–53. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66610-5_4
25. Kraf, R., Pander Maat, H.: Leesbaarheidsonderzoek: oude problemen, nieuwe kansen. Tijdschrift Voor Taalbeheersing **31**(2), 97–123 (2014). https://doi.org/10.5117/tvt2009.2.lees356
26. Maat, H.P., et al.: T-Scan: a new tool for analyzing Dutch text. Comput. Linguist. Netherlands J. **4**, 53–74 (2014)
27. Bouma, G., van Noord, G., Malouf, R., Noord, G.V.: Alpino: wide-coverage computational analysis of Dutch. Lang. Comput. **37**, 45–59 (2000)
28. Viera, A.J., Garrett, J.M.: Understanding interobserver agreement: the kappa statistic. Fam. Med. **37**(5), 360–363 (2005)
29. Leray, P., Gallinari, P.: Feature selection with neural networks. Behaviormetrika **26**(1), 145–166 (1999)
30. Koller, D., Sahami, M.: Toward optimal feature selection. In: ICML 1996 Proceedings of the Thirteenth International Conference on International Conference on Machine Learning, Bari, pp. 284–292 (1996)
31. Verhoeven, B., Daelemans, W.: CLiPS Stylometry Investigation (CSI) corpus: a Dutch corpus for the detection of age, gender, personality, sentiment and deception in text. In: The 9th International Conference on Language Resources and Evaluation (LREC) (2014)
32. T-Scan Online Tool. https://webservices-lst.science.ru.nl/tscan/. Accessed 18 Nov 2018
33. Levenberg, K.: A method for the solution of certain non-linear problems in least squares. Q. Appl. Math. **2**, 164–168 (1944). https://doi.org/10.1090/qam/10666
34. Marquardt, D.W.: An algorithm for least-squares estimation of nonlinear parameters. J. Soc. Ind. Appl. Math. **11**(2), 431–441 (1963)
35. Hagan, M.T., Menhaj, M.B.: Training feedforward networks with the marquardt algorithm. IEEE Trans. Neural Networks **5**(6), 989–993 (1994)
36. Kettunen, K.: Can type-token ratio be used to show morphological complexity of languages? J. Quant. Linguist. **21**(3), 223–245 (2014). https://doi.org/10.1080/09296174.2014.911506
37. Chen, S.F., Beeferman, D., Rosenfeld, R.: Evaluation metrics for language models. In: Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop (1998)

38. Baldi, P.: Gradient descent learning algorithm overview: a general dynamical systems perspective. IEEE Trans. Neural Networks **6**(1), 182–195 (1995). https://doi.org/10.1109/72.363438
39. Robitaille, B., Marcos, B., Veillette, M., Payre, G.: Modified quasi-newton methods for training neural networks. Comput. Chem. Eng. **20**(9), 1133–1140 (1993). https://doi.org/10.1016/0098-1354(95)00228-6