A Posteriori Agreement as a Quality Measure for Readability Prediction Systems

Philip van Oosten[1,2], Véronique Hoste[1,2], and Dries Tanghe[1,2]

[1] LT[3] Language and Translation Technology Team, University College Ghent, Groot-Brittanniëlaan 45, 9000 Ghent, Belgium
[2] Ghent University, Krijgslaan 281, 9000 Ghent, Belgium

Abstract. All readability research is ultimately concerned with the research question whether it is possible for a prediction system to automatically determine the level of readability of an unseen text. A significant problem for such a system is that readability might depend in part on the reader. If different readers assess the readability of texts in fundamentally different ways, there is insufficient a priori agreement to justify the correctness of a readability prediction system based on the texts assessed by those readers. We built a data set of readability assessments by expert readers. We clustered the experts into groups with greater a priori agreement and then measured for each group whether classifiers trained only on data from this group exhibited a classification bias. As this was found to be the case, the classification mechanism cannot be unproblematically generalized to a different user group.

1 Introduction

In the most general terms, the goal of authoring a text is to get a message across to an intended audience. The readability of a text, then, can be defined as the relative ease of that audience to understand the author's message. It is intuitively clear that, even when defined in such general terms, the inherent subjectivity of the concept of readability cannot be ignored. The ease with which a given reader can correctly identify the message conveyed in a text is, among other things, inextricably related to the reader's background knowledge of the subject at hand [11].

The domain of readability research has at its primary research goal the design of a method to automatically predict the readability of a text. In recent years, a tendency seems to have arisen to explicitly address the subjective aspect of readability. [14] ultimately base their readability prediction method exclusively on the extent to which readers found a text to be "well-written". [10] take the assessments supplied by a number of experts as their gold standard, and test their readability prediction method as well as assessments by novices against these expert opinions. Similarly, [13] compile a gold standard for readability prediction by collecting assessments by expert and naive readers.

Subjective assessment entails the problem of reliably aggregating data that were obtained from various sources. This is a recurring issue in Natural Language

Processing, and is routinely caused by several contributors making different decisions regarding some manual annotation task. [2] give a good overview of the standard practice that has arisen within the NLP domain, viz. to calculate some measure of inter-annotator agreement. If this measure is high enough, the data are deemed acceptable to serve as a gold standard.

In readability research, however, this practice does not seem to have gained much ground. Given that many readability prediction methods (e.g. [6, 5, 17]) were developed before it became commonplace, it is not surprising that inter-annotator agreement played no great part in the development of those readability formulas. However, even recent publications such as [14] and [10] make no mention of the issue, and uncritically average out results collected from different readers. This should be done with great caution indeed: [1] claimed that if the data on which readability formulas are based were not aggregated on the school grade level but considered at the individual level, their predictive power would drop from around 80% to an estimated 10%.

We aim to determine whether a readability prediction system can be generalized to a broader audience, even when lacking a priori agreement measures. This is done by evaluating the accuracy of different readability systems on different groups of experts with a large a priori agreement. Poor performance would then imply that the annotation behaviour of the expert group deviates from the larger group of annotators, which leads to the conclusion that the readability system is not appropriate for the general public. To compose the groups of experts, we used a simple clustering technique, combining experts with similar annotations together. Classification accuracy is used to measure the deviations between an expert group and the rest, i.e. the concatenation of the other expert groups.

Instead of calculating inter-annotator agreement prior to training a readability prediction system, we verify whether the classification accuracies of systems trained on a single cluster and the concatenation of the other clusters differ for the same test set.

The remaining sections of this article contain details on how we composed our data set (section 2), a discussion of the issue of determining inter-annotator agreement in our data set and a proposed approach to locate generalization problems (section 3), experimental results (section 4) and conclusions and further work (section 5).

2 Annotation process and data set

2.1 Training corpus

Readability research is often concerned with the readability prediction of texts for relatively unaccomplished readers. The goal, then, is to identify reading material suited to the reading competence of a given individual [6, 17, 16, 18]. Training data for the readability prediction system can then be drawn from textbooks intended for different competence levels [16, 7]. However, since our system must be applicable to generic Dutch text, such educational material is insufficient, and we assembled a new training corpus.

We selected 105 texts from the Lassy corpus [12], which is a corpus annotated with lexical and syntactic features. From the selected texts, fragments of one or more paragraphs were used for readability assessment. The length of the fragments ranged from 81 to 306 tokens, resulting in a total amount of just under 17K assessed tokens. In order to develop a generically applicable system that can predict readability across text domains, we attempted to construct a cross-domain training corpus. Therefore, the texts in the corpus were selected manually from several sources, such as children's literature, Wikipedia, newspaper articles and technical reports. Each of the text fragments received on average 22 individual assessments, with a standard deviation of 9.12. As different annotators applied different scoring strategies, it is impossible to give an overall description of the way in which assessments were distributed in the range of possible values.

2.2 The Expert Readers annotation tool

The corpus was assessed for readability by a number of experts, who are professionally involved with the Dutch language. The experts used a password protected web application to assess the texts.

In the application, multiple texts can be placed underneath each other in a column, that visually represents an overview of the ratings an expert assigned during the current session and helps the annotators to build up a frame of reference against which to assess newly loaded texts.

An annotator can load texts and assign a score between 0 (easy) and 100 (difficult) to them. Previously assigned scores can be revised.

A batch of texts with accompanying scores can be sent to the database by pressing a button. The texts are then removed, except if the annotators indicated they wanted to keep them available, so as to maintain a frame of reference across batches. When a user submits the current assessments, all scores in the batch are logged.

Texts are provided to the annotators randomly, with equal probability of providing a text from each text type, and independent from which texts were previously provided. However, a text can never appear twice in the same batch.

Apart from the readability scores and the rankings in the batches, the experts can also enter comments on what makes each text more or less readable. That allows for qualitative analysis. We did not ask more detailed questions about certain aspects of readability, because we wanted to avoid influencing the text properties experts pay attention to. Neither did we inform the experts in any way how they should judge readability. Any presumption about which features are important readability indicators was thus avoided. We do not know which experts based their assessments on which text properties, and which relative weights they attributed to them. Yet our main interest is to design a system that is robust enough to model readability as generally as possible.

2.3 Data provided through the application

The assessments of the experts are stored in a database. For each expert, all the batches, containing texts and corresponding scores are available. A qualitative survey reveals that different experts sometimes employ a different scoring strategy. For example, some people only use scores that are multiples of 10, while others use the full range of possible scores. This is not a trivial observation: such a difference in score assignment compromises the possibility to use the scores directly for regression.

The batches can also be seen as rankings of texts. We further consider the *text pairs* that can be extracted from the batches. From each batch, we extract all pairs of texts that differ in score and for which at least two other texts are ranked between the pair. In this way, we can reasonably assume that the expert evaluated the lower-ranked text as more readable than the text with the higher score.

3 Detecting disagreement between annotators

In this article, we use one particular type of readability prediction system as a working example: a binary classifier which is able to predict which of two given texts is the more readable one. To construct such a readability prediction system, a possible approach would be to first determine inter-annotator agreement for each text pair. The text pairs for which reasonable agreement [2] is found can then serve as the basis for a gold standard, which can then be used to train a binary classifier. More generally, composing a gold standard prior to performing supervised learning experiments is the standard practice.

However, in the data set provided by our experts, not everyone has assessed all of the same texts, let alone text pairs. It is therefore not possible to determine the agreement for all text pairs with sufficient accuracy, prior to training a binary classifier: there are too many missing values. Not all annotators spent the same amount of time assessing texts and some assessed more texts per batch than others. Therefore, not all annotators contribute the same amount of text pairs and there is not always an overlap between the texts they have seen. We also want to be able to maximally employ minor contributions. Furthermore, we found disagreement concerning some text pairs, and we want to examine whether those disagreements are incidental or whether they betray a more fundamental controversy in readability assessment.

We can identify two possible causes for the disagreements: there is no clear difference in readability between the two texts in the text pair; or different experts have contrasting opinions on what factors constitute readability. Overcoming both issues would require more experiments and a qualitative analysis. Further in this article, we don't attempt to distinguish between these issues, but we perform a quantitative analysis to uncover their effects.

As explained above, a readability prediction system can be developed by merging all the text pairs into a gold standard and training a classifier. In order

to merge the training data with an acceptable degree of reliability, there should be sufficient agreement between different experts' assessments of the same text pairs. An estimate of the classification accuracy, for example through cross-validation, then indicates how well the trained system works. However, since inter-annotator agreement could be too low to speak of a gold standard, we also need to investigate in further detail to what extent the resulting system can be generalized. That means that apart from achieving a high classification accuracy, it is also important that the eventual system delivers results that are acceptable for all experts. To facilitate a priori agreement and to be able to check a posteriori whether no expert views were excluded, we created groups of experts who provided the most similar annotations.

3.1 Preparation of the data sets

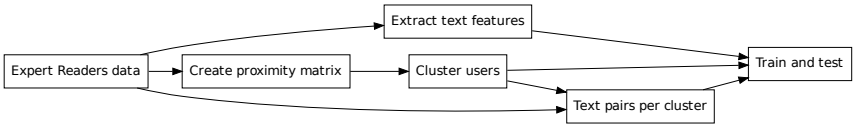

Fig. 1. Outline of how the data sets are composed from the expert assessments.

Figure 1 gives a schematic overview of how the data sets used for classification are prepared. Each block in the figure represents the execution of a set of commands. If an arrow points from one block to another, the former is executed before the latter and output from the former is passed as input to the latter.

Expert Readers data In this node, data are extracted from the Expert Readers Application database. Annotators who provided 25 text pairs or fewer are excluded.

Create proximity matrix For our experiments, we need groups of experts who have a shared view on readability. To divide the experts in those groups, we need a *proximity* measure: a metric to indicate to what extent the judgements of different experts are similar. The metric should allow us to distinguish experts who agree on how to order texts from those who disagree. Precision of the text pairs of one annotator with regard to the other meets this requirement.

In general, precision and recall are calculated by the following formulas: $P = \frac{TP}{TP+FP}$ and $R = \frac{TP}{TP+FN}$, where TP is the number of *true positives* (i.e. text pairs on which both annotators agree), and FP is the number of *false positives*

(i.e. text pairs on which the annotators disagree). *Negatives* with regard to a particular pair of experts would be the text pairs that only one of the two experts has reviewed. Since the annotation procedure does not require all annotators to see the same text pairs, no sensible distinction can be made between *true negatives* and *false negatives*. Therefore, the number of false negatives (FN) cannot be determined in this context, and we cannot calculate meaningful recall figures. The proximity between two experts is therefore the precision: the number of ordered text pairs that both annotators agree on, divided by the total number of text pairs that appear in the data sets of both annotators. The result of this block is a square symmetric matrix with proximity measures.

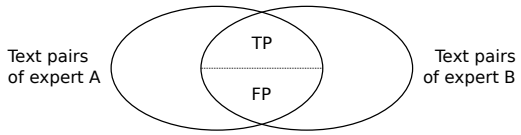

Fig. 2. True and false positives for the text pairs of two experts. Since the experts have not annotated all text pairs, there is no sensible notion of true and false negatives.

Cluster users Using the proximity degrees between all experts, it is possible to divide them into groups, so that the assessments of each expert correspond more to those of every other expert within the same group, than to those of other experts. We thereby make groups of experts with high a priori agreement. To create the groups, we use a simple agglomerative clustering algorithm [8]. Initially, a cluster is created for each individual expert. Subsequently, the two clusters with the highest degree of proximity are merged into a single one, until there is only one cluster left. The proximity between clusters is calculated as the minimal proximity between any of the members of each of the clusters. In this way, the agreement between all experts per cluster is maximized. The dendrogram in figure 3 shows the result of the clustering algorithm. Finally, in order to divide the experts into similar groups, we branch the dendrogram, keeping only the greatest possible clusters of experts among which the precision is higher than a given cut-off value. We experimented with different cut-off values, ranging from 0.5 to 0.9. If, for example, the precision is more than 0.5, there are at least as many text pairs about which each pair of annotators agree, as there are pairs about which they disagree.

Text pairs per cluster Given the set of experts in each of the clusters, their text pairs are merged into a single set. The set of text pairs for the cluster is simply the union of the text pairs of the annotators in the cluster.

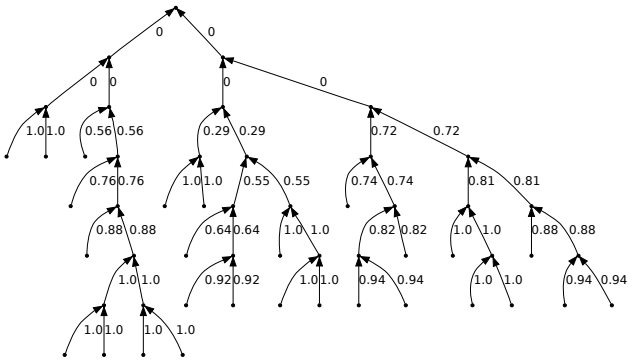

Fig. 3. A dendrogram showing the result of clustering the annotators. The edge labels show the precision between the child nodes. Each node represents a set of annotators. The leaf nodes represent the individual annotators.

Extract text features. For all the texts in the corpus, a number of features are extracted that can be used as training material for a classification algorithm. These are primarily indicators for lexical complexity, such as mean word length in number of characters [5, 17] and number of syllables [6], TF-IDF, log-likelihood and mutual information [9] computed against a large reference corpus [15], as well as character bigram and trigram frequencies. Additionally, some syntactic information is encoded in the form of proportions of different part of speech classes as tagged by the Tadpole parser [3].

3.2 Train and test

Class	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5	Feature 6
1	0.46	0.01	2.92	-0.01	0.2	0.03
-1	-0.46	-0.01	-2.92	0.01	-0.2	-0.03

Table 1. An example of two feature vectors for a text pair. The vectors are truncated.

Given the feature vector V_a for text T_a and the vector V_b for text T_b, we construct a single vector V_{ab} for the text pair T_{ab} by calculating the difference in feature values: the values from V_b are subtracted from the corresponding values from V_a. The class label 1 is assigned to V_{ab} if T_a is assessed as more readable than T_b, and label -1 is assigned if the opposite is true. Additionally, the vector V_{ba} is constructed with the reverse values and the reverse class label. Self-evidently,

such corresponding vectors are never distributed over the training and test data, as that would amount to contamination of the test data. A simplified example of a feature vector is shown in table 1. These feature vectors can serve as training data for a binary classifier, which can then be used to predict which of two texts is more readable than the other (see [18] for a similar procedure).

For each cluster, two data sets are generated. One set contains the feature vectors of the text pairs as assessed by the annotators in the cluster, and the other set contains those of the concatenation of the other clusters (the *complement*). The two data sets are then split up to perform 10-fold cross validation. An outline of the experiments per fold is shown in figure 4. The folds are created by splitting up the text sets (rather than the sets of text pairs) in 10 parts, since splitting only the text pair sets could result in contamination of the test sets. Text pairs of which at least one text is assigned to the test fold are added to the test set. The rest of the text pairs are added to the training set. This division of text pairs is done both for the cluster and for the complement.

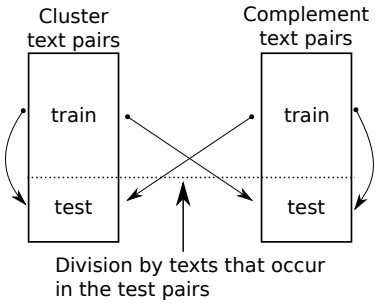

Fig. 4. Division in training data and test data per fold. A classifier is trained for both training sets and both classifiers are then tested on both test sets, so that the classification accuracy can be compared per test set.

To avoid that the amount of available data in either of the training sets might skew the classification results, we downsample the greater training set by randomly selecting an amount of text pairs that is equal to the amount of pairs present in the smaller training set.[3]

For each fold in each cluster, this results in two data sets that serve as training data for a binary classifier, and two test sets. We call the corresponding data sets the cluster training set, cluster test set, complement training set and complement test set. Both training sets are used to train a binary classifier [4]. We call a classifier trained on a cluster training set a *cluster classifier* and a

[3] In order to prevent that a particular downsampling of the training data might yield anomalous results, 10 random downsamplings have been performed and tested for each of the 10 test folds. We consider the mean classification accuracy over these 10 downsamplings within a fold as the classification accuracy of the test fold.

classifier trained on a complement training set a *complement classifier*. Both of these resulting classifiers are tested on both test sets to obtain the classification accuracy.

The goal of this experiment is to measure the influence of diverging annotation strategies on classification accuracy. If different annotation strategies have no influence on classification performance, both cluster and complement classifiers should perform equally well on cluster and complement test sets, or one of the classifiers should outperform the other for both test sets. If, however, each classifier performs better on the test set corresponding with its training set, that indicates a bias between training and test set, revealed by the combination of the feature set and the learning method that is used. If such a bias is found, the generalization ability of the learning method with the given feature set is questionable.

4 Results

Classification accuracy is given by the formula $CA = \frac{TP+TN}{TP+TN+FP+FN}$, where TP, TN, FP and FN are the number of true positives, true negatives, false positives and false negatives, respectively. In our experiments, we always observed $TP = TN$ and $FP = FN$, which is the result of the symmetric construction of the training and test data.

Table 2 gives an overview of our results.[4] The second to fourth column of the subtables show the average over the folds and subsamplings of the classification accuracies for training and testing on the data sets indicated in the header rows. Accuracies are only comparable when the same test set is used, so the second and third columns can be compared to each other and the two last columns are comparable.

The results can be interpreted as follows. If a classifier is generalizable, that implies that the test sets are not biased towards the classifier trained on the corresponding training set. The upshot of this is that the classification accuracies should indicate that either the cluster classifier or the complement classifier performs better on both test sets.

To clarify, we consider cluster 5 at cut-off level 0.8. Here, we see that the complement classifier performs better on both test sets than the classifier trained on cluster 5 itself. When testing on the cluster test set, we observe a higher classification accuracy for the complement classifier: 0.72 versus 0.64 for the cluster classifier. Similarly, when testing on the complement test set, the complement classifier achieves higher accuracy than the cluster classifier: 0.64 versus 0.58. When taking only the results for cluster 5 at cut-off level 0.8 into account, then, it would be plausible that the classification results can be generalized.

[4] We also computed results at cut-off level 0.9, but since too many individual annotators appear as expert groups, the cluster test sets often became too small to calculate meaningful results (cfr. figure 3). Therefore, only results at cut-off levels 0.5 to 0.8 are shown.

Train / Test	Cluster / Cluster	Compl. / Cluster	Compl. / Compl.	Cluster / Compl.
1	**0.75**	0.63	**0.68**	0.52
2	**0.78**	0.75	**0.67**	0.63
3	**0.68**	0.63	**0.70**	0.62
4	**0.71**	0.67	**0.68**	0.61
5	**0.68**	0.65	**0.72**	0.61
Mean	**0.72**	0.66	**0.69**	0.60

(a) Cut-off 0.5

Train / Test	Cluster / Cluster	Compl. / Cluster	Compl. / Compl.	Cluster / Compl.
1	**0.71**	0.62	**0.68**	0.59
2	0.60	**0.70**	**0.64**	0.58
3	0.77	**0.78**	**0.66**	0.62
4	**0.68**	0.65	**0.72**	0.60
5	**0.77**	0.63	**0.68**	0.53
6	**0.68**	0.63	**0.70**	0.62
7	**0.74**	0.74	**0.67**	0.61
Mean	**0.71**	0.68	**0.68**	0.59

(b) Cut-off 0.6

Train / Test	Cluster / Cluster	Compl. / Cluster	Compl. / Compl.	Cluster / Compl.
1	**0.71**	0.64	**0.68**	0.59
2	0.80	**0.84**	**0.67**	0.64
3	0.61	**0.75**	**0.64**	0.58
4	**0.66**	0.61	**0.70**	0.59
5	0.77	**0.81**	**0.66**	0.63
6	**0.79**	0.63	**0.69**	0.52
7	**0.68**	0.65	**0.72**	0.61
8	**0.73**	0.72	**0.67**	0.61
Mean	**0.72**	0.71	**0.68**	0.60

(c) Cut-off 0.7

Train / Test	Cluster / Cluster	Compl. / Cluster	Compl. / Compl.	Cluster / Compl.
1	0.69	**0.70**	**0.68**	0.61
2	**0.71**	0.65	**0.68**	0.61
3	**0.79**	0.72	**0.67**	0.59
4	0.81	**0.83**	**0.67**	0.64
5	0.64	**0.72**	**0.64**	0.58
6	**0.71**	0.68	**0.67**	0.56
7	**0.68**	0.61	**0.71**	0.59
8	0.76	**0.80**	**0.67**	0.62
9	0.71	**0.72**	**0.68**	0.61
10	0.74	**0.79**	**0.66**	0.61
11	**0.78**	0.63	**0.69**	0.52
Mean	**0.73**	0.71	**0.67**	0.59

(d) Cut-off 0.8

Table 2. Classification accuracy for each cluster and complement, at different cut-off levels. The average of the cluster averages is given in the last row. The greater of each pair of comparable accuracies is shown in bold.

However, for 5 out of 11 clusters at cut-off level 0.8, the situation is more problematic. When we consider cluster 2 at cut-off level 0.8, we observe a different situation: each classifier achieves higher accuracy on the test set corresponding with its own training set. The cluster classifier performs better on the cluster test set (0.71 versus 0.65), while the complement classifier performs better on the complement test set (0.68 versus 0.61). This indicates a bias in the classifiers to the test set corresponding with their own training set, which compromises the generalizability. We observe the same situation for a further 4 clusters out of 11 at cut-off level 0.8, and at cut-off level 0.5, the bias even manifests itself for all clusters.

We consider the average of the classification accuracies over all clusters as the criterion to decide whether a posteriori agreement is sufficient to call the results generalizable. If the mean cluster classification accuracy is higher for the

cluster test set and the mean complement classification accuracy higher for the complement test set, a posteriori agreement is insufficient. It then seems that in general, classifiers expose a bias towards the test set corresponding to the training set the classifier was trained on. For our experiments, that observation holds for all cut-off levels, as can be seen in the last row of the subtables of table 2. As a consequence, a posteriori agreement is insufficient to call a classifier as outlined in this article generalizable to a broader audience.

Although the accuracies on different test sets are incomparable, it seems that the complement classifier consistently performs better on the cluster test set than the cluster classifier on the complement test set. That may indicate that the complement classifier generally has a stronger prediction ability, even after subsampling. Further research is required to verify that hypothesis.

It seems that an increased cut-off level results in more clusters for which the complement classifier performs better on the cluster test set than the cluster classifier. With cut-off 0.5, this is nowhere the case, for 0.6 for 2 clusters, 3 clusters for 0.7 and and 6 for cut-off 0.8. Due to a redivision in folds per cut-off level, the classification accuracies are incomparable across levels. However, future work will establish whether this trend generally holds.

5 Conclusions and further work

NLP-problems customarily require some sort of inter-annotator agreement to be determined prior to performing classification experiments. The degree of agreement can then be seen as a quality measure for a data set. However, in a domain that is as potentially sensitive to annotator bias as readability, standard inter-annotator agreement statistics seem inadequate, as it is not unproblematic to simply average out the available data. Furthermore, in a data set consisting of a large number of sources supplying only a partial assessment of the data, such agreement measures quickly become more or less meaningless due to the relative sparsity of overlapping data points. To overcome these issues, we have developed a method to determine the generalizability of the classification method *after* training and testing. Determining a posteriori agreement is useful for data sets with low a priori agreement or when determining a priori agreement is problematic.

For the learning method and data set used in this article, we found insufficient a posteriori agreement, so further analysis is needed in order to determine whether a way to find consensus among experts is crucial, or whether a different combination of learning methods and feature sets must be used.

Future work includes further development of readability prediction systems and methodologies. We will extend the feature set used to predict readability and perform experiments with a range of classification and regression methods. We will also further extend our data set by collecting more assessments from experts, and by adding new texts to our corpus. We will use the method outlined in this article to assess the quality of the newly collected data, as well as the overall accuracy. Apart from the difference in classification accuracy, we will look

into other informative measures to determine the generalizability of readability prediction systems.

Acknowledgements

This research was funded by the University College Ghent Research Fund.

References

1. Anderson, R.C., Davison, A.: Conceptual and Empirical Bases of Readability Formulas. Tech. Rep. 392, University of Illinois at Urbana-Champaign (October 1986)
2. Beigman Klebanov, B., Beigman, E.: From Annotator Agreement to Noise Models. Computational Linguistics 35(4), 495–503 (2009)
3. van den Bosch, A., Busser, B., Canisius, S., Daelemans, W.: An efficient memory-based morphosyntactic tagger and parser for dutch. In: van Eynde, F., Dirix, P., Schuurman, I., Vandeghinste, V. (eds.) Proceedings of CLIN17. pp. 99–114 (2007)
4. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
5. Coleman, M., Liau, T.L.: A computer readability formula designed for machine scoring. Journal of Applied Psychology 60, 283–284 (1975)
6. Flesch, R.: A new readability yardstick. Journal of Applied Psychology 32(3), 221–233 (June 1948)
7. Heilman, M.J., Collins-Thompson, K., Callan, J., Eskenazi, M.: Combining lexical and grammatical features to improve readability measures for first and second language texts. In: Proceedings of HLT (2007)
8. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Comput. Surv. 31(3), 264–323 (1999)
9. Jurafsky, D., Martin, J.H.: Speech and Language Processing. Prentice Hall (2008)
10. Kate, R.J., Luo, X., Patwardhan, S., Franz, M., Florian, R., Mooney, R.J., Roukos, S., Welty, C.: Learning to Predict Readability using Diverse Linguistic Features. In: Proceedings of Coling23 (2010)
11. McNamara, D.S., Kintsch, E., Songer, N.B., Kintsch, W.: Are good texts always better? Interactions of text coherence, background knowledge, and levels of understanding in learning from text. Tech. rep., University of Colorado (1993)
12. van Noord, G.J.: Large Scale Syntactic Annotation of written Dutch (LASSY) (January 2009), `http://www.let.rug.nl/vannoord/Lassy/`
13. van Oosten, P., Tanghe, D., Hoste, V.: Towards an Improved Methodology for Automated Readability Prediction. In: Proceedings of LREC7 (2010)
14. Pitler, E., Nenkova, A.: Revisiting readability: A unified framework for predicting text quality. In: EMNLP. pp. 186–195. ACL (2008)
15. Schuurman, I., Hoste, V., Monachesi, P.: Cultivating Trees: Adding Several Semantic Layers to the Lassy Treebank in SoNaR. In: Proceedings of TLT7. Groningen, The Netherlands (2009)
16. Schwarm, S.E., Ostendorf, M.: Reading level assessment using support vector machines and statistical language models. In: Proceedings of ACL43. pp. 523–530. Association of Computational Linguistics, Ann Arbor (June 2005)
17. Staphorsius, G.: Leesbaarheid en leesvaardigheid. De ontwikkeling van een domeingericht meetinstrument. Cito, Arnhem (1994)
18. Tanaka-Ishii, K., Tezuka, S., Terada, H.: Sorting texts by readability. Computational Linguistics 36(2), 203–227 (2010)