Title

Neural Language Model-based Readability Assessment of Computer Science Introductory Texts for English-as-a-Second Language Learners

Permalink

https://escholarship.org/uc/item/2kt5p1bz

Journal

Proceedings of the Annual Meeting of the Cognitive Science Society, 44(44)

Author

Ehara, Yo

Publication Date

2022

Peer reviewed

Neural Language Model-based Readability Assessment of Computer Science Introductory Texts for English-as-a-Second Language Learners

Yo Ehara (ehara@u-gakugei.ac.jp)
Tokyo Gakugei University, 4-1-1 Nukuikita-machi
Koganei, Tokyo, 1848501 Japan

Abstract

English is the dominant language in computer science. In addition to English-based academic papers, English is frequently the only language provided in introduction sections and manuals of command and software libraries, which are essential aspects of computer programming. Hence, English-as-a-second-language (ESL) learners may have difficulty studying computer science because they must learn this field while also learning English. Despite this problem, few studies have assessed the difficulty level of computer science texts for ESL learners. Ideally, the difficulty levels of texts are assessed by having groups of ESL learners read them. However, owing to the excessive time and financial costs involved, such practices can be impractical. Hence, using two highly accurate automatic readability assessors based on natural language processing (NLP) techniques, we assessed the readability of various computer-science-related texts for ESL learners. The first assessor is based on state-of-the-art deep transfer learning, and the second is based on classical machine learning and applied linguistics. For training the assessors, we used a standard corpus employed in NLP, which was annotated by professional English teachers to evaluate the readability of the texts for ESL learners. To conduct the experiments, we built a collection of computer science texts ranging from academic papers to software manuals (READMEs) crawled from a source-code hosting website, namely GitHub. The experimental results showed that intermediate ESL learners were able to read most of the computer science related texts.

Keywords: Readability, Neural Language Model, Assessment

Introduction

English is the language used by most computer science (CS) publications, and is a second language for many scientists and those learning about science. Hence, the readability of scientific publications for English-as-a-second-language (ESL) learners is essential for determining and developing the support needed by such learners in terms of their science, technology, engineering, and mathematics (STEM) skills.

The language gap between native speakers and ESL learners may cause certain misunderstandings in the learners' interpretation of CS papers, significantly hindering the development of this field. However, only a few studies have investigated this issue.

To this end, in this paper, we assess the readability of CS publications for ESL learners. The readability (as an English text) of the main body of a paper can be excessively technical, making a proper evaluation difficult, even for human readers. Instead, we targeted the readability of the title and abstract, which are typically used to determine whether the main body of the paper should be read. To avoid biasing our analysis toward one particular field, we obtained texts from the databases of two different fields, and using natural language processing, we obtained the text in documents from the top-10 projects posted on GitHub and the 27,686 abstracts taken from the ACL Anthology and used them for our analyses. Because a large-scale manual readability assessment is impractical owing to financial and time constraints, we constructed two contrastive automatic readability assessors: one near state-of-the-art assessor with low interpretability and one vocabulary-based assessor with high interpretability.

To this end, we first sought to understand the difficulty that CS poses to non-native speakers of English. This problem was approached in two ways. The first is based on the field of educational NLP (Vajjala & Lučić, 2018). Using a standard corpus, we constructed a machine learning classifier for determining the difficulty of a text with high accuracy using deep learning methods, including bidirectional encoder representations from transformers (BERT) (Devlin, Chang, Lee, & Toutanova, 2019).

The second approach was conducting readability assessments based on information regarding the vocabulary of English learners. Such methods have been thoroughly studied in the field of applied linguistics, and numerous studies have shown that English learners need to know more than 95% of the words in a document to properly read and understand the text (Nation, 2006; Laufer & Ravenhorst-Kalovski, 2010). Assessing the readability of a text using the vocabulary level of the individual learner is beneficial for interpreting the results of a readability assessment. We therefore also constructed a classifier that ascertains the number of words in a text known by an English learner using a dataset of vocabulary tests for such learners (Ehara, 2018).

In experiments on evaluating the readability (Vajjala & Lučić, 2018) in educational NLP, which were carried out on a standard dataset, the two approaches used to assess the results of the readability were in close agreement.

The contributions of this study are as follows.

1. To determine the extent to which the language gap affects learning in computer science, we proposed an investigation method using automatic readability assessors.

2. Two methods were proposed for the readability assessors: a BERT-based assessor and a classical-machine-learning-

In J. Culbertson, A. Perfors, H. Rabagliati & V. Ramenzoni (Eds.), *Proceedings of the 44th Annual Conference of the Cognitive Science Society*.

based assessor that leverages findings from applied linguistics on vocabulary tests of ESL learners.

3. The BERT-based assessor confirmed that most of the README.md is readable by intermediate English learners.

4. The results of both methods show that CS academic texts are more difficult than software manuals.

Formalizing Readability Assessment Tasks

In this section, we formalize an automatic readability assessment based on (Ehara, 2021). The set of texts is denoted by $\{\mathcal{T}_i | i \in \{1, \ldots, N\}\}$, where N indicates the number of texts to be considered. Under a supervised machine learning setting, the automatic assessor uses the annotated readability labels for each text during the training phase. Typically, such annotation labels take discrete values such as the Likert scales rather than continuous one. The set of labels used for readability is denoted as $\mathcal{Y}$. For example, in the OneStopEnglish dataset (Vajjala & Lučić, 2018), 0 is considered elementary, 1 is intermediate, and 2 is advanced. Hence, in summary, $\mathcal{Y} = \{0, 1, 2\}$. In his way, the labels are usually ordered, and larger labels indicate that a text is more difficult to read. The number of levels depends on the evaluation corpus. Let y_i be the readability of text $\mathcal{T}_i$. Using $\mathcal{Y}$, we can express $y_i \in \mathcal{Y}$.

Given each text $\mathcal{T}_i$, we consider the problem of predicting the readability score s_i. The properties of s_i differ under supervised and unsupervised settings. Under a supervised setting, from the labels in the supervised data, the assessor knows the number of levels in which y_i can take. In other words, the assessor knows $|\mathcal{Y}|$. Therefore, the value of s_i output by the assessor is within the range of $\mathcal{Y}$, and thus $s_i \in \mathcal{Y}$. The prediction can then be simply measured based on the accuracy, which is the ratio occurring when $s_i = y_i$.

By contrast, under an unsupervised setting, the assessor does not even know the number of labels $|\mathcal{Y}|$ because no supervised data are available. Therefore, even if $\mathcal{Y}$ is a finite set, s_i is outputted as a simple real number representing the readability of the text. Of course, even if the readability represented by s_i is relatively reasonable compared to the other texts, $y_i = s_i$ usually does not hold. Therefore, rank correlation coefficients are preferable to be used to evaluate the performance.

Under both supervised and unsupervised settings, regardless of whether the ranges of s_is and y_is are identical, the goal can be defined in a uniform manner. Given N texts $[\mathcal{T}_i | i \in \{1, \ldots, N\}]$, the goal is to build an assessor that can output an array of readability scores $[s_i | i \in \{1, \ldots, N\}]$ that correlate well with the array of labels $[y_i | i \in \{1, \ldots, N\}]$. To denote the arrays, we use [and] throughout this study.

Here, although there are several types of correlation coefficients between the array of scores and the array of labels, according to (Ehara, 2021), a rank correlation coefficient, such as Spearman's ρ, is suitable because our main focus is on whether the order of readability of the evaluation corpus is correct. By contrast, the typical correlation coefficient, namely Pearson's ρ, depends heavily on the actual score of s_i, which is particularly problematic under an unsupervised setting.

```
15. deficit:
The company <had a large deficit>.
a: spent a lot more money than it earned
b: went down a lot in value
c: had a plan for its spending
                that used a lot of money
d: had a lot of money stored in the bank
```

Figure 1: Examples of the Vocabulary Size Test. They are asked to choose the option that paraphrases the part between "<" and ">" from a, b, c, and d.

Vocabulary Testing and Readability

A text is a sequence of words, and our goal was to measure the difficulty of text $\mathcal{T}_i$. If we can objectively measure the difficulty of the words in a text for ESL learners, such as the average difficulty, it would be possible to construct a readability measure using word difficulty values, such as using the average difficulty of words used in the text.

We therefore considered an approach to objectively evaluate and obtain the difficulty of words for ESL learners from a test for them. For this purpose, we use questions from the vocabulary size test, a widely used vocabulary test in applied linguistics (Beglar & Nation, 2007), as illustrated in Figure 1. Each question is multiple-choice, in which the test taker selects the option that most closely matches the meaning of the word in the sentence. The test has 100 questions, such as the one in Figure 1, ordered from easiest to hardest, and usually takes 30 to 40 minutes to complete.

To ensure the reproducibility, a publicly available dataset was preferred. Hence, in this study, we used the dataset by (Ehara, 2018). To develop this dataset, 100 ESL learners were tested using the vocabulary size test, and their responses were collected. This dataset was used to construct the assessor.

How can the results of a vocabulary test be analyzed to obtain word difficulty values that represent the language knowledge of the learners? For this purpose, we employed the item response theory, which is a statistical model that allows us to estimate the ability of a learner and the difficulty of a test question based on the learner's responses to questions, as summarized in (Baker, 2004).

We denote $\mathcal{V}$ to indicate a set of vocabulary words and $\mathcal{L}$ to indicate a set of learners. We write $z_{v,l}$ as a variable denoting whether the learner answered the question correctly. If learner l answered correctly to word v, $z_{l,v} = 1$; otherwise, $z_{l,v} = 0$. If the answer is correct, l is assumed to know word v.

We then trained the following model using $\{z_{v,l}\}$ as the training data:

$$p(z = 1 | v, l) = \text{sigmoid}(a_l - d_v), \qquad (1)$$

where a_l is the ability parameter of learner l, and d_v is the difficulty of word w. In addition, sigmoid denotes the sigmoid function, which is defined as $\text{sigmoid}(x) = \frac{1}{1+\exp(-x)}$.

The sigmoid function is a simplification of the softmax function, which is commonly used in neural classifiers. This is a monotonically increasing function within the range $(0,1)$. When the ability a_l of the learner is greater than the word difficulty d_v, the following property holds for the probability $p(z = 1|v,l)$: learner l knows the word v if and only if $p(z = 1|v,l) > \frac{1}{2}$. In this manner, Equation 1 compares the ability of the learner and the word difficulty in the same dimension.

To estimate the learner ability and word difficulty parameters, $z_{v,l}$ is given as z in Equation 1 during training. Thus, the item response theory estimates these parameters from the test result data such that the learner ability and word difficulty can be compared.

In Equation 1, d_v represents the word difficulty estimated from the vocabulary test. In addition to the difficulty of words in the vocabulary test, we wanted to obtain the difficulty of all words that may appear in the target language. To do so, we computed d_v from the word frequencies in a large balanced corpus as follows:

$$d_v = -\sum_{k=1}^{K} w_k \log(\text{freq}_k(v) + 1), \tag{2}$$

where K is the number of corpora used. Here, $\text{freq}_k(v)$ is the frequency of word v in the k-th corpus, and w_k is the weight parameter of the k-th corpus. In summary, given the lexical test result $\{z_{v,l}\}$ and corpus frequency feature $\text{freq}_k(v)$, we can estimate the weight w_k of the k-th corpus, as well as the parameter a_l of learner l's ability. To implement this model, we followed (Ehara, 2018) and used a logistic regression. Note that this model is unsupervised because it does not use valuable readability labels y_i during the training phase.

The process described thus far describes how the parameter estimation is conducted. After the parameter estimation, it is necessary to convert this to the text readability of $\mathcal{T}_i$ using the obtained word difficulty parameter d_v. Here, Equation 1 calculates the probability that learner j understands word i. By contrast, the learner is unspecified in the readability formula. We bridge this gap by computing the probability values for a learner with average ability; that is, j is set to the learner whose a_l is closest to the mean of all $\{a_l\}$s. We call this learner, l_{avg}. Finally, we simply used the sum of all words in the given text $\mathcal{T}_\rangle$ as the readability score s_i.

$$s_i = score(\mathcal{T}_i) = -\log\left(\prod_{v\in\mathcal{T}_i} p(z = 1|v, l_{\text{avg}})\right) \tag{3}$$

Experimental Settings

Readability Dataset and Experiments

The basis of this study was the construction of highly accurate readability assessors. Therefore, we conducted an experiment to confirm whether the created assessors were highly accurate based on (Ehara, 2021).

The OneStopEnglish dataset (Vajjala & Lučić, 2018) was used as the source of readability for ESL learners. This dataset is publicly available, and thus the results of the following studies can be reproduced. Because it was designed to address the issues from previous studies, according to (Vajjala & Rama, 2018), another rationale for choosing this dataset is its reliability. For example, if texts with an easy label are shorter than texts with a difficult label, the labels can be predicted without using the content of the texts and merely using the length.

The Guardian newspaper was the source of the original articles. In this dataset, texts were annotated with three labels by professional English teachers teaching ESL learners, i.e., elementary, intermediate, and advanced. The dataset was designed as a parallel corpus such that the readability labels cannot be inferred from the topics discussed, rather than the difficulty of the text. Being a parallel corpus means that language teachers manually rewrote the original articles into the three aforementioned readability levels, rather than simply annotating the labels.

All three levels had 189 texts, with 567 texts in total. We split these texts into a *training set* consisting of 339 texts, a *validation* set consisting of 114 texts, and a *test* set consisting of 114 texts. The *training* and *validation* sets were used to train the supervised methods for comparison. Unsupervised methods do not use the training and validation sets and use only the test set.

Compared Methods

Because BERT-based sequence classification has been reported to achieve excellent results (Devlin et al., 2019), we applied the standard BERT-based sequence classification approach involving pretraining and fine-tuning. For the pretrained model, we used several models taken from the HuggingFace models [1]. Each HuggingFace pre-trained model is named **bert-large-cased-whole-word-masking**. These names include basic information regarding the model size (base/large), whether the model is case-sensitive or not (cased/uncased), and whether a strategy called "whole-word-masking" was applied during training. We named our supervised models based on the pre-training models they utilized. The model using **bert-large-cased-whole-word-masking** was named **BERTlcw**.

The same fine-tuning was conducted for all models using 339 training texts. For fine tuning, we used the Adam optimizer (Kingma & Ba, 2015) with a setting of 10 epochs and a 0.00001 training rate.

To implement conventional readability formulae, we used the **readability** PyPI package [2]. We used almost all readability formulae implemented in this package for our experiments, namely, **Flesch-Kincaid** (Flesch-Kincaid Grade

[1] https://huggingface.co/models

[2] https://pypi.org/project/readability/

Table 1: Predictive Performance of Readability. Only **BERTlcw** is supervised, and the others are unsupervised.

Method	Spearman's ρ	Pearson's ρ
Flesch-Kincaid	0.324	0.359
ARI	0.317	0.351
Coleman-Liau	0.373	0.372
FleschReadingEase	-0.387	-0.426
GunningFogIndex	0.331	0.362
LIX	0.348	0.383
SMOGIndex	0.456	0.479
RIX	0.437	0.462
DaleChallIndex	0.495	0.506
TCN RSRS-simple	-	0.615(*)
Vocabulary-based	**0.730**	**0.715**
BERTlcw	0.866	0.864

Level, FKGL) (Kincaid, Fishburne Jr, Rogers, & Chissom, 1975), **ARI** (Automated Readability Index) (Senter & Smith, 1967), the **Coleman-Liau** Index (Coleman & Liau, 1975), **Flesch Reading Ease** (Flesch, 1948), **Gunning Fog Index** (Gunning, 1952), **LIX** (Björnsson, 1968), **SMOG Index** (Mc Laughlin, 1969), the RIX index (Anderson, 1983), and **Dale-Chall Index** (Dale & Chall, 1948). Further details of these formulae and their implementations are described on the project page. All of these readability formulae are *unsupervised* in the sense that they do not require any training data.

To build **Vocabulary-based**, we trained the model using (Ehara, 2018), which is a publicly available dataset. The training of **Vocaublary-based** requires corpus word frequencies to roughly capture the difficulty of the words. To this end, we used a combination of the two corpora. The first is the British National Corpus (BNC Consortium, 2007), which is a balanced corpus of British English. The second is the Corpus of Contemporary American English (COCA) (Davies, 2008), a balanced corpus of American English. The use of this combination was also hinted at by the findings from applied linguistics, which used a combination of these corpora to make wordlists suitable for beginner ESL learners and other educational resources important for English learning (Nation, 2006). Although the stemming of British and American English may differ, we employed stemming similar to the *word family* stemming used in applied linguistics. This simply ignores the word forms, e.g., "playing" is counted as "play."

Experimental Results

The experimental results are presented in Table Table 1. Importantly, it can be seen that both our supervised learning method and the vocabulary-based methods show higher rank correlation coefficients than the existing approaches. Thus, the proposed method is sufficiently accurate. In Table 1, **TCN RSRS-simple** is reported to be the best (Martinc, Pollak, & Robnik-Šikonja, 2021). Because the test set data applied in (Martinc et al., 2021) were unavailable, we used (*) to show that a direct comparison is difficult to achieve. In addition, while the authors provided the value of the Pearson's correlation coefficients for **TCN RSRS-simple**, they did not present the rank correlation coefficient values that we used; therefore, we denote this by "-" for the other fields of **TCN RSRS-simple**. Interestingly, although **BERTlcw** and **Vocabulary-based** use quite different approaches, both achieve a high accuracy.

Experiments with CS Texts

Datasets for Academic Texts

To analyze the CS texts, we retrieved freely available CS texts from two fields: GitHub and ACL Anthology. ACL Anthology hosts many natural language processing papers, including abstracts. Unlike typical paper-hosting websites, such as the ACM Digital Library, with ACL Anthology, most PDFs of the main body of papers are freely downloadable. Therefore, we chose ACL Anthology as our source of academic papers. 1,000 randomly selected abstracts out of the all obtained abstracts were used for the experiment. For another source of academic texts, we also obtained 55,410 abstracts from the PubMed website (`https://pubmed.ncbi.nlm.nih.gov/download/`) and used $1,000$ randomly selected abstracts out of the all abstracts.

Datasets for Software Manuals

When collecting software manuals, which are the focus of this paper, we made the following considerations. First, we excluded the software manuals of commercial software from our analysis. Such manuals are usually proofread by a professional proofreading company and are therefore outside the scope of this study, which focuses on the text difficulty of software manuals for ESL learners. Analyzing commercial software manuals would simply reveal the proofreading standards used by the proofreading company.

Instead, we are interested in the readability of software manuals for ESL learners for the following reasons. In the case of open-source software, both ESL learners and native English speakers are closely involved in software development, and there is usually no standard for the readability of software manuals developed in the open-source community. This is in contrast to the structure of a software manual, in which there are many rules for the documentation structure, even for open-source software.

For the above reasons, we selected GitHub, an open-source software hosting site, for this study's analysis. Although there are many projects on GitHub, there are also many software repositories that have not been maintained for years or have been developed by a single developer. Obviously, such software repositories are outside the scope of this study on the readability of software manuals for ESL learners. Therefore, we excluded such software repositories from the analysis.

Selection of Repositories

We need to find software repositories that are used by many software developers, including ESL learners. To this end, we analyzed the repositories on GitHub, one of the most popular code-hosting websites [3].

GitHub has a feature to find active projects: "GitHub Trending Repositories." This feature ranks the software that is active under the specified conditions. We can indicate the spoken language used in the projects, the (programming) language used in the projects, and the timespan of trending.

The trending timespan can be months, weeks, or days. "weeks" and "days" may be influenced by the week's or the day's programming contests or other competitions that happen to be held during a certain week. To eliminate such noise, we chose "months" and analyzed the monthly ranking. As the ranking of repositories changed significantly each month, we analyzed the top 10 README.md files by looking at monthly trends from November 2021 to January 2022.

The programming language is set to "Any" because there is no need to limit it this time.

If the spoken language is set to English, the target group of English learners may not be included in the survey. However, projects whose main language is not English are not relevant to ESL learners, so we want to exclude them. Therefore, we did the following. We set the spoken language to "Any," the (programming) language to "any," and the date range to "monthly" and displayed the Trending Repositories. A list of 25 repositories and README.md files are then shown. Here, we excluded repositories that did not use English in their README.md from our target analysis group. The top 10 repositories were then included in the survey.

While several of the top projects were maintained entirely in Chinese, we omitted these projects from our analysis since our focus is ESL learners.

Extracting README.md

Almost all manuals in GitHub's software repository are README.md files written in Markdown format. We analyzed the README.md of each of the repositories targeted using the above procedure; the README.md files may contain programming code. If the programming code portions are also part of the text, the readability will be inaccurate. Therefore, the **commonmark** library was used to analyze the Markdown format, and only the portions where text was used were included in the analysis.

After extracting the texts from both data sources, we applied two automatic readability assessors to each source. Texts were inputted into the assessors without sentence splitting because the assessors were not designed to accept inputs that were split sentence wise.

Results

Despite these differences, our experimental results demonstrated that the assessments of the two assessor types were generally similar. First, for both databases, as the assessment of the former assessor, the majority of abstracts were readable to intermediate English learners. The definition of the term "intermediate" follows that in (Vajjala & Lučić, 2018).

Table 2: Readability Assessment Results of CS Texts for ESL learners

-	Elem.	Int.	Adv.
GitHub Texts (Raw)	0.056	0.778	0.167
GitHub Texts (Code removed)	0.083	0.861	0.056
ACL Anthology	0.030	0.413	0.557
PubMed	0.005	0.189	0.806

The results are presented in Table 2. We trained the **BERTlcw** classifier as described in Section using the OneStopEnglish dataset and applied the classifier to assess the readability of each corpus. In Table 2, each element shows the ratio of each readability level. The sum of each row is 1.

From the table, we first see that GitHub Texts (Raw) and GitHub Texts (Code removed) have lower percentages of "Adv.," which is difficult to read for most English learners, is lower than that of the ACL Anthology. This indicates that GitHub is clearly more readable for ESL learners compared with ACL Anthology. Furthermore, the appropriate exclusion of program code from GitHub (Raw) texts decreases the ratio of advanced texts, indicating an increase in readability. Since GitHub Texts (Code removed) are mostly at the intermediate level, an intermediate English learner is likely to understand most of the GitHub texts.

In Table 2, the difference between ACL Anthology and GitHub Texts (Code Removed) and that between ACL Anthology and GitHub Texts (Raw) was statistically significant (Mann-Whitney U test, $p < 0.01$). This result clearly indicates that ACL Anthology was more difficult than GitHub Texts (Raw), and GitHub Texts (Code removed). In contrast, no statistical significance was found between GitHub Texts (Code removed) and GitHub Texts (Raw). This implies that the effect of removing code can be limited.

We also analyzed the text using the **Vocabulary-based** assessor. This assessor also assessed that GitHub is easier than ACL Anthology: The average readability score for GitHub (Code removed) was 0.117, and that for ACL Anthology was 0.140. A higher score indicates that the text is more difficult to read. The results of both assessors showed that the GitHub texts were easier to read than the ACL Anthology abstracts at a statistically significant level (Mann-Whitney tests, $p < 0.01$). This is presumably because academic writing in CS papers is particularly difficult for ESL learners, whereas such academic terminology is rarely used in software manuals. The qualitative results of the **Vocabulary-based** assessor confirmed this tendency. For example, the words that were were assessed as particularly difficult for ESL learners in the GitHub texts include *blockchain* and *automerge*, whereas those in the in ACL Anthology were *lexicosemantic*

[3] https://github.com/trending?since=monthly

and *colingual*.

Discussions

There are, of course, various limitations to the methodologies in this paper. First, the subdomain of computer science is not discussed in this paper, and tackling this will be a part of our future work. It is quite likely that some subdomains of CS are more difficult than others. Instead of discussing the subdomain of CS, we showed the results for the abstracts taken from PubMed in Table 2. The results confirm that there are more difficult words in PubMed than in ACL as in Table 2.

The code removal from the GitHub (Raw) texts is not perfect. We can remove apparent code blocks marked by the Markdown language, however, in GitHub texts, proper nouns such as software, function, and variable names may appear within English sentences. In such cases, proper nouns should not be removed because they are part of natural language sentences. This might cause the relatively small difference in readability between GitHub (Code removed) and GitHub (Raw) in Table 2.

Also, while we chose GitHub in this paper as an open source community where many ESL learners may be involved, obviously, GitHub is not the only open source community. The percentage of ESL learners within GitHub does not seem to have been investigated. It may be possible to estimate the extent to which ESL learners are involved in the open source community by identifying the languages used in open source projects using techniques such as language identification.

The time factor was not included in this study, and we assumed that the time a text was written did not affect its readability. One reason for this is that one of the goals of this study is to help ESL learners become active in the computer science community. As computer science technology rapidly evolves, ESL learners are unlikely to read older software texts to learn computer science.

Another reason is that the OneStopEnglish dataset (Vajjala & Lučić, 2018), which we used as the reliable readability source, was created using recent English news articles because the paper was published in 2018. It is questionable whether the readability of older English works can be accurately measured due to changes in the English language. Since the majority of texts in the ACL Anthology are from the 1970s onward, the impact of the changes in the English language on calculating readability may be limited.

We may need a manual evaluation of readability, especially for the distinction between intermediate and advanced texts. While the overall order of text readability is stable, whether texts are classified into intermediate or advanced sometimes depends on the initialization of deep-learning classifiers. (Ehara, 2021) also noted that the classification of intermediate and advanced is difficult. This can cause a discrepancy in actual ratios of elementary, intermediate, and advanced. Vocabulary-based text readability papers include (Ehara, Sato, Oiwa, & Nakagawa, 2012; Ehara, Miyao, Oiwa, Sato, & Nakagawa, 2014; Ehara, Baba, Utiyama, & Sumita, 2016; Lee & Yeung, 2018; Yeung & Lee, 2018).

Conclusions

We showed that many CS paper abstracts are unreadable by intermediate ESL learners, whereas CS software manuals are mostly readable to ESL learners. This implies that such learners need assistance in reading CS papers, whereas they need little assistance in reading software manuals. In this study, we identified the major tendencies in CS texts. Future work should confirm whether our findings hold true for a wider range of CS texts.

Future work should include a wider range of CS texts in the analysis. Further experimental results and follow-up studies of this work will be introduced in `http://yoehara.com/` or in `http://readability.jp/`.

Acknowledgments

This work was supported by JST ACT-X Grant Number JPMJAX2006, Japan. We used the ABCI infrastructure of AIST and the miniRaiden system of RIKEN for computational resources.

References

Anderson, J. (1983). Lix and rix: Variations on a little-known readability index. *Journal of Reading*, *26*(6), 490–496.

Baker, F. B. (2004). *Item Response Theory : Parameter Estimation Techniques, Second Edition.* CRC Press.

Beglar, D., & Nation, P. (2007). A vocabulary size test. *The Language Teacher*, *31*(7), 9–13.

Björnsson, C. H. (1968). Stockholm: Läsbarhet.

BNC Consortium. (2007). *The british national corpus, version 3 (bnc xml edition).* Distributed by Bodleian Libraries, University of Oxford, on behalf of the BNC Consortium `http://www.natcorp.ox.ac.uk/`.

Coleman, M., & Liau, T. L. (1975). A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, *60*(2), 283.

Dale, E., & Chall, J. S. (1948). A formula for predicting readability: Instructions. *Educational research bulletin*, 37–54.

Davies, M. (2008). *The corpus of contemporary american english (coca).* Available online at `https://www.english-corpora.org/coca/`.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019, June). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proc. of NAACL* (pp. 4171–4186). Minneapolis, Minnesota.

Ehara, Y. (2018, May). Building an English Vocabulary Knowledge Dataset of Japanese English-as-a-Second-Language Learners Using Crowdsourcing. In *Proc. of LREC.*

Ehara, Y. (2021). Lurat: a lightweight unsupervised automatic readability assessment toolkit for second language learners. In *2021 ieee 33rd international conference on tools with artificial intelligence (ictai)* (pp. 806–814).

Ehara, Y., Baba, Y., Utiyama, M., & Sumita, E. (2016). Assessing Translation Ability through Vocabulary Ability Assessment. In *Proc. of IJCAI.*

Ehara, Y., Miyao, Y., Oiwa, H., Sato, I., & Nakagawa, H. (2014). Formalizing Word Sampling for Vocabulary Prediction as Graph-based Active Learning. In *Proc. of EMNLP* (pp. 1374–1384). Retrieved 2019-11-30, from `https://www.aclweb.org/anthology/D14-1143` doi: 10.3115/v1/D14-1143

Ehara, Y., Sato, I., Oiwa, H., & Nakagawa, H. (2012, December). Mining Words in the Minds of Second Language Learners: Learner-Specific Word Difficulty. In *Proceedings of COLING 2012* (pp. 799–814). Mumbai, India: The COLING 2012 Organizing Committee.

Flesch, R. (1948). A new readability yardstick. *Journal of Applied Psychology*, *32*(3), 221–233.

Gunning, R. (1952). *The technique of clear writing*. McGraw-Hill.

Kincaid, J. P., Fishburne Jr, R. P., Rogers, R. L., & Chissom, B. S. (1975). *Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel* (Tech. Rep.). Naval Technical Training Command Millington TN Research Branch.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *Proc. of ICLR.*

Laufer, B., & Ravenhorst-Kalovski, G. C. (2010, April). Lexical Threshold Revisited: Lexical Text Coverage, Learners' Vocabulary Size and Reading Comprehension. *Reading in a Foreign Language*, *22*(1), 15–30. Retrieved 2019-10-06, from `https://eric.ed.gov/?id=EJ887873`

Lee, J., & Yeung, C. Y. (2018, August). Personalizing Lexical Simplification. In *Proceedings of the 27th International Conference on Computational Linguistics* (pp. 224–232). Santa Fe, New Mexico, USA: Association for Computational Linguistics. Retrieved 2020-05-23, from `https://www.aclweb.org/anthology/C18-1019`

Martinc, M., Pollak, S., & Robnik-Šikonja, M. (2021, April). Supervised and Unsupervised Neural Approaches to Text Readability. *Computational Linguistics*, *47*(1), 141–179.

Mc Laughlin, G. H. (1969). Smog grading-a new readability formula. *Journal of reading*, *12*(8), 639–646.

Nation, I. (2006, October). How Large a Vocabulary is Needed For Reading and Listening? *Canadian Modern Language Review*, *63*(1), 59–82.

Senter, R., & Smith, E. A. (1967). *Automated readability index* (Tech. Rep.). CINCINNATI UNIV OH.

Vajjala, S., & Lučić, I. (2018, June). OneStopEnglish corpus: A new corpus for automatic readability assessment and text simplification. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications* (pp. 297–304). New Orleans, Louisiana: Association for Computational Linguistics. Retrieved 2021-06-20, from `https://www.aclweb.org/anthology/W18-0535` doi: 10.18653/v1/W18-0535

Vajjala, S., & Rama, T. (2018, June). Experiments with Universal CEFR Classification. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications* (pp. 147–153). New Orleans, Louisiana: Association for Computational Linguistics. Retrieved 2021-06-20, from `https://www.aclweb.org/anthology/W18-0515` doi: 10.18653/v1/W18-0515

Yeung, C. Y., & Lee, J. (2018, August). Personalized Text Retrieval for Learners of Chinese as a Foreign Language. In *Proc. of COLING* (pp. 3448–3455).