ReadNet: A Hierarchical Transformer Framework for Web Article Readability Analysis

Changping Meng[1⋆], Muhao Chen[2], Jie Mao[3], Jennifer Neville[1]

Department of Computer Science, Purdue University, West Lafayette[1]
Department of Computer Science, University of California, Los Angeles[2]
Google Inc., Mountain View[3]
{meng40, neville}@purdue.edu; mjmjmtl@google.com; muhaochen@cs.ucla.edu

Abstract. Analyzing the *readability* of articles has been an important sociolinguistic task. Addressing this task is necessary to the automatic recommendation of appropriate articles to readers with different comprehension abilities, and it further benefits education systems, web information systems, and digital libraries. Current methods for assessing readability employ empirical measures or statistical learning techniques that are limited by their ability to characterize complex patterns such as article structures and semantic meanings of sentences. In this paper, we propose a new and comprehensive framework which uses a hierarchical self-attention model to analyze document readability. In this model, measurements of sentence-level difficulty are captured along with the semantic meanings of each sentence. Additionally, the sentence-level features are incorporated to characterize the overall readability of an article with consideration of article structures. We evaluate our proposed approach on three widely-used benchmark datasets against several strong baseline approaches. Experimental results show that our proposed method achieves the state-of-the-art performance on estimating the readability for various web articles and literature.

1 Introduction

Readability is an important linguistic measurement that indicates how easily readers can comprehend a particular document. Due to the explosion of Web and digital information, there are often hundreds of articles describing the same topic, but vary in levels of readability. This can make it challenging for users to find the articles online that better suit their comprehension abilities. Therefore, an automated approach to assessing readability is a critical component for the development of recommendation strategies for web information systems, including digital libraries and web encyclopedias.

Text readability is defined as the overall effect of language usage and composition on readers' ability to easily and quickly comprehend the document [14]. In this work, we focus on evaluating document difficulty based on the composition of words and sentences. Consider the following two descriptions of the concept *rainbow* as an example.

⋆ This work was done during the summer internships of CM and MC at Google, Mountain View. We thank the anonymous reviewers for their insightful comments.

1. **A more rigid scientific definition from *English Wikipedia***: A rainbow is a meteorological phenomenon that is caused by reflection, refraction and dispersion of light in water droplets resulting in a spectrum of light appearing in the sky.
2. **A more generic description from the *Simple English Wikipedia***: A rainbow is an arc of color in the sky that can be seen when the sun shines through falling rain. The pattern of colors starts with red on the outside and changes through orange, yellow, green, blue, to violet on the inside.

Clearly, the first description provides more rigidly expressed contents, but is more sophisticated due to complicated sentence structures and the use of professional words. In contrast, the second description is simpler, with respect to both grammatical and document structures. From the reader's perspective, the first definition is more appropriate for technically sophisticated audiences, while the second one is suitable for general audiences, such as parents who want to explain rainbows to their young children.

The goal of *Readability Analysis* is to provide a rating regarding the difficulty of an article for average readers. As the above example illustrates that, many approaches for automatically judging the difficulty of the articles are rooted in two factors: the difficulty of the words or phrases, and the complexity of syntax [11]. To characterize these factors, existing works [3,29] mainly rely on some explicit features such as *Average Syllables Per Word*, *Average Words Per Sentence*, etc. For example, the Flesch-Kincaid index is a representative empirical measure defined as a linear combination of these factors [4]. Some later approaches mainly focus on proposing new features with the latest CohMetrix 3.0 [36] providing 108 features, and they combine and use the features using either linear functions or statistical models such as Support Vector Machines or multilayer perceptron [43,12,41,40,51]. While these approaches have shown some merits, they also lead to several drawbacks. Specifically (1) they do not consider sequential and structural information, and (2) they do not capture sentences-level or document-level semantics that are latent but essential to the task [11].

To address these issues, we propose ReadNet, a comprehensive readability classification framework that uses a hierarchical transformer network. The self-attention portion of the transformer encoder is better able to model long-range and global dependencies among words. The hierarchical structure can capture how words form sentences, and how sentences form documents, meanwhile reduce the model complexity exponentially. Moreover, explicit features indicating the readability of different granularities of text can be leveraged and aggregated from multiple levels of the model. We compare our proposed model to a number of widely-adopted document encoding techniques, as well as traditional readability analysis approaches based on explicit features. Experimental results on three benchmark datasets show that our work properly identifies the document representation techniques, and achieves the state-of-the-art performance by significantly outperform previous approaches.

2 Related Work

Existing computational methods for readability analysis [3,29,53,11,40] mainly use empirical measures on the symbolic aspects of the text, while ignoring the sequence of

words and the structure of the article. The Flesch-Kincaid index [28] and related variations use a linear combination of explicit features.

Although models based on these traditional features are helpful to the quantification of readability for small and domain-specific groups of articles, they are far from generally applicable for a larger body of web articles [45,10,17]. Because those features or formulas generated from a small number of training text specifically selected by domain experts, they are far from generally representing the readability of large collections of corpora. Recent machine learning methods on readability evaluation are generally in the primitive stage. [18] proposes to combine language models and logistic regression. The existing way to integrate features is through a statistical learning method such as SVM [20,43,12,41,40,51]. These approaches ignore the sequential or structural information on how sentences construct articles. Efforts have also been made to select optimal features from current hundreds of features [15]. Some computational linguistic methods have been developed to extract higher-level language features. The widely-adopted Coh-Metrix [22,37] provides multiple features based on cohesion such as referential cohesion and deep cohesion.

Plenty of works have been conducted on utilizing neural models for sentimental or topical document classification or ranking, while few have paid attention to the readability analysis task. The convolutional neural network (CNN) [27] is often adopted in sentence-level classification which leverages local semantic features of sentence composition that are provided by word representation approaches. In another line of approaches, a recursive neural network [46] is adopted, which focuses on modeling the sequence of words or sentences. Hierarchical structures of such encoding techniques are proposed to capture structural information of articles, and have been widely used in tasks of document classification [48,32,7], and sequence generation [30] and sub-article matching [6]. Hierarchical attention network [52] is the current state-of-the-art method for document classification, which employs attention mechanisms on both word and sentence levels to capture the uneven contribution of different words and sentences to the overall meaning of the document. The Transformer model [50] uses multi-head self-attention to perform sequence-to-sequence translation. Self-attention is also adopted in text summarization, entailment and representation [38,31]. Unlike topic and sentiment-related document classification tasks that focus on leveraging portions of lexemes that are significant to the overall meanings and sentiment of the document, readability analysis requires the aggregation of difficulty through all sentence components. Besides, precisely capturing the readability of documents requires the model to incorporate comprehensive readability-aware features, including difficulty, sequence and structure information, to the corresponding learning framework.

3 Preliminary

In this section, we present the problem definition, as well as some representative explicit features that are empirically adopted for the readability analysis task.

3.1 Problem Definition

The readability analysis problem is defined as an ordinal regression problem for articles. Given an article with up to n sentences and each sentence with up to m words, an article can be represented as a matrix $\boldsymbol{A}$ whose i-th row $\boldsymbol{A}_{i,:}$ corresponds to the i-th sentences, and $A_{i,j}$ denotes the j-th word of the i-th sentence. Given an article $\boldsymbol{A}$, a label will be provided to indicate the readability of this article.

We consider the examples introduced in Section 1, where two articles describe the same term "*rainbow*". The first rigorous scientific article can be classified as "difficult", and the second general description article can be classified as "easy".

Instead of classifying articles into binary labels like "easy" or "difficult", more fine-grained labels can help people better understand the levels of readability. For instance, we can map the articles in standardization systems of English tests such as 5-level Cambridge English Exam (CEE), where articles from professional level English exam (CPE) are regarded than those from introductory English exam (KET).

3.2 Explicit Features

Previous works [28,21,25,24,34,11,22] have proposed empirical features to evaluate readability. Correspondingly, we divide these features into sentence-level features and document-level features. Sentence-level features seek to evaluate the difficulty of sentences. For instance, the sentence-level feature "number of words" for sentences can be averaged into "number of words per sentence" to evaluate the difficulty of documents. Document-level features include the traditional readability indices and cohesion's proposed by Coh-Metrix[22]. These features are listed in Table 1.

Current approaches [43,12,41] average the sentence-level features of each sentence to construct document level features. Furthermore, these features are concatenated with document-level features, and use an SVM to learn on these features. The limitation lies in failing to capture the structure information of sentences and documents. For instance, in order to get the sentence level features for the document, it averages all these features of each sentence. It ignores how these sentences construct an article and which parts of the document more significantly decides the readability of the document. While cohesion features provided by Coh-Metrix tries to captures relationships between sentences, these features mainly depend on the repeat of words across multiple sentences. They did not directly model how these sentences construct a document in perspectives of structure and sequence.

Briefly speaking, existing works are mainly contributing more features as shown in Table 1. But the current models used to aggregate these features are based on SVM and linear models. In this work, we target to propose a more advanced model to better combine these features with document information.

4 Hierarchical Transformer for Readability Analysis

In order to address the limitations of traditional approaches, we propose ReadNet: the Hierarchical Transformer model for readability analysis as shown in Figure 1.

Name	Description
Sentence-level features	
#characters_per_word	The average number of characters per word, which provides a character-level measure for the difficulty of words.
#syllabi_per_word	The average number of syllabi per word, which measures the difficulty of words from the syllabus level.
#words	The number of words that measures the verbosity of the sentence.
#long_words	The number of words longer than 6 characters in a sentence.
#difficult_words	The number of difficult word in a sentence. Difficult word is a word not listed in the 3000 words for fourth-grade American students.
#pronoun	The number of pronoun in a sentence.
Document-level features	
Flesch Reading Ease [28]	The United States Military Standard of readability scoring for technical manuals, which is calculated as $206.835 - 1.015 \times \frac{\#\text{words}}{\#\text{sentences}} - 84.6 \times \frac{\#\text{syllables}}{\#\text{words}}$.
Flesch-Kincaid grade level [28]	An empirical readability metric which maps to a U.S. school grade level, calculated as $0.39 \times \frac{\#\text{words}}{\#\text{sentences}} + 11.8 \times \frac{\#\text{syllables}}{\#\text{words}} - 15.59$.
Automated Readability Index [44]	A metric that also produces an approximate representation of the US grade level needed to comprehend the text, calculated as $4.71 \times \frac{\#\text{characters}}{\#\text{words}} + 0.5 \times \frac{\#\text{words}}{\#\text{sentences}}$ - 21.43. Instead of considering syllables, this metric more generally characterizes on the character level.
Coleman-Liau Index [9]	An index used to gauge the understandability of a text from the character-level: $0.0588 \times \frac{\#\text{letters}}{\#\text{words} \times 100} + 0.296 \times \frac{\#\text{sentences}}{\#\text{words}} \times 100$.
Gunning Fog Index [23]	$0.4 \times (\frac{\#\text{words}}{\#\text{sentences}} + 100 \times \frac{\#\text{complex\_words}}{\#\text{words}})$; It estimates the years of formal education a person needs to understand the text on the first reading.
LIX [2]	A measure indicating the difficulty of reading a text based on the proportions of long words and verbosity of sentences: $\frac{\text{word longer than 6 letters \#}}{\#\text{words}} + \frac{\#\text{words}}{\#\text{sentences}}$.
RIX [1]	A metric based on the proportion of long words in text, $\frac{\text{\# long words}}{\#\text{sentences}}$.
SMOG Index [35]	A measure of readability that seeks to estimate the years of education needed to understand a piece of writing: $1.0430 \times \sqrt{\text{\# of polysyllables} \times \frac{30}{\#\text{sentences}}} + 3.1291$.
Dale Chall Index [19]	$0.1579 \times \frac{\#\text{difficult\_words}}{\#\text{words}} \times 100 + 0.0496 \times \frac{\#\text{words}}{\#\text{sentences}}$. Difficult word is a word not listed in the 3000 words for fourth-grade American students
Incidence of connectives [33]	5 numerical features indicate additive, logic, temporal, causal and negative connectives.
Logic operator connectivity [13]	Logical connectives between logical particles such as "and", "if" proposed by Coh-Metrix.
Lexical diversity	The character-level density of the lexicon: $\frac{\#\text{unique\_words}}{\#\text{words}}$.
Content diversity	$\frac{\#\text{content\_words}}{\#\text{words}}$. It measures the diversity of content. Content words are adjectives, nouns, verbs and adverbs.
Incidence of part-of-speech elements	Incidence of word categories (adjectives, nouns, verbs, adverbs, pronouns) per 1000 words in the text

Table 1: Explicit Features

The proposed model incorporates the explicit features with a hierarchical document encoder that encodes the sequence and structural information of an article. The first level of the hierarchical learning architecture models the formation of sentences from words. The second level models the formation of the article from sentences. The self-attention encoder (to be described in subsection 4.1) is adapted from the vanilla Transformer encoder [50]. The hierarchical structure, attention aggregation layer, combination with explicit features and transfer layer are specially designed for this readability analysis task.

4.1 From Words to Sentences

In this subsection, we introduce the encoding process of sentences in hierarchical mutli-head self-attention. The encoding process has three steps: *1)* the self-attention encoder transforms the input sequence into a series of latent vectors; *2)* the attention layer aggregates the encoded sequential information based on the induced significance of input units; *3)* The encoded information is combined with the explicit features.

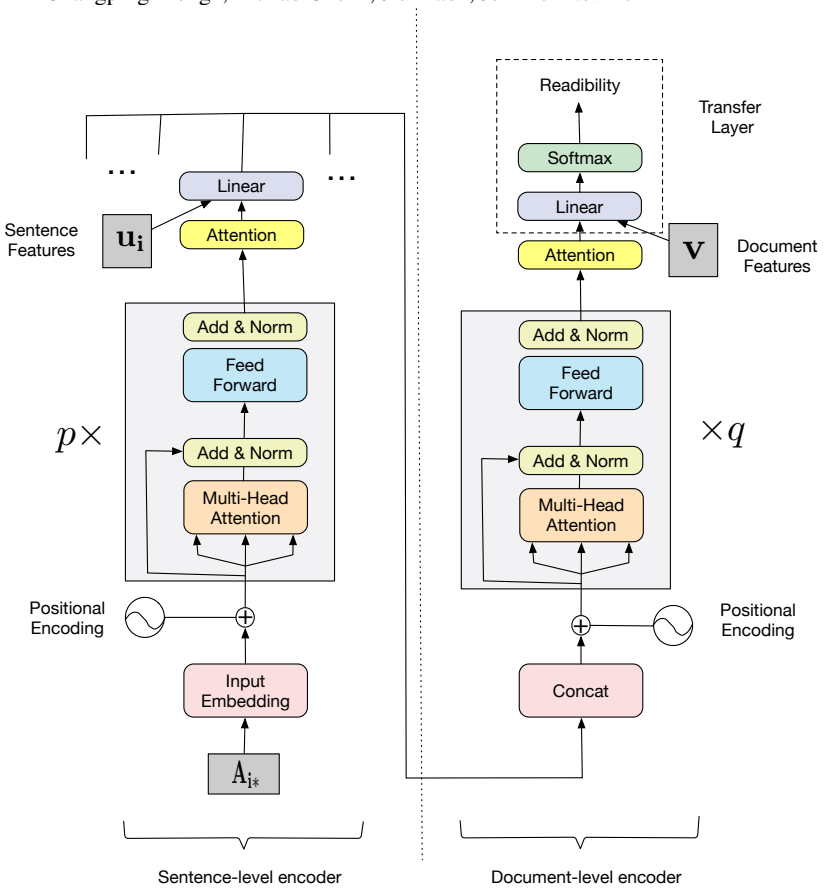

Fig. 1: ReadNet: proposed hierarchical transformer model specialized for readability analysis

Transformer Self-Attention Encoder This encoder is adapted from the vanilla Transformer encoder [50]. The input for this encoder is $\boldsymbol{A}_{i,:}$, which represents the i-th sentence.

The Embedding layer encodes each word $A_{i,j}$ into a d-dimensional vector based on word embedding. The output is a $m \times d$-dimensional matrix $\boldsymbol{B}$ where d is the embedding dimension and m is the number of words.

The position encoding layer indicates the relative position of each word $A_{i,j}$. The elements of positional embedding matrix $\boldsymbol{P}$ where values in the i-th row j-th column is defined as follows.

$$P_{i,j} = \begin{cases} \sin(i/10^{4j/d}) & j \text{ is even} \\ \cos(i/10^{4(j-1)/d}) & j \text{ is odd} \end{cases} \tag{1}$$

The embedded matrix $\boldsymbol{B}$ and positional embedding matrix $\boldsymbol{P}$ are added into the initial hidden state matrix $\boldsymbol{H}^{(0)} = \boldsymbol{B} + \boldsymbol{P}$. $\boldsymbol{H}^{(0)}$ will go through a stack of p identical layers. Each layer contains two parts: (i) the Multi-Head Attention donated as function

f_{MHA} defined in Equation 2, and (ii) the Position-wise Feed-Forward f_{FFN} defined in Equation 4. Layer normalization is used to avoid gradient vanishing or explosion.

Multi-head Self-Attention function(f_{MHA}) [50] encodes the relationship among query matrix $\boldsymbol{Q}$, key matrix $\boldsymbol{K}$ and value matrix $\boldsymbol{V}$ from different representation subspaces at different positions. $d_k = d/h$. $\boldsymbol{W}$ is a $d \times d$ weight matrix. $\oplus$ denotes concatenation. $\boldsymbol{W}_{Ki}, \boldsymbol{W}_{Vi}, \boldsymbol{W}_{Qi}$ are $d \times d_k$ weight matrix for head function g_i.

$$f_{MHA}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = (g_1(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})) \oplus ... \oplus g_h(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}))\boldsymbol{W} \tag{2}$$

$$g_i(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \mathrm{softmax}(\frac{\boldsymbol{Q}\boldsymbol{W}_{Qi}(\boldsymbol{K}\boldsymbol{W}_{Ki})^T}{\sqrt{d_k}})(\boldsymbol{V}\boldsymbol{W}_{Vi}) \tag{3}$$

Position-wise Feed-Forward Function f_{FFN} [50] adopts two 1-Dimensional convolution layers with kernel size 1 to encode input matrix $\boldsymbol{X}$.

$$f_{FFN}(\boldsymbol{X}) = \mathrm{Conv1D}(\mathrm{ReLU}(\mathrm{Conv1D}(\boldsymbol{X}))) \tag{4}$$

For the l-th encoder layer, $\boldsymbol{H}^{(l)}$ is encoded into $\boldsymbol{H}^{(l+1)}$ according to Equation 5

$$\boldsymbol{H}^{(l+1)} = f_{FFN}(f_{MHA}(\boldsymbol{H}^{(l)}, \boldsymbol{H}^{(l)}, \boldsymbol{H}^{(l)})) \tag{5}$$

Attention Aggregation Layer After p transformer encoder layers, each sentence $\boldsymbol{A}_{i,:}$ is encoded into a $m \times d$-dimensional matrix $\boldsymbol{H}^{(p)}$.

We first pass $\boldsymbol{H}^{(p)}$ through a feed forward layer with $d \times d$ dimensional weights $\boldsymbol{W}_1$ and bias term b_1 to obtain a hidden representation as $\boldsymbol{U}$:

$$\boldsymbol{U} = \tanh(\boldsymbol{H}^{(p)}\boldsymbol{W}_1 + b_1),$$

then compute the similarity between $\boldsymbol{U}$ and the trainable $d \times 1$ dimensional context matrix $\boldsymbol{C}$ via

$$\boldsymbol{w} = \mathrm{softmax}(\boldsymbol{U}\boldsymbol{C})\,,$$

which we use as importance weights to obtain the final embedding of the sentence $\boldsymbol{A}_{i,:}$:

$$\boldsymbol{h}_i = \sum_{byRow} \boldsymbol{H}^{(p)} \cdot \boldsymbol{w} \tag{6}$$

Combination of explicit features The sentence level features $\boldsymbol{u}_i$ introduced in Section 3.2 Table 1 for i-th sentence are concatenated by $\boldsymbol{h}_i^* = \boldsymbol{h}_i \oplus \boldsymbol{u}_i$.

4.2 From Sentences to Articles

The second level of the hierarchical learning architecture is on top of the first layer. n encoded vector $\boldsymbol{h}_i^* (1 \leq i \leq n)$ are concatenated as the input for this layer. The structure of second level is the same as the first level. The output of this level is a vector $\boldsymbol{y}$ as the overall embedding of this article.

4.3 Transfer layer

The goal of the transfer layer is to improve prediction quality on a target task where training data are scarce, while a large amount of other training data are available for a set of related tasks.

The readability analysis problem suffers from the lack of labeled data. Traditional benchmark datasets labeled by domain experts typically contain a small number of articles. For instance, CEE contains 800 articles and Weebit contains around 8 thousand articles. Such quantities of articles are far smaller than those for sentiment or topic-related document classification tasks which typically involve over ten thousand articles even for binary classification [27,7]. On the other hand, with the emerging of online encyclopedia applications such as Wikipedia, it provides a huge amount of training dataset. For instance, English Wikipedia and Simple-English Wikipedia contain more than 100 thousand articles which can be used to train a deep learning model.

One fully connected layer combines the article embedding vector $\boldsymbol{y}$ and document-level features $\boldsymbol{v}$ from Table 1 to output the readability label vector $\boldsymbol{r}$ after a Softmax function. $\boldsymbol{W}_t$ is the weight of the fully connected layer. For dataset with m categories of readability ratings, each document is embedded into $\boldsymbol{r}$ with $m-1$ dimensions.

$$\boldsymbol{r} = \mathrm{softmax}(\boldsymbol{W}_t(\boldsymbol{y} \oplus \boldsymbol{v}))$$

If transfer learning is needed, instead of random initialization, this network is initialized with a pre-trained network based on a larger corpus. During the training process, update the transfer layer while keeping all other layers frozen. If transfer learning is not needed, all layers are updated during the training process.

4.4 Learning Objective

Given dataset with m categories of readability ratings, the goal is to minimize ordinal regression loss [42] defined as Equation 7. $\boldsymbol{r}_k$ represents the k-th dimension of the $\boldsymbol{r}$ vector. y is the true label. The threshold parameter $\theta_1, \theta_2, ...\theta_{m-1}$ are also learned automatically from the data.

$$L(\boldsymbol{r}; y) = -\sum_{k=1}^{m-1} f(s(k;y)(\theta_k - \boldsymbol{r}_k)), \quad where \quad s(k;y) = \begin{cases} -1 & k < y \\ +1 & k \geq y \end{cases} \tag{7}$$

Here, the objective of learning the readability analysis model is essentially different from that of a regular document classification model, since the classes here do form a partial-order. However, the case of two classes degenerates the learning to the same as that of a binary classifer.

4.5 Why Hierarchical Self-attention

For self-attention, the path length in the computation graph between long-range dependencies in the network is $O(1)$ instead of $O(n)$ for recurrent models such as LSTM.

Shorter path length in the computation graph makes it easier to learn the interactions between any elements in the sequence. For readability analysis, modeling the overall interaction between words is more important than modeling the consequent words. For semantic understanding, the consequence of two words such as " very good " and "not good" make distinct semantic meanings. While for readability analysis, it does not make difference in difficulty to understand it. The overall evaluation of the words difficulties in the sentences matters.

The hierarchical learning structure benefits in two ways. First, it mimics human reading behaviors, since the sentence is a reasonable unit for people to read, process and understand. People rarely check the interactions between arbitrary words across different sentences in order to understand the article. Second, the hierarchical structure can reduce parameter complexity. For a document with n sentences, m words per sentence, d dimension per word, the parameter complexity of the model is $O((nm)^2d)$ for single level structure. While for the hierarchical structure, the parameter complexity is $O(m^2d + n^2d)$.

5 Experiments

In this section, we present the experimental evaluation of the proposed approach. We first introduce the datasets used for the experiments, followed by the comparison of the proposed approach and baselines based on held-out evaluation, as well as detailed ablation analysis of different techniques enabled by our approach.

5.1 Datasets

We use the following three datasets in our experiment. Table 2 reports the statistics of the three datasets including the average number of sentences per article n_{sent} and the average number of words per sentence n_{word}.

Wiki dataset [26] contains *English Wikipedia* and *Simple English Wikipedia*. Simple English Wikipedia thereof is a simplified version of English Wikipedia which only uses simple English words and grammars. This dataset contains 59,775 English Wikipedia articles and 59,775 corresponding Simple English Wikipedia articles.

Cambridge English Exam (CEE) [51] categorizes articles based on the criteria of five Cambridge English Exam level (KET, PET, FCE, CAE, CPE). The five ratings are sequentially from the easiest KET to the hardest CPE. In total, it contains 110 KET articles, 107 PET articles, 153 FCE articles, 263 CAE articles and 155 CPE articles. Even though this dataset designed for non-native speakers may differ from materials for native English speakers, the difficulty between five levels is still comparable. We test our model on this dataset in order to check whether our model can effectively evaluate the difficulty of English articles according to an existing standard.

Weebit [49] is one of the largest dataset for readability analysis. It contains 7,676 articles targeted at different age group readers from Weekly Reader magazine and BBC-Bitesize website. Weekly Reader magazine categorizes articles according to the ages of targeted readers in 7-8, 8-9 and 9-10 years old. BBC-Bitesize has two levels for age 11-14 and 15-16. The targeted age is used to evaluate readability levels.

Datasets	Wiki		Cambridge English Exam					WeeBit				
	En	Simple En	KET	PET	FCE	CAE	CPE	WR 2	WR 3	WR 4	KS3	GCSE
n_{sent}	37.46	7.74	6.30	8.80	16.47	10.63	16.69	23.41	23.28	28.12	22.71	27.85
n_{word}	17.03	14.41	9.40	16.63	17.96	16.39	23.47	12.56	13.48	16.29	20.04	18.62

Table 2: Statistics of datasets Wiki, Cambridge English Exam and Weebit

Accuracy	Explicit Features			Semantic Features			Explicit+Semantic	
	Logistic	SVM	MLP	CNN	LSTM	HATT	HATT+	ReadNet
Wiki	0.822 (±0.006)	0.848 (±0.008)	0.819 (±0.007)	0.583 (±0.035)	0.849 (±0.007)	0.877 (±0.007)	0.898 (±0.007)	**0.912 (±0.006)**
CEE	0.462 (±0.027)	0.492 (±0.041)	0.475 (±0.044)	0.277 (±0.031)	0.473 (±0.047)	0.512 (±0.043)	0.513 (±0.041)	0.528 (±0.045)
Weebit	0.724 (±0.007)	0.846 (±0.006)	0.845 (±0.006)	0.635 (±0.043)	0.886 (±0.005)	0.884 (±0.007)	0.902 (±0.006)	**0.917 (±0.006)**

Table 3: Cross-validation classification accuracy and standard deviation (in parentheses) on Wikipedia(Wiki), Cambridge English Exam (CEE) and Weebit dataset. We report accuracy on three groups of models: (1) statistical classification algorithms including multi-class logistic regression, Linear SVM and Multilayer Perceptron (MLP); (2) Three types of document classifier CNN, hierarchical GRNN using LSTM cells (LSTM), Hierarchical Attention Network (HATT); (3) Hierarchical Attention Network combined with explicit features(HATT+), and our proposed approach which combines explicit features and semantics with Hierarchical Self-Attention (ReadNet). Transfer learning is not used, and all parameters in the model are initialized randomly (Transfer learning is evaluated separately in Table 5).

	KET	PET	FCE	CAE	CPE
Scores	0.381 ± 0.078	0.544 ± 0.092	0.620 ± 0.054	0.671 ± 0.085	0.837 ± 0.071

Table 4: Average readability scores of 10 randomly selected articles in Cambridge English Test predicted by our model trained using Wikipedia. PET, KET , FCE, CPE and CAE have increasing difficulty levels according to Cambridge English. The scores are the confidence scores of classified as regular English Wikipedia instead of simple English Wikipedia.

5.2 Evaluation

In this subsection, we provide a detailed evaluation of the proposed approach.

Baseline approaches. We compare our proposed approach (denoted ReadNet) against the following baseline methods.

- Statistical classification algorithms based on explicit features: this category of baselines including the statistical classification algorithms that are widely adopted in a line of previous works [20,43,12,41,40,51], such as multi-class Logistic Regression, the Linear SVM, and the Multilayer Perceptron (MLP) [49]. Explicit features on which these models are trained have been introduced in Section 3.2. Since this work targets at proposing a more advanced model to utilize features instead of proposing new features, all these features from Table 1 are used.
- Neural document classifiers: this category of baselines represents the other line of previous works that adopt variants of neural document models for sentence or document classification. Corresponding approaches including the Convolutional Neural Networks (CNN) [27], the Hierarchical Gated Neural Network with Long Short-term Memory (LSTM) [48], and the Hierarchical Attention Network (HATT) [52].

– The Hierarchical Attention Network combined with explicit features (HATT+), for which we use the same mechanism as our proposed approach to incorporate the explicit features into the representation of each sentence by the attentive RNN.

Model configurations. For article encoding, we limit the number of sentences of each article to up to 50, zero-pad short ones and truncate over-length ones. According to the data statistics in Table 2, 50 sentences are enough to capture the majority of information of articles in the datasets. For each sentence, we also normalize the number of words to be fed into the model as 50, also via zero-padding and truncating. We fix the batch size to 32, and use Adam [16] as the optimizer with a learning rate 0.001. The epochs of training for the neural models are limited to 300. We set the number of encoder layers p and q to 6. The embedding dimension $d = 100$. Number of heads h in f_{MHA} is 3. CNN adopts the same configuration as [27]. Other statistical classification algorithms are trained until converge. Source code will be available in the final version.

Evaluation protocol. We formalize the task as a classification task following previous works on the three benchmark datasets. In order to provide a valid quantitative evaluation, we have to follow the existing evaluation method to show the advantage of our proposed model compared with the baselines. We adopt 5-fold cross-validation to evaluate the proposed model and baselines. We report the classification accuracy that is aggregated on all folds of validation.

Results. The results are reported in Table 3. Traditional explicit features can provide satisfying results. Since the multi-class logistic regression, SVM and MLP models can combine the features *number of words per sentence* and *number of syllabi per word* which are included in Flesch-Kincaid score, they provide the reasonable result. CNN is only slightly better than random guess. We assume that this is because CNN does not capture the sequential and structural information of documents. The HATT approach provides the best among models without explicit features. The reasons root in the structure of the model which is able to capture length and structural information of the article. Since it also adopted a hierarchical structure, the conciseness of each sentence and that of the overall article structure is captured, which appears to be significant to the task. The explicit features further improve the results of HATT as shown by HATT+. Even without explicit features, our proposed approach is better than HATT+. HATT has appeared to be successful at highlighting some lexemes and sentence components that are significant to the overall meanings or sentiment of a document. However, unlike topic and sentiment-related document classification tasks, readability does not rely on several consecutive lexemes, but the aggregation of all sentence components. The path length in the computation graph between arbitrary components dependencies in ReadNet is $O(1)$ instead of $O(n)$ for HATT. Shorter path length in the computation graph makes it easier to learn the interactions between any arbitrary words in sentence level, or sentences in document-level.

Compared with traditional approaches, the main advantage of the proposed approach is that it uses the document encoder to learn how words are connected into sentences and how sentences are connected into documents. Baseline approaches only use the averaged explicit features of all the sentences. For these datasets, several extremely difficult and complicated sentences usually determine the readability of a document.

This useful information is averaged and weakened by the total number of sentences in baselines.

5.3 Analysis on Transfer Learning

As shown in Table 3, the standard deviation of the CEE task is large compared with those in Wiki and Weebit tasks since the quantity of CEE articles is not enough to train a complex deep learning model. Transfer layer in ReadNet is utilized in three steps. First is to train and save the model from larger datasets such as Wiki or Weebit. Then, we initialize the model for CEE task and load the parameter weights from the saved model except for the transfer layer. Eventually on the target task, the transfer layer is trained while keeping all other layers fixed. As shown in Table 5, loading a pre-trained model based on Weebit or Wiki can increase the accuracy and decrease standard deviation on the CEE task. It is shown that a more accurate and stable model can be achieved by utilizing the transfer layer and well-trained models from related tasks.

	Original	Load Weebit	Load Wiki
Accuracy	0.528 (0.045)	0.568 (0.012)	0.561 (0.014)

Table 5: Accuracy for CEE classification using the transfer layer. Original is the model not using transfer learning, and without loading trained weights from other dataset. *Load Weebit* is to load the parameters weights trained in Weebit except the transfer layer. *Load Wiki* is to load the parameters weights trained in Wiki except the transfer layer.

Besides directly training and evaluating the same dataset, we also tried the model trained using Wikipedia dataset and evaluate on Cambridge English dataset. 10 articles are randomly selected from each level of Cambridge English Test. The probability of being classified as regular English Wikipedia instead of simple English Wikipedia is treated as the difficulty score. The average difficulty scores predicted by the model are shown in Table 4, which shows that our produced readability score implies correctly the difficulty of English documents for different levels of exams. A larger score indicates higher difficulty. These scores correctly indicate the difficulty levels of these exams.

6 Conclusion and Future Work

We have proposed a model to evaluate the readability of articles which can make great contributions to a variety of applications. Our proposed Hierarchical Self-Attention framework outperforms existing approaches by combining hierarchical document encoders with the explicit features proposed by linguistics. For future works, we are interested in providing the personalized recommendation of articles based on the combination of article readability and the understanding ability of the user. Currently, readability of articles only evaluate the texts of articles, other modalities such as images [39] and taxonomies [8] considered to improve readers' understanding. More comprehensive document encoders such as RCNN [5] and tree LSTM [47] may also be considered.

References

1. Anderson, J.: Lix and rix: Variations on a little-known readability index. Journal of Reading **26**(6), 490–496 (1983)
2. Brown, J., Eskenazi, M.: Student, text and curriculum modeling for reader-specific document retrieval. In: Proceedings of the IASTED International Conference on Human-Computer Interaction. Phoenix, AZ (2005)
3. Chall, J.S.: Readability: An appraisal of research and application (34) (1958)
4. Chall, J.S., Dale, E.: Readability revisited: The new Dale-Chall readability formula. Brookline Books (1995)
5. Chen, M., Ju, C., Zhou, G., Chen, X., Zhang, T., Chang, K.W., Zaniolo, C., Wang, W.: Multifaceted protein-protein interaction prediction based on siamese residual rcnn. Bioinformatics **35**(14), i305–i314 (07 2019)
6. Chen, M., Meng, C.P., Huang, G., Zaniolo, C.: Neural article pair modeling for wikipedia sub-article machine. In: ECML (2018)
7. Chen, M., Meng, C., Huang, G., Zaniolo, C.: Learning to differentiate between main-articles and sub-articles in wikipedia. In: Proceedings of the IEEE International Conference on Big Data (2019)
8. Chen, M., Tian, Y., Chen, X., Xue, Z., Zaniolo, C.: On2vec: Embedding-based relation prediction for ontology population. In: Proceedings of the 2018 SIAM International Conference on Data Mining. pp. 315–323. SIAM (2018)
9. Coleman, M., Liau, T.L.: A computer readability formula designed for machine scoring. Journal of Applied Psychology **60**(2), 283 (1975)
10. Collins-Thompson, K., Callan, J.: A language modeling approach to predicting reading difficulty. In: Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004 (2004)
11. Collins-Thompson, K.: Computational assessment of text readability: A survey of current and future research (2014)
12. Collins-Thompson, K., Callan, J.: Predicting reading difficulty with statistical language models. Journal of the American Society for Information Science and Technology **56**(13), 1448–1462 (2005)
13. Coxhead, A.: A new academic word list. TESOL quarterly **34**(2), 213–238 (2000)
14. Dale, E., Chall, J.S.: The concept of readability. Elementary English **26**(1), 19–26 (1949)
15. De Clercq, O., Hoste, V.: All mixed up? finding the optimal feature set for general readability prediction and its application to english and dutch. Computational Linguistics **42**(3), 457–490 (2016)
16. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research **12**(Jul), 2121–2159 (2011)
17. Feng, L., Elhadad, N., Huenerfauth, M.: Cognitively motivated features for readability assessment. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics. pp. 229–237. Association for Computational Linguistics (2009)
18. François, T.L.: Combining a statistical language model with logistic regression to predict the lexical and syntactic difficulty of texts for ffl. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop. pp. 19–27. Association for Computational Linguistics (2009)
19. Fry, E.: A readability formula that saves time. Journal of reading **11**(7), 513–578 (1968)
20. Fry, E.B.: The varied uses of readability measurement today. Journal of Reading (1987)
21. Gibson, E.: Linguistic complexity: Locality of syntactic dependencies. Cognition (1998)

22. Graesser, A.C., McNamara, D.S., Louwerse, M.M., Cai, Z.: Coh-metrix: Analysis of text on cohesion and language. Behavior research methods, instruments, & computers **36**(2), 193–202 (2004)
23. Gunning, R.: The fog index after twenty years. Journal of Business Communication **6**(2), 3–13 (1969)
24. Heilman, Michael, K.C.T., Eskenazi, M.: An analysis of statistical models and features for reading difficulty prediction. In: 3rd workshop on innovative use of NLP for building educational applications (2008)
25. Heilman, M., etc.: Combining lexical and grammatical features to improve readability measures for first and second language texts. In: Human Language Technologies (2007)
26. Kauchak, D.: Improving text simplification language modeling using unsimplified text data. In: Proceedings of the 51st annual meeting of the association for computational linguistics (volume 1: Long papers). vol. 1, pp. 1537–1546 (2013)
27. Kim, Y.: Convolutional neural networks for sentence classification. In: Empirical Methods in Natural Language Processing (2014)
28. Kincaid, J.P., Jr, R.P.F., Rogers, R.L., Chissom, B.S.: Derivation of new readability formulas for navy enlisted personnel (1975)
29. Klare, G.R.: The measurement of readability: useful information for communicators. ACM Journal of Computer Documentation (JCD) **24**(3), 107–121 (2000)
30. Li, J., Luong, M.T., Jurafsky, D.: A hierarchical neural autoencoder for paragraphs and documents. arXiv preprint arXiv:1506.01057 (2015)
31. Li, Z., Wei, Y., Zhang, Y., Yang, Q.: Hierarchical attention transfer network for cross-domain sentiment classification. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
32. Lin, R., Liu, S., Yang, M., Li, M., Zhou, M., Li, S.: Hierarchical recurrent neural network for document modeling. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 899–907 (2015)
33. Louwerse, M.: An analytic and cognitive parametrization of coherence relations. Cognitive linguistics **12**(3), 291–316 (2001)
34. Malvern, D., Richards, B.: Measures of lexical richness. The encyclopedia of applied linguistics (2012)
35. Mc Laughlin, G.H.: Smog grading-a new readability formula. Journal of reading **12**(8), 639–646 (1969)
36. McNamara, D.S., Graesser, A.C., McCarthy, P.M., Cai, Z.: Automated evaluation of text and discourse with Coh-Metrix. Cambridge University Press (2014)
37. McNamara, D.S., Louwerse, M.M., McCarthy, P.M., Graesser, A.C.: Coh-metrix: Capturing linguistic features of cohesion. Discourse Processes **47**(4), 292–330 (2010)
38. Parikh, A.P., Täckström, O., Das, D., Uszkoreit, J.: A decomposable attention model for natural language inference. arXiv preprint arXiv:1606.01933 (2016)
39. Pezeshkpour, P., Chen, L., Singh, S.: Embedding multimodal relational data for knowledge base completion. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 3208–3218 (2018)
40. Pilán, I., Volodina, E., Zesch, T.: Predicting proficiency levels in learner writings by transferring a linguistic complexity model from expert-written coursebooks. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. pp. 2101–2111 (2016)
41. Pitler, E., Nenkova, A.: Revisiting readability: A unified framework for predicting text quality. In: Proceedings of the conference on empirical methods in natural language processing. pp. 186–195. Association for Computational Linguistics (2008)

42. Rennie, J.D., Srebro, N.: Loss functions for preference levels: Regression with discrete ordered labels. In: Proceedings of the IJCAI multidisciplinary workshop on advances in preference handling. pp. 180–186. Kluwer Norwell, MA (2005)
43. Schwarm, S.E., Ostendorf, M.: Reading level assessment using support vector machines and statistical language models. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. pp. 523–530. Association for Computational Linguistics (2005)
44. Senter, R., Smith, E.A.: Automated readability index. Tech. rep., CINCINNATI UNIV OH (1967)
45. Si, L., Callan, J.: A statistical model for scientific readability. In: CIKM. vol. 1, pp. 574–576 (2001)
46. Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A., Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the 2013 conference on empirical methods in natural language processing. pp. 1631–1642 (2013)
47. Tai, K.S., Socher, R., Manning, C.D.: Improved semantic representations from tree-structured long short-term memory networks. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 1556–1566 (2015)
48. Tang, D., Qin, B., Liu, T.: Document modeling with gated recurrent neural network for sentiment classification. In: Proceedings of the 2015 conference on empirical methods in natural language processing. pp. 1422–1432 (2015)
49. Vajjala, S., Meurers, D.: On improving the accuracy of readability classification using insights from second language acquisition. In: Proceedings of the seventh workshop on building educational applications using NLP. pp. 163–173. Association for Computational Linguistics (2012)
50. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
51. Xia, M., Kochmar, E., Briscoe, T.: Text readability assessment for second language learners. In: Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications. pp. 12–22 (2016)
52. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E.: Hierarchical attention networks for document classification. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 1480–1489 (2016)
53. Zakaluk, B.L., Samuels, S.J.: Readability: Its Past, Present, and Future. ERIC (1988)