Enriching Word Embeddings with Domain Knowledge for Readability Assessment

Zhiwei Jiang and **Qing Gu**[*] and **Yafeng Yin** and **Daoxu Chen**
State Key Laboratory for Novel Software Technology,
Nanjing University, Nanjing 210023, China
jiangzhiwei@outlook.com, {guq,yafeng,cdx}@nju.edu.cn

Abstract

In this paper, we present a method which learns the word embedding for readability assessment. For the existing word embedding models, they typically focus on the syntactic or semantic relations of words, while ignoring the reading difficulty, thus they may not be suitable for readability assessment. Hence, we provide the knowledge-enriched word embedding (KEWE), which encodes the knowledge on reading difficulty into the representation of words. Specifically, we extract the knowledge on word-level difficulty from three perspectives to construct a knowledge graph, and develop two word embedding models to incorporate the difficulty context derived from the knowledge graph to define the loss functions. Experiments are designed to apply KEWE for readability assessment on both English and Chinese datasets, and the results demonstrate both effectiveness and potential of KEWE.

1 Introduction

Readability assessment is a classic problem in natural language processing, which attracts many researchers' attention in recent years (Todirascu et al., 2016; Schumacher et al., 2016; Cha et al., 2017). The objective is to evaluate the readability of texts by levels or scores. The majority of recent readability assessment methods are based on the framework of supervised learning (Schwarm and Ostendorf, 2005) and build classifiers from hand-crafted features extracted from the texts. The performance of these methods depends on designing effective features to build high-quality classifiers.

Designing hand-crafted features are essential but labor-intensive. It is desirable to learn representative features from the texts automatically. For document-level readability assessment, an effective feature learning method is to construct the representation of documents by combining the representation of the words contained (Kim, 2014). For the representation of word, a useful technique is to learn the word representation as a dense and low-dimensional vector, which is called word embedding. Existing word embedding models (Collobert et al., 2011; Mikolov et al., 2013; Pennington et al., 2014) can be used for readability assessment, but the effectiveness is compromised by the fact that these models typically focus on the syntactic or semantic relations of words, while ignoring the reading difficulty. As a result, words with similar functions or topics, such as "man" and "gentleman", are mapped into close vectors although their reading difficulties are different. It calls for incorporating the knowledge on reading difficulty when training the word embedding.

In this paper, we provide the knowledge-enriched word embedding (KEWE) for readability assessment, which encodes the knowledge on reading difficulty into the representation of words. Specifically, we define the word-level difficulty from three perspectives, and use the extracted knowledge to construct a knowledge graph. After that, we derive the difficulty context of words from the knowledge graph, and develop two word embedding models to incorporate the difficulty context to define the loss functions.

We apply KEWE for document-level readability assessment under the supervised framework. The experiments are conducted on four datasets of either English or Chinese. The results demonstrate that

[*] Corresponding Author

Proceedings of the 27th International Conference on Computational Linguistics, pages 366–378
Santa Fe, New Mexico, USA, August 20-26, 2018.

our method can outperform other well-known readability assessment methods, and the classic text-based word embedding models on all the datasets. By concatenating our knowledge-enriched word embedding with the hand-crafted features, the performance can be further improved.

The rest of the paper is organized as follows. Section 2 provides the related work for readability assessment. Section 3 describes the details of KEWE. Section 4 presents the experiments and results. Finally Section 5 concludes the paper with future work.

2 Related Work

In this section, we briefly introduce three research topics relevant to our work: readability assessment, word embedding, and graph embedding.

Readability Assessment. The researches on readability assessment have a relatively long history from the beginning of last century (Collinsthompson, 2014). Early studies mainly focused on designing readability formulas to evaluate the reading scores of texts. Some of the well-known readability formulas include the SMOG formula (McLaughlin, 1969), the FK formula (Kincaid et al., 1975), and the Dale-Chall formula (Chall, 1995). At the beginning of the 21th century, supervised approaches have been introduced and then explored for readability assessment (Si and Callan, 2001; Collins-Thompson and Callan, 2004; Schwarm and Ostendorf, 2005). Researchers have focused on improving the performance by designing highly effective features (Pitler and Nenkova, 2008; Heilman et al., 2008; Feng et al., 2010; Vajjala and Meurers, 2012) and employing effective classification models (Heilman et al., 2007; Kate et al., 2010; Ma et al., 2012; Jiang et al., 2015; Cha et al., 2017). While most studies are conducted for English, there are studies for other languages, such as French (François and Fairon, 2012), German (Hancke et al., 2012), Bangla (Sinha et al., 2014), Basque (Gonzalez-Dios et al., 2014), Chinese (Jiang et al., 2014), and Japanese (Wang and Andersen, 2016).

Word Embedding. Researchers have proposed various methods on word embedding, which mainly include two broad categories: neural network based methods (Bengio et al., 2003; Collobert et al., 2011; Mikolov et al., 2013) and co-occurrence matrix based methods (Turney and Pantel, 2010; Levy and Goldberg, 2014b; Pennington et al., 2014). Neural network based methods learn word embedding through training neural network models, which include NNLM (Bengio et al., 2003), C&W (Collobert and Weston, 2008), and word2vec (Mikolov et al., 2013). Co-occurrence matrix based methods learn word embedding based on the co-occurrence matrices, which include LSA (Deerwester, 1990), Implicit Matrix Factorization (Levy and Goldberg, 2014b), and GloVe (Pennington et al., 2014). Besides the general word embedding learning methods, researchers have also proposed methods to learn word embedding to include certain properties (Liu et al., 2015; Shen and Liu, 2016) or for certain domains (Tang et al., 2014; Ren et al., 2016; Alikaniotis et al., 2016; Wu et al., 2017).

Graph embedding. Graph embedding aims to learn continuous representations of the nodes or edges based on the structure of a graph. The graph embedding methods can be classified into three categories (Goyal and Ferrara, 2017): factorization based (Roweis and Saul, 2000; Belkin and Niyogi, 2001), random walk based (Perozzi et al., 2014; Grover and Leskovec, 2016), and deep learning based (Wang et al., 2016). Among them, the random walk based methods are easy to comprehend and can effectively reserve the centrality and similarity of the nodes. Deepwalks (Perozzi et al., 2014) and node2vec (Grover and Leskovec, 2016) are two representatives of the random walk based methods. The basic idea of Deepwalk is viewing random walk paths as sentences, and feeding them to a general word embedding model. node2vec is similar to Deepwalk, although it simulates a biased random walk over graphs, and often provides efficient random walk paths.

3 Learning Knowledge-Enriched Word Embedding for Readability Assessment

In this section, we present the details of Knowledge-Enriched Word Embedding (KEWE) for readability assessment. By incorporating the word-level readability knowledge, we extend the existing word embedding model and design two models with different learning structures. As shown in Figure 1, the above one is the knowledge-only word embedding model (KEWE$_k$) which only takes in the domain knowledge,

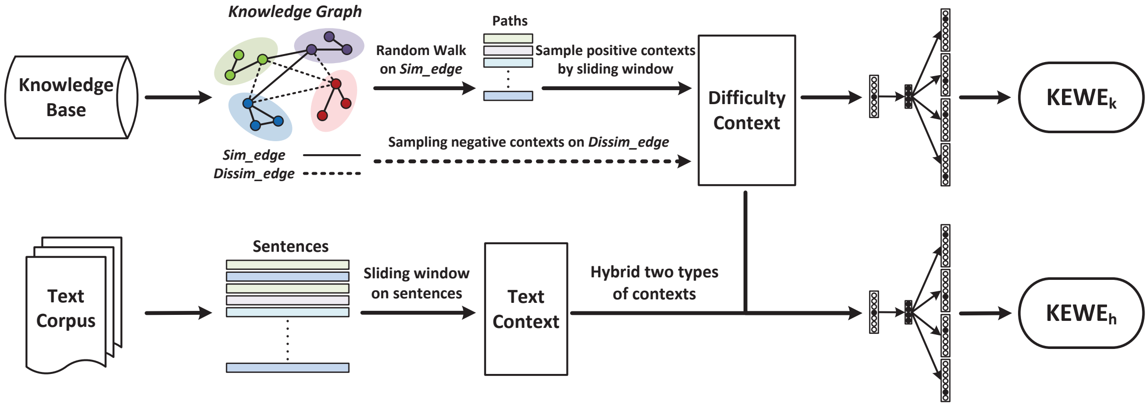
Knowledge Base
Knowledge Graph
Sim_edge
Dissim_edge
Random Walk on Sim_edge
Paths
Sample positive contexts by sliding window
Sampling negative contexts on Dissim_edge
Difficulty Context
KEWE_k
Text Corpus
Sentences
Sliding window on sentences
Text Context
Hybrid two types of contexts
KEWE_h

Figure 1: Illustration of the knowledge-enriched word embedding models. KEWE$_k$ is based on the difficulty context, while KEWE$_h$ is based on both the difficulty and text contexts.

the other is the hybrid word embedding model (KEWE$_h$) which compensates the domain knowledge with text corpus.

3.1 The Knowledge-only Word Embedding Model (KEWE$_k$)

In the classic word embedding models, such as C&W, CBOW, and Skip-Gram, the context of a word is represented by its surrounding words in the text corpus. Levy and Goldberg (2014a) have incorporated the syntactic context from the dependency parse-trees and found that the trained word embedding could capture more functional and less topical similarity. For readability assessment, reading difficulty other than function or topic becomes more important. Hence, we introduce a kind of difficulty context, and try to learn a difficulty-focusing word embedding, which leads to KEWE$_k$. In the following, we describe this model in three steps: domain knowledge extraction, knowledge graph construction, and graph-based word embedding learning. The former two steps focus on modeling the relationship among words on reading difficulty, and the final step on deriving the difficulty context and learning the word embedding.

3.1.1 Domain Knowledge Extraction

To model the relationship among words on reading difficulty, we first introduce how to extract the knowledge on word-level difficulty from different perspectives. Specifically, we consider three types of word-level difficulty: acquisition difficulty, usage difficulty, and structure difficulty.

Acquisition difficulty. Word acquisition refers to the temporal stage at which children learn the meaning of new words. Researchers have shown that the information on word acquisition is useful for readability assessment (Kidwell et al., 2009; Schumacher et al., 2016). Generally, the words acquired at primary school are easier than those acquired at high school. We call the reading difficulty reflected by word acquisition as the acquisition difficulty. Formally, given a word w, its acquisition difficulty is described by a distribution K_w^A over the age-of-acquisition (AoA) (Kidwell et al., 2009).

Since the rating on AoA is an unsolved problem in cognitive science (Brysbaert and Biemiller, 2016) and not available for many languages, we explore extra materials to describe the acquisition difficulty. In particular, we collect three kinds of knowledge teaching materials, i.e., in-class teaching material, extra-curricular teaching material, and proficiency test material. These materials are arranged as lists of words, each of which contains words learned in the same time period and hence corresponds to a certain level of acquisition difficulty. For example, given a lists of words, we can define $K_w^A \in \mathbb{R}^a$, where $K_{w,i}^A = 1$ if a word w belongs to the list i, and $K_{w,i}^A = 0$ otherwise.

Usage difficulty. Researchers used to count the usage frequency to measure the difficulty of words (Dale and Chall, 1948), which can separate the words which are frequently used from those rarely used. We call the difficulty reflected by usage preference as the usage difficulty. Formally, given a word w, its usage difficulty is described by a distribution K_w^U over the usage preferences.

We provide two ways to measure the usage difficulty. One way is estimating the level of words' usage frequency by counting the word frequency lists from the text corpus. The other way is estimating the probability distribution of words over the sentence-level difficulties, which is motivated by Jiang et al. (2015). Usage difficulty is defined on both. By discretizing the range of word frequency into b intervals of equal size, the usage frequency level of a word w is i, if its frequency resides in the ith intervals. By estimating the probability distribution vector P_w from sentence-level difficulties, we can define $K_w^U \in \mathbb{R}^{1+|P_w|}$, and $K_{w,i}^U = [i, P_w]$.

Structure difficulty. When building readability formulas, researchers have found that the structure of words could imply its difficulty (Flesch, 1948; Gunning, 1952; McLaughlin, 1969). For example, words with more syllables are usually more difficult than words with less syllables. We call the difficulty reflected by structure of words as the structure difficulty. Formally, given a word w, its structure difficulty can be described by a distribution K_w^S over the word structures.

Words in different languages may have their own special structural characteristics. For example, in English, the structural characteristics of words relate to syllables, characters, affixes, and subwords. Whereas in Chinese, the structural characteristics of words relate to strokes and radicals of Chinese characters. Here we use the number of syllables (strokes for Chinese) and characters in a word w to describe its structure difficulty. By discretizing the range of each number into intervals, K_w^S is obtained by counting the interval in which w resides, respectively.

3.1.2 Knowledge Graph Construction

After extracting the domain knowledge on word-level difficulty, we then quantitatively represent the knowledge by a graph. We define the knowledge graph as an undirected graph $G = (V, E)$, where V is the set of vertices, each of which represents a word, and E is the set of edges, each of which represents the relation (i.e., similarity) between two words on difficulty. Each edge $e \in E$ is a vertex pair (w_i, w_j) and is associated with a weight z_{ij}, which indicates the strength of the relation. If no edge exists between w_i and w_j, the weight $z_{ij} = 0$. We define two edge types in the graph: *Sim_edge* and *Dissim_edge*. The former indicates that its end words have similar difficulty and is associated with a positive weight. The latter indicates that its end words have significant different difficulty and is associated with a negative weight. We derived the edges from the similarities computed between pairs of the words' knowledge vectors. Formally, given the extracted knowledge vector $K_w = [K_w^A, K_w^U, K_w^S]$ of a word w, E can be constructed using the similarity between pairs of words (w_i, w_j) as follows:

$$z_{ij} = \begin{cases} sim(K_{w_i}, K_{w_j}) & w_j \in \mathcal{N}_p(w_i) \\ -sim(K_{w_i}, K_{w_j}) & w_j \in \mathcal{N}_n(w_i) \\ 0 & otherwise \end{cases} \tag{1}$$

where $sim()$ is a similarity function (e.g., cosine similarity), $\mathcal{N}_p(w_i)$ refers to the set of k most similar (i.e., greatest similarity) neighbors of w_i, and $\mathcal{N}_n(w_i)$ refers to the set of k most dissimilar (i.e., least similarity) neighbors of w_i.

3.1.3 Knowledge Graph-based Word Embedding

After constructing the knowledge graph, which models the relationship among words on difficulty, we can derive the difficulty context from the graph and train the word embedding focused on reading difficulty. For the graph-based difficulty context, given a word w, we define its difficulty context as the set of other words that have relevance to w on difficulty. Specifically, we define two types of difficulty context, positive context and negative context, corresponding to the two types of edges in the knowledge graph (i.e., *Sim_edge* and *Dissim_edge*).

Unlike the context defined on texts, which can be sampled by sliding windows over consecutive words, the context defined on a graph requires special sampling strategies. Different sampling strategies may define the context differently. For difficulty context, we design two relatively intuitive strategies, the random walk strategy and the immediate neighbors strategy, for the sampling of either positive or negative context.

From the type *Sim_edge*, we sample the positive target-context pairs where the target word and the context words are similar on difficulty. Since the similarity is generally transitive, we adopt the random walk strategy to sample the positive context. Following the idea of node2vec (Grover and Leskovec, 2016), we sample the positive contexts of words by simulating a 2^{nd} order random walk on the knowledge graph with only *Sim_edge*. After that, by applying a sliding window of fixed length s over the sampled random walk paths, we can get the positive target-context pairs $\{(w_t, w_c)\}$.

From the type *Dissim_edge*, we sample the negative target-context pairs where the target word and the context words are dissimilar on difficulty. Since dissimilarity is generally not transitive, we adopt the immediate neighbor strategy to sample the negative context. Specifically, on the knowledge graph with only *Dissim_edge*, we collect the negative context from the immediate neighbors of the target node w_t and get the negative context list $C_n(w_t)$.

By replacing the text-based linear context with our graph-based difficulty context, we can train the word embedding using the classic word embedding models, such as C&W, CBOW, and Skip-Gram. Here we use the Skip-Gram model with Negative Sampling (SGNS) proposed by Mikolov et al. (2013). Specifically, given N positive target-context pairs (w_t, w_c) and the negative context list of the target word $C_n(w_t)$, the objective of KEWE$_k$ is to minimize the loss function $\mathcal{L}_k$, which is defined as follows:

$$\mathcal{L}_k = -\frac{1}{N} \sum_{(w_t, w_c)} \Big[\log \sigma(\mathbf{u}_{w_c}^\top \mathbf{v}_{w_t}) + \mathbb{E}_{w_i \in C_n(w_t)} \log \sigma(-\mathbf{u}_{w_i}^\top \mathbf{v}_{w_t}) \Big] \tag{2}$$

where $\mathbf{v}_w$ and $\mathbf{u}_w$ are the "input" and "output" vector representation of w, and σ is the sigmoid function defined as $\sigma(x) = \frac{1}{(1+e^{-x})}$. This loss function enables the positive context (e.g., w_c) to be distinguished from the negative context (e.g., w_i).

3.2 The Hybrid Word Embedding Model (KEWE$_h$)

The classic text-based word embedding models yield word embedding focusing on syntactic and semantic contexts, while ignoring the word difficulty. By contrast, KEWE$_k$ trains the word embedding focusing on the word difficulty, while leaving out the syntactic and semantic information. Since readability may also relate to both syntax and semantics, we develop a hybrid word embedding model (KEWE$_h$), to incorporate both domain knowledge and text corpus. The loss function of the hybrid model $\mathcal{L}_h$ can be expressed as follows:

$$\mathcal{L}_h = \lambda \mathcal{L}_k + (1 - \lambda) \mathcal{L}_t \tag{3}$$

where $\mathcal{L}_k$ is the loss of predicting the knowledge graph-based difficulty contexts, $\mathcal{L}_t$ is the loss of predicting the text-based syntactic and semantic contexts, and $\lambda \in [0, 1]$ is a weighting factor. Clearly, the case of $\lambda = 1$ reduces the hybrid model to the knowledge-only model.

As there are many text-based word embedding models, the text-based loss $\mathcal{L}_t$ can be defined in various ways. To be consistent with KEWE$_k$, we formalize $\mathcal{L}_t$ based on the Skip-Gram model. Given a text corpus, the Skip-Gram model aims to find word representations that are good at predicting the context words. Specifically, given a sequence of training words, denoted as $w_1, w_2, \cdots, w_T$, the objective of Skip-Gram model is to minimize the log loss of predicting the context using target word embedding, which can be expressed as follows:

$$\mathcal{L}_t = -\frac{1}{T} \sum_{t=1}^{T} \sum_{-s \leq j \leq s, j \neq 0} \log p(w_{t+j} | w_t) \tag{4}$$

where s is the window size of the context sampling. Since the full softmax function used to define $p(w_{t+j}|w_t)$ is computationally expensive, we employ the negative sampling strategy (Mikolov et al., 2013) and replace every $\log p(w_c|w_t)$ in $\mathcal{L}_t$ by the following formula:

$$\log p(w_c|w_t) = \log \sigma(\mathbf{u}_{w_c}^\top \mathbf{v}_{w_t}) + \sum_{i=1}^{k} \mathbb{E}_{w_i \sim P_n(w)} \log \sigma(-\mathbf{u}_{w_i}^\top \mathbf{v}_{w_t}) \tag{5}$$

where $\mathbf{v}_w$, $\mathbf{u}_w$, and σ are of the same meanings as in Eq. 2, k is the number of negative samples, and $P_n(w)$ is the noise distribution. This strategy enables the actual context w_c to be distinguished from the noise context w_i drawn from the noise distribution $P_n(w)$.

3.3 Model Training

We adopt the stochastic gradient descent to train our models. Specifically, for the hybrid model (KEWE$_h$), we adopt the mini-batch mode used in (Yang et al., 2016). Firstly, we sample a batch of random walk paths of size N_1 and take a gradient step to optimize the loss function $\mathcal{L}_k$. Secondly, we sample a batch of text sentences of size N_2 and take a gradient step to optimize the loss function $\mathcal{L}_t$. We repeat the above procedures until the model converged or the predefined number of iterations is reached. The ratio $\frac{N_1}{N_1+N_2}$ is used to approximate the weighting factor λ in Eq. 3. For training the knowledge-only model, the process is the same without $\mathcal{L}_t$ and λ.

3.4 Readability Assessment

We apply KEWE for document-level readability assessment under a supervised learning framework proposed by Schwarm and Ostendorf (2005). The classifier for readability assessment is built on documents with annotated reading levels. Instead of using the hand-crafted features, we use the word embedding to produce the features of documents.

To extract high level features from documents using word embedding, we design the max layer similar to the one used in the convolutional sentence approach proposed by Collobert et al. (Collobert et al., 2011). The max layer is used to generate a fix-size feature vector from variant length sequences. Specifically, given a document represented by a matrix $M \in \mathbb{R}^{m \times n}$, where the kth column is the word embedding of the kth word in the document, the max layer output a vector $f_{max}(M)$:

$$[f_{max}(M)]_i = \max_t \, [M]_{i,t} \qquad 1 \leq i \leq m \tag{6}$$

where $t = \{1, 2, \ldots, n\}$ represents ”time” of a sequence, and m is the dimension of embedding vectors. Besides the max layer, the min and $average$ layers are also used to extract features. By concatenating all three feature vectors, we get the final feature vector $f(M)$ of the document M as follows, which can be fed to the classifier for readability assessment.

$$f(M) = [f_{max}(M), f_{min}(M), f_{avg}(M)] \tag{7}$$

4 Experiments

In this section, we conduct experiments based on four datasets of two languages, to investigate the following two research questions:

RQ1: Whether KEWE is effective for readability assessment, compared with other well-known readability assessment methods, and other word embedding models?

RQ2: What are the effects of the quality of input (i.e., the quality of the knowledge base and text corpus) and the hybrid ratio (i.e., the weighting factor λ) on the prediction performance of KEWE?

4.1 The Datasets and Knowledge Base

The experiments are conducted on four datasets, including two English datasets: ENCT and EHCT, two Chinese datasets: CPT and CPC. ENCT, CPT (Jiang et al., 2015), and EHCT are extracted from textbooks [1], where the documents have already been leveled into grades; CPC is extracted from the students' compositions [2] written by Chinese primary school students, where the documents are leveled by the six grades of authors. EHCT is collected from the English textbooks of Chinese high schools and colleges, which contains 4 levels corresponding to the 3 grades of high school plus undergraduates. The details of the four datasets are shown in Table 1.

Since the experiments are conducted on two languages, we collect the knowledge bases for both English and Chinese, which are used for extracting domain knowledge and constructing the knowledge graphs, as described in Section 3.1. The details are shown in Table 2.

[1] http://www.dzkbw.com

[2] http://www.eduxiao.com

Langauage	Dataset	#Level	#doc	#sent/doc	#word/doc	|V|	#doc in reading levels
English	ENCT	4	276	16.92	266.62	7747	[72, 96, 60, 48]
	EHCT	4	252	40.82	646.70	10485	[67, 67, 58, 60]
Chinese	CPT	6	637	25.47	448.16	19254	[96, 110, 106, 108, 113, 104]
	CPC	6	300	14.91	305.16	9459	[50, 50, 50, 50, 50, 50]

Table 1: Statistics of the Four Datasets

Perspective	Type of Material	English	Chinese
Acqusition	AoA rating	AoA norms for over 50,000 English Words	–
	In-class teaching	New word list (primary, junior, high)	New word list (primary, junior, high)
	Extra-curricular teaching	New Concept English vocabulary	Overseas Chinese Language vocabulary
	Proficiency test	CET(4,6), PETS(1,2,5) vocabulary	HSK(1-6) vocabulary
Usage	Frequency List	Word Frequency List of American English	Chinese High Frequency Word List
	Sentence Corpus	English Wikipedia	Chinese Wikipedia
Structure	Dictionary	Syllabary	Stroke dictionary

Table 2: Details of the collected knowledge bases

4.2 Experiment Settings

For readability assessment, we design experiments on the four datasets using the hold-out validation, which randomly divides a dataset into training and test sets by stratified sampling. The test ratio is set as 0.3. To reduce randomness, under each case, 100 rounds of hold-out validations are performed, and the average results are reported. To tune the hyper-parameters, we randomly choose three-tenths of the training set as development set. We choose three widely used metrics (Jiang et al., 2014): Accuracy (Acc), Adjacent Accuracy (±Acc) and Pearson's Correlation Coefficient (PCC), to measure the performance on readability assessment.

For the training of word embedding, we use all the sentences in the target dataset as the training corpus (denote as the internal corpus), to ensure sufficient word coverage. To avoid mixing the level information into the internal corpus, we shuffle the sentences in it before feeding to the model. For the hyper-parameters of text-based word embedding, we set the dimension as 300, the window size as 5, the number of negative samples as 5, and the iteration number as 5. For KEWE, the default length of random walk path is set as 80 (Grover and Leskovec, 2016). k and λ are tuned using the development set.

4.3 Comparison with Other Methods

To address RQ1, we firstly compare KEWE with the following readability assessment methods:

- SMOG (McLaughlin, 1969) and FK (Kincaid et al., 1975) are two widely-used readability formulas.
- SUM (Collins-Thompson and Callan, 2004) is a smoothed unigram model.
- HCF is the hand-crafted feature based method. For English and Chinese, we employ the state-of-the-art feature set proposed by Vajjala and Meurers (2012) and Jiang et al. (2014) respectively.
- BOW refers to the bag-of-words model.

For the feature-based methods (i.e., HCF, BOW, and KEWE), we use both Logistic Regression (LR) and Random Forest (RF) as the classifiers for readability assessment . Table 3 lists the performance measures of all these methods on four datasets. The value marked in bold in each column refers to the maximum (best) measures acquired by the methods on each dataset by certain metric.

From Table 3, we can see that the performances of readability formulas (SMOG and FK) are not good on all the datasets, except the adjacent accuracies on ENCT. The smoothed unigram model (SUM) outperforms SOMG and FK on all the datasets, and on EHCT, its accuracy is only slightly inferior to KEWE$_h$. HCF performs the best among methods other than KEWE. Even compared with KEWE, it achieves the best performance on CPC and the best adjacent accuracy on ENCT. KEWE is only slightly inferior to HCF in four columns, but outperforms all the other methods in the other eight columns.

Method		Dataset											
		ENCT			EHCT			CPT			CPC		
		Acc	±Acc	PCC	Acc	±Acc	PCC	Acc	±Acc	PCC	Acc	±Acc	PCC
SMOG		46.04	94.13	0.7441	28.47	84.05	0.5517	30.95	70.21	0.6063	19.58	60.10	0.4076
FK		50.98	97.67	0.7870	25.17	76.16	0.3417	25.15	64.61	0.4912	16.93	50.49	0.0808
SUM		66.64	97.19	0.8345	70.41	91.41	0.7257	33.61	72.04	0.5866	26.71	56.59	0.2734
HCF	LR	87.24	98.01	0.9136	53.69	89.85	0.6937	43.71	81.69	0.7652	27.08	61.54	0.4275
	RF	90.96	**100**	0.9592	60.36	91.24	0.7421	47.97	87.99	0.8159	**35.09**	**72.49**	**0.5880**
BOW	LR	81.57	99.28	0.9049	65.88	89.61	0.7257	34.82	74.67	0.6593	29.01	56.71	0.3083
	RF	78.89	94.71	0.8294	59.03	89.96	0.7384	39.56	80.73	0.7486	31.13	62.74	0.5247
KEWE$_k$	LR	91.12	99.72	0.9545	64.03	91.43	0.7745	52.69	87.35	0.8091	30.59	61.32	0.5272
	RF	92.34	99.71	0.9606	69.13	95.28	0.8251	52.94	88.39	0.8167	30.73	66.14	0.5278
KEWE$_h$	LR	93.67	99.58	0.9654	65.25	93.40	0.8129	54.33	88.03	0.8233	34.81	65.11	0.5507
	RF	**94.48**	99.72	**0.9705**	**71.20**	**96.12**	**0.8449**	**54.83**	**88.94**	**0.8287**	33.26	65.63	0.5111
HCF+KEWE$_h$	LR	87.37	98.07	0.9153	53.96	90.05	0.6976	45.92	83.53	0.7876	27.31	61.84	0.4300
	RF	**95.87**	99.89	**0.9794**	70.05	95.85	0.8380	**60.23**	**92.28**	**0.8776**	**35.52**	70.24	0.5801

Table 3: Performance comparison between KEWE and other readability assessment methods

Model	Dataset											
	ENCT			EHCT			CPT			CPC		
	Acc	±Acc	PCC	Acc	±Acc	PCC	Acc	±Acc	PCC	Acc	±Acc	PCC
Random	31.73	75.57	0.0834	24.96	61.35	0.0786	16.96	45.96	0.0590	16.81	45.59	0.0743
NNLM	79.80	98.66	0.8843	58.43	88.36	0.7046	43.47	85.22	0.7806	30.68	65.07	0.5195
C&W	82.35	99.05	0.9039	59.05	90.06	0.7160	45.59	84.70	0.7874	30.60	64.83	0.5023
GloVe	82.66	99.52	0.9129	58.15	90.53	0.7302	43.63	85.80	0.7882	30.33	64.48	0.4933
CBOW	73.94	96.86	0.8299	62.40	91.29	0.7518	40.49	82.83	0.7564	27.40	59.48	0.3929
SG	82.20	98.60	0.9001	62.73	62.90	0.7563	44.31	85.54	0.7869	31.92	63.04	0.4527
KV	89.10	98.71	0.9295	64.15	90.85	0.7483	50.63	84.84	0.7806	28.19	61.90	0.4658
SG+KV	84.71	98.81	0.9136	66.00	92.73	0.7833	46.49	86.08	0.7921	32.03	63.34	0.4729
KEWE$_k$	92.34	99.71	0.9606	69.13	95.28	0.8251	52.94	88.39	0.8167	30.73	**66.14**	**0.5278**
KEWE$_h$	**94.48**	**99.72**	**0.9705**	**71.20**	**96.12**	**0.8449**	**54.83**	**88.94**	**0.8287**	**33.26**	65.63	0.5111

Table 4: Performance comparison between KEWE and other word embedding models

Overall, KEWE is competitive for readability assessment. In addition, by combining KEWE with the hand-crafted feature set of HCF, the performance can be further improved in many columns.

Secondly, we compare KEWE with other text-based word embedding models for readability assessment. These models include NNLM (Bengio et al., 2003), C&W (Collobert and Weston, 2008), GloVe (Pennington et al., 2014), CBOW and Skip-Gram (SG) (Mikolov et al., 2013). Random embedding (Random) and knowledge vectors from the three perspectives (KV) are also used as baselines. All the text-based models mentioned are trained on the four datasets respectively. Table 4 lists the results of applying word embedding for readability assessment using Random Forest as the classifier. From Table 4, the model KEWE$_h$ gets the best performance among all the embedding baselines, including SG+KV, which also takes in both knowledge and text corpus. The results demonstrate the superiority of hybridizing the knowledge graph and text corpus to learn the representation of words.

4.4 Model Analysis

To address RQ2, we analyze the effects of the knowledge graph, the text corpus, and the hybrid ratio on the performance of KEWE. Firstly, we study the impacts of the knowledge graph on the performance of KEWE$_k$. Specifically, we study the following three parameters of the knowledge graph: the knowledge vectors generated from the three word-level difficulties (i.e., acquisition difficulty, usage difficulty, and structure difficulty), the similarity function (i.e., $sim()$), and the number of neighbors of each node in the knowledge graph (i.e., k). Figure 2 shows the performance measures of using each of the three knowledge vectors respectively in bar chart. It can be found that the acquisition difficulty outperforms the other two on all four datasets. By combining the three knowledge vectors, the performance can be further improved. Figure 3 shows the performance measures of using different similarity functions and different values of k in line charts. It can be found that the knowledge graphs achieve a relatively good performance, when the cosine similarity is used and k is close to 50 or 100.

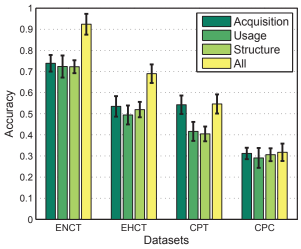

Figure 2: The performance of KEWE$_k$ using different types of knowledge vectors

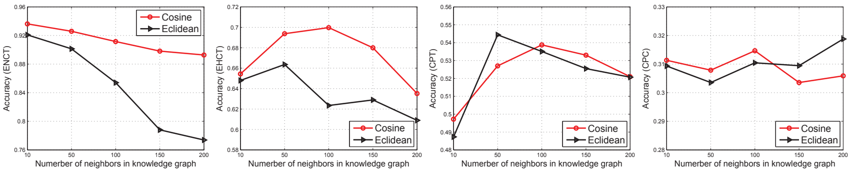

Figure 3: The performance of KEWE$_k$ using knowledge graphs with different similarity functions and neighbor numbers

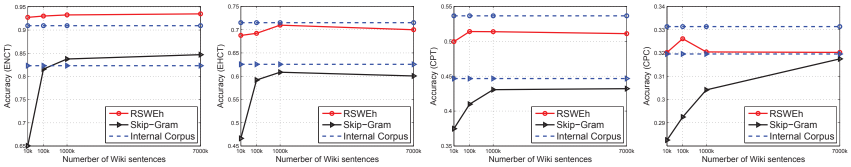

Figure 4: The performance of KEWE$_h$ using different volumes of external corpus

Secondly, we study the impacts of using external corpus on the performance of KEWE$_h$. To learn text-based word embedding for both languages, we collect the external corpus from both English Wikipedia (Ewiki) and Chinese Wikipedia (Cwiki). After preprocessing, Ewiki contains $7M$ sentences and $172M$ words, and Cwiki contains $7.4M$ sentences and $160M$ words. We sample different number of sentences from the corpus for training word embedding, and then measure the performance of KEWE$_h$ on readability assessment. Figure 4 shows the performance measures by using different volumes of external corpus in line charts, with Skip-Gram trained for comparison. Both models are also trained using the internal corpus, and the performance measures (dotted lines) are depicted as baselines. From Figure 4, it can be found that the performance of both Skip-Gram and KEWE$_h$ increases as the volume of external corpus increases, and keeps stable when the corpus is large enough. Besides, on English datasets, word embedding trained using external corpus achieves comparable performance with that using internal corpus. The above suggests that external corpus is a good substitution for the internal corpus, especially on English datasets, and a relative large volume of corpus is required to achieve stable performance.

Finally, we study the impacts of the weighting factor λ on the performance of KEWE$_h$. Since λ is approximated by $N_1/(N_1 + N_2)$ in our model, we vary λ from 0 to 1 by setting $N_1 + N_2 = 100$ and then varying N_1 from 0 to 100 stepping by 10. Figure 5 shows the performance of KEWE$_h$ with varied λ in line charts with error bars, where $\lambda = 0$ and 1 correspond to Skip-Gram and KEWE$_k$ respectively. From Figure 5, it can be found that KEWE$_h$ can outperform its base models (i.e., KEWE$_k$ and Skip-Gram), by setting λ with a suitable value near the base model which performs better. However, it requires further study to find a suitable λ for training KEWE$_h$.

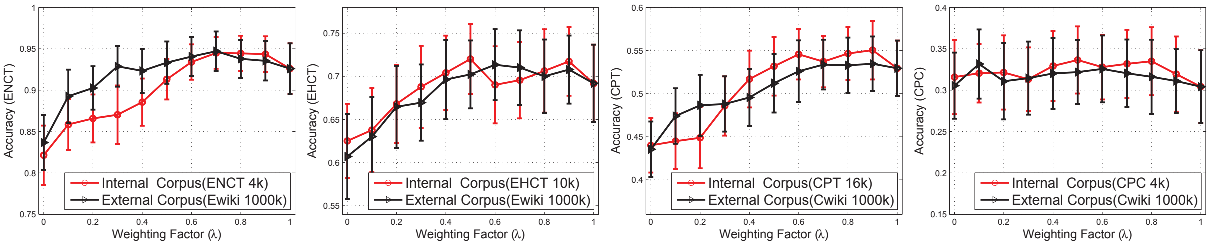

Figure 5: The performance of KEWE$_h$ with varied weighting factor λ

4.5 Error Analysis

To better understand our method, we perform error analysis on the classification results. We mainly describe two sources of errors: word representation failure and word order capturing failure. From the perspective of word representation, the classification error is caused by the fact that the difficulty information may not be properly encoded into the word embedding. Table 4 shows that KEWE performs better than the classical text-based word embeddings in encoding difficulty information into representation, but in the final results there still exists the failure of word representation in KEWE. From the perspective of word order capturing, the classification error is caused by the fact that the order among words may be neglected, so that the syntactic difficulty, pragmatic difficulty, and discourse difficulty of documents are ignored during the process of readability assessment. Table 3 shows that KEWE$_h$ can be further improved by combining with the features related to the word order (i.e., HCF+KEWE$_h$), which means that there exists the failure of word order capturing in KEWE. These two kinds of error sources reveal the limitation of our method. In future work, the neural network accompanied with word embedding is a good alternative, which can produce better representation of documents.

5 Conclusion

In this paper, we propose the knowledge-enriched word embedding (KEWE) for readability assessment. We extract the domain knowledge on word-level difficulty from three different perspectives and construct a knowledge graph. Based on the difficulty context derived from the knowledge graph, we develop two word embedding models (i.e., KEWE$_k$ and KEWE$_h$). The experimental results on English and Chinese datasets demonstrate that KEWE can outperform other well-known readability assessment methods, and the classic text-based word embedding models. Future work is planned to involve extra datasets and additional word embedding strategies so that the soundness of KEWE can be further approved.

Acknowledgements

This work is supported by the National Key R&D Program of China under Grant No. 2018YFB1003800; National Natural Science Foundation of China under Grant Nos. 61373012, 61321491, 91218302; the Fundamental Research Funds for the Central Universities under Grant No. 020214380049. This work is partially supported by the Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

Dimitrios Alikaniotis, Helen Yannakoudakis, and Marek Rei. 2016. Automatic text scoring using neural networks. In *Proceedings of the 54th Annual Meeting of the Association for ComputationalLinguistics*, pages 715–725.

Mikhail Belkin and Partha Niyogi. 2001. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in Neural Information Processing Systems*, 14(6).

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Marc Brysbaert and Andrew Biemiller. 2016. Test-based age-of-acquisition norms for 44 thousand english word meanings. *Behavior Research Methods*, pages 1–4.

Miriam Cha, Youngjune Gwon, and H. T. Kung. 2017. Language modeling by clustering with word embeddings for text readability assessment. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017*, pages 2003–2006.

Jeanne Sternlicht Chall. 1995. *Readability revisited: The new Dale-Chall readability formula*, volume 118. Brookline Books Cambridge, MA.

Kevyn Collins-Thompson and James P Callan. 2004. A language modeling approach to predicting reading difficulty. In *Proceedings of the 2004 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 193–200.

Kevyn Collinsthompson. 2014. Computational assessment of text readability: A survey of current and future research. *International Journal of Applied Linguistics*, 165(2):97–135.

R. Collobert and J. Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference(ICML 2008)*, pages 160–167.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Edgar Dale and Jeanne S. Chall. 1948. A formula for predicting readability. *Educational Research Bulletin*, 27(1):11–28.

Scott Deerwester. 1990. Indexing by latent semantic analysis. *Journal of the Association for Information Science & Technology*, 41(6):391–407.

Lijun Feng, Martin Jansche, Matt Huenerfauth, and Noémie Elhadad. 2010. A comparison of features for automatic readability assessment. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 276–284.

Rudolph Flesch. 1948. A new readability yardstick. *Journal of applied psychology*, 32(3):221.

Thomas François and Cédrick Fairon. 2012. An ai readability formula for french as a foreign language. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 466–477.

Itziar Gonzalez-Dios, Marıa Jesús Aranzabe, Arantza Dıaz de Ilarraza, and Haritz Salaberri. 2014. Simple or complex? assessing the readability of basque texts. In *Proceedings of the 25th International Conference on Computational Linguistics*, pages 334–344.

Palash Goyal and Emilio Ferrara. 2017. Graph embedding techniques, applications, and performance: A survey. *arXiv preprint arXiv:1705.02801*.

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864.

Robert Gunning. 1952. *The technique of clear writing*. McGraw-Hill.

Julia Hancke, Sowmya Vajjala, and Detmar Meurers. 2012. Readability classification for german using lexical, syntactic, and morphological features. In *Proceedings of the 24th International Conference on Computational Linguistics*, pages 1063–1080.

Michael Heilman, Collins-Thompson, Jamie Callan, and Maxine Eskenazi. 2007. Combining lexical and grammatical features to improve readability measures for first and second language texts. In *Proceedings of NAACL HLT*, pages 460–467.

Michael Heilman, Kevyn Collins-Thompson, and Maxine Eskenazi. 2008. An analysis of statistical models and features for reading difficulty prediction. In *Proceedings of the Third Workshop on Innovative Use of NLP for Building Educational Applications*, pages 71–79.

Zhiwei Jiang, Gang Sun, Qing Gu, and Daoxu Chen. 2014. An ordinal multi-class classification method for readability assessment of chinese documents. In *Knowledge Science, Engineering and Management*, pages 61–72. Springer.

Zhiwei Jiang, Gang Sun, Qing Gu, Tao Bai, and Daoxu Chen. 2015. A graph-based readability assessment method using word coupling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 411–420. Association for Computational Linguistics.

Rohit J Kate, Xiaoqiang Luo, Siddharth Patwardhan, Martin Franz, Radu Florian, Raymond J Mooney, Salim Roukos, and Chris Welty. 2010. Learning to predict readability using diverse linguistic features. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 546–554.

Paul Kidwell, Guy Lebanon, and Kevyn Collins-Thompson. 2009. Statistical estimation of word acquisition with application to readability prediction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 900–909.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Naval Air Station, Memphis, TN.

Omer Levy and Yoav Goldberg. 2014a. Dependency-based word embeddings. In *Meeting of the Association for Computational Linguistics*, pages 302–308.

Omer Levy and Yoav Goldberg. 2014b. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, pages 2177–2185.

Quan Liu, Hui Jiang, Si Wei, Zhen-Hua Ling, and Yu Hu. 2015. Learning semantic word embeddings based on ordinal knowledge constraints. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pages 1501–1511.

Yi Ma, Eric Fosler-Lussier, and Robert Lofthus. 2012. Ranking-based readability assessment for early primary children's literature. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 548–552.

G Harry McLaughlin. 1969. Smog grading: A new readability formula. *Journal of reading*, 12(8):639–646.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710.

Emily Pitler and Ani Nenkova. 2008. Revisiting readability: A unified framework for predicting text quality. In *Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 186–195.

Yafeng Ren, Yue Zhang, Meishan Zhang, and Donghong Ji. 2016. Improving twitter sentiment classification using topic-enriched multi-prototype word embeddings. In *Thirtieth AAAI Conference on Artificial Intelligence*, pages 3038–3044.

Sam T. Roweis and Lawrence K. Saul. 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–6.

Elliot Schumacher, Maxine Eskenazi, Gwen Frishkoff, and Kevyn Collins-Thompson. 2016. Predicting the relative difficulty of single sentences with and without surrounding context. In *Conference on Empirical Methods in Natural Language Processing*, pages 1871–1881.

Sarah E Schwarm and Mari Ostendorf. 2005. Reading level assessment using support vector machines and statistical language models. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 523–530.

Jie Shen and Cong Liu. 2016. Improved word embeddings with implicit structure information. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference*, pages 2408–2417.

Luo Si and James P. Callan. 2001. A statistical model for scientific readability. In *Proceedings of the 2001 ACM CIKM International Conference on Information and Knowledge Management*, pages 574–576.

Manjira Sinha, Tirthankar Dasgupta, and Anupam Basu. 2014. Influence of target reader background and text features on text readability in bangla: A computational approach. In *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference*, pages 345–354.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *Meeting of the Association for Computational Linguistics*, pages 1555–1565.

Amalia Todirascu, Thomas François, Delphine Bernhard, Nuria Gala, and Anne-Laure Ligozat. 2016. Are cohesive features relevant for text readability evaluation? In *26th International Conference on Computational Linguistics (COLING 2016)*, pages 987–997.

Peter D Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37:141–188.

Sowmya Vajjala and Detmar Meurers. 2012. 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, BEA@NAACL-HLT 2012, June 7, 2012, Montréal, Canada*, pages 163–173.

Shuhan Wang and Erik Andersen. 2016. Grammatical templates: Improving text difficulty evaluation for language learners. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference*, pages 1692–1702.

Daixin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural deep network embedding. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1225–1234.

Changxing Wu, Xiaodong Shi, Yidong Chen, Jinsong Su, and Boli Wang. 2017. Improving implicit discourse relation recognition with discourse-specific word embeddings. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 269–274.

Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. 2016. Revisiting semi-supervised learning with graph embeddings. pages 40–48.