Semi-Supervised Joint Estimation of Word and Document Readability

Yoshinari Fujinuma
University of Colorado Boulder
fujinumay@gmail.com

Masato Hagiwara
Octanove Labs
masato@octanove.com

Abstract

Readability or difficulty estimation of words and documents has been investigated independently in the literature, often assuming the existence of extensive annotated resources for the other. Motivated by our analysis showing that there is a recursive relationship between word and document difficulty, we propose to jointly estimate word and document difficulty through a graph convolutional network (GCN) in a semi-supervised fashion. Our experimental results reveal that the GCN-based method can achieve higher accuracy than strong baselines, and stays robust even with a smaller amount of labeled data.[1]

1 Introduction

Accurately estimating the readability or difficulty of words and text has been an important fundamental task in NLP and education, with a wide range of applications including reading resource suggestion (Heilman et al., 2008), text simplification (Yimam et al., 2018), and automated essay scoring (Vajjala and Rama, 2018).

A number of linguistic resources have been created either manually or semi-automatically for non-native learners of languages such as English (Capel, 2010, 2012), French (François et al., 2014), and Swedish (François et al., 2016; Alfter and Volodina, 2018), often referencing the Common European Framework of Reference (Council of Europe, 2001, CEFR). However, few linguistic resources exist outside these major European languages and manually constructing such resources demands linguistic expertise and efforts.

This led to the proliferation of NLP-based *readability* or *difficulty assessment* methods to automatically estimate the difficulty of words and texts (Vajjala and Meurers, 2012; Wang and Andersen, 2016; Alfter and Volodina, 2018; Vajjala and Rama, 2018; Settles et al., 2020). However, bootstrapping lexical resources with difficulty information often assumes the existence of textual datasets (e.g., digitized coursebooks) annotated with difficulty. Similarly, many text readability estimation methods (Wang and Andersen, 2016; Xia et al., 2016) assume the existence of abundant lexical or grammatical resources annotated with difficulty information. Individual research studies focus only on one side, either words or texts, although in reality they are closely intertwined—there is a *recursive relationship between word and text difficulty*, where the difficulty of a word is correlated to the *minimum* difficulty of the document where that word appears, and the difficulty of a document is correlated to the *maximum* difficulty of a word in that document (Figure 2).

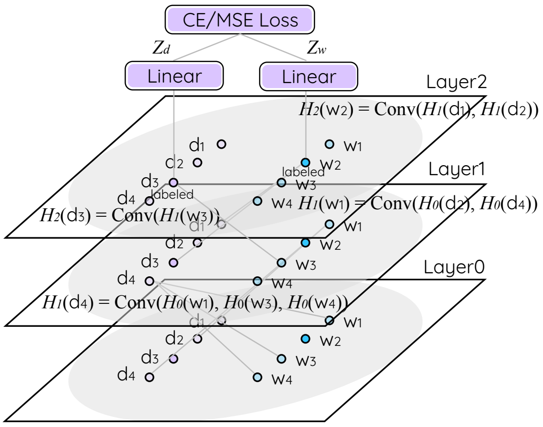

Figure 1: Overview of the proposed GCN architecture which recursively connects word w_i and document d_j to exploit the recursive relationship of their difficulty.

We propose a method to jointly estimate word and text readability in a semi-supervised fashion from a smaller number of labeled data by leveraging the recursive relationship between words and documents. Specifically, we leverage recent developments in graph convolutional networks (Kipf and Welling, 2017, GCNs) and predict the difficulty of

[1] Our code is at https://github.com/akkikiki/diff_joint_estimate

words and documents simultaneously by modeling those as nodes in a graph structure and recursively inferring their embeddings using the convolutional layers (Figure 1). Our model leverages not only the supervision signals but also the recursive nature of word-document relationship. The contributions of this paper are two fold:

- We reframe the word and document readability estimation task as a semi-supervised, joint estimation problem motivated by their recursive relationship of difficulty.
- We show that GCNs are effective for solving this by exploiting unlabeled data effectively, even when less labeled data is available.

2 Task Definition

Given a set of words $\mathcal{W}$ and documents $\mathcal{D}$, the goal of the joint readability estimation task is to find a function f that maps both words and documents to their difficulty label $f : \mathcal{W} \cup \mathcal{D} \to Y$. Documents here can be text of an arbitrary length, although we use paragraphs as the basic unit of prediction. This task can be solved as a classification problem or a regression problem where $Y \in \mathbb{R}$. We use six CEFR-labels representing six levels of difficulty, such as $Y \in \{$A1 (lowest), A2, B1, B2, C1, C2 (highest)$\}$ for classification, and a real-valued readability estimate $\beta \in \mathbb{R}$ inspired by the item response theory (Lord, 1980, IRT) for regression[2]. The β for each six CEFR level are A1$= -1.38$, A2$= -0.67$, B1$= -0.21$, B2$= 0.21$, C1$= 0.67$, and C2$= 1.38$.

Words and documents consist of mutually exclusive unlabeled subsets $\mathcal{W}_U$ and $\mathcal{D}_U$ and labeled subsets $\mathcal{W}_L$ and $\mathcal{D}_L$. The function f is inferred using the supervision signal from $\mathcal{W}_L$ and $\mathcal{D}_L$, and potentially other signals from $\mathcal{W}_U$ and $\mathcal{D}_U$ (e.g., relationship between words and documents).

3 Exploiting Recursive Relationship by Graph Convolutional Networks

We first show how the readability of words and documents are recursively related to each other. We then introduce a method based on graph convolutional networks (GCN) to capture such relationship.

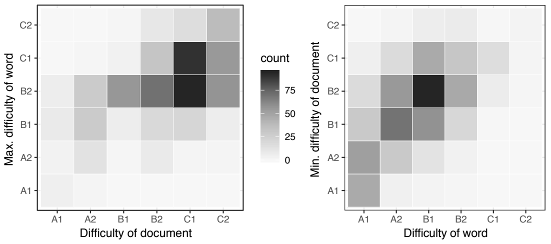

Figure 2: Recursive relationship of word/document difficulty. Word difficulty is correlated to the *minimum* difficulty of the document where that word appears, and document difficulty is correlated to the *maximum* difficulty of a word in that document.

3.1 Recursive Relationship of Word and Document Difficulty

The motivation of using a graph-based method for difficulty classification is the recursive relationship of word and document difficulty. Figure 2 shows such recursive relationship using the difficulty-labeled datasets explained in Section 5. One insight here is the strong correlation between the difficulty of a document and *the maximum difficulty of a word in that document*. This is intuitive and shares motivation with a method which exploits hierarchical structure of a document (Yang et al., 2016). However, the key insight here is the strong correlation between the difficulty of a word and *the minimum difficulty of a document where that word appears*, indicating that the readability of words informs that of documents, and vise versa.

3.2 Graph Convolutional Networks on Word-Document Graph

To capture the recursive, potentially nonlinear relationship between word and document readability while leveraging supervision signals and features, we propose to use graph convolutional networks (Kipf and Welling, 2017, GCNs) specifically built for text classification (Yao et al., 2019), which treats words and documents as nodes. Intuitively, the hidden layers in GCN, which recursively connects word and document nodes, encourage exploiting the recursive word-document relationship.

Given a heterogeneous word-document graph $G = (V, E)$ and its adjacency matrix $A \in \mathbb{R}^{|V|\times|V|}$, the hidden states for each layer $H_n \in \mathbb{R}^{|V|\times h_n}$ in a GCN with N hidden layers is com-

[2] We assumed the difficulty estimate β is normally distributed and used the mid-point of six equal portions of $N(0, 1)$ when mapping CEFR levels to β.

puted using the previous layer H_{n-1} as:

$$H_n = \sigma(\tilde{A} H_{n-1} W_n) \tag{1}$$

where σ is the ReLU function[3], $\tilde{A} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ i.e., a symmetrically normalized matrix of A with its degree matrix D, and $W_n \in \mathbb{R}^{h_{n-1} \times h_n}$ is the weight matrix for the nth layer. The input to the first layer H_1 is $H_0 = X$ where $X \in \mathbb{R}^{|V| \times h_0}$ is the feature matrix with h_0 dimensions for each node in V. We use three different edge weights following Yao et al. (2019): (1) $A_{ij} = \text{tfidf}_{ij}$ if i is a document and j is a word, (2) the normalized point-wise mutual information (PMI) i.e., $A_{ij} = \text{PMI}(i, j)$ if both i and j are words, and (3) self-loops, i.e., $A_{ii} = 1$ for all i.

We now describe the components which differs from Yao et al. (2019). We use separate final linear layers for words and documents[4]:

$$\begin{aligned} Z_w &= H_N W_w + b_w && (2) \\ Z_d &= H_N W_d + b_d && (3) \end{aligned}$$

where W and b are the weight and bias of the layer, and used a linear combination of word and document losses weighted by α (Figure 1)

$$\mathcal{L} = \alpha \mathcal{L}(Z_w) + (1 - \alpha) \mathcal{L}(Z_d) \tag{4}$$

For regression, we used Z (Z_w for words and Z_d for documents) as the prediction of node v and used the mean squared error (MSE):

$$\mathcal{L}(Z) = \frac{1}{|V_L|} \sum_{v \in V_L} (Z_v - Y_v)^2 \tag{5}$$

where $V_L = \mathcal{W}_L \cup \mathcal{D}_L$ is the set of labeled nodes. For classification, we use a softmax layer followed by a cross-entropy (CE) loss:

$$\mathcal{L}(Z) = -\sum_{v \in V_L} \log \frac{\exp(Z_{v,Y_v})}{\sum_i \exp(Z_{v,i})}. \tag{6}$$

Since GCN is transductive, node set V also includes the unlabeled nodes from the evaluation sets and have predicted difficulty labels assigned when training is finished.

Dataset	Train	Dev	Test
Words (CEFR-J + C1/C2)	2,043	447	389
Documents (Cambridge + A1)	482	103	98

Table 1: Dataset size for words and documents

4 Experiments

Datasets We use publicly available English CEFR-annotated resources for second language learners, such as CEFR-J (Negishi et al., 2013) Vocabulary Profile as words and Cambridge English Readability Dataset (Xia et al., 2016) as documents (Table 1). Since these two datasets lack C1/C2-level words and A1 documents, we hired a linguistic PhD to write these missing portions[5].

Baselines We compare our method against methods used in previous work (Feng et al., 2010; Vajjala and Meurers, 2012; Martinc et al., 2019; Deutsch et al., 2020): (1) logistic regression for classification (LR cls), (2) linear regression for regression (LR regr), (3) Gradient Boosted Decision Tree (GBDT), and (4) Hierarchical Attention Network (Yang et al., 2016, HAN), which is reported as one of the state-of-the-art methods in readability assessment for documents (Martinc et al., 2019; Deutsch et al., 2020).

Features For all methods except for HAN, we use both surface or "traditional" (Vajjala and Meurers, 2012) and embedding features on words and documents which are shown to be effective for readability estimation (Culligan, 2015; Settles et al., 2020; Deutsch et al., 2020). For words, we use their length (in characters), the log frequency in Wikipedia (Ginter et al., 2017), and GloVe (Pennington et al., 2014). For documents, we use the number of NLTK (Loper and Bird, 2002)-tokenized words in a document, and the output of embeddings from BERT-base model (Devlin et al., 2019) which are averaged over all tokens in a given sentence.

Hyperparameters We conduct random hyperparameter search with 200 samples, separately selecting two different sets of hyperparameters, one optimized for word difficulty and the other for document. We set the number of hidden layers $N = 2$ with $h_n = 512$ for documents and $N = 1$ with $h_n = 64$ for words. See Appendix A for the details on other hyperparameters.

[3] A simplified version of GCN with linear layers (Wu et al., 2019) in preliminary experiments shows that hidden layers with ReLU performed better.

[4] A model variant with a common linear layer (i.e., original GCN) for both words and documents did not perform as well.

[5] The dataset is available at `https://github.com/openlanguageprofiles/olp-en-cefrj`.