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).

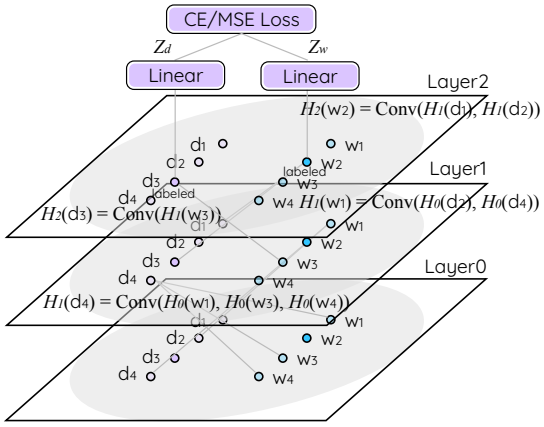

Figure 1: Overview of the proposed GCN architecture which recursively connects word w_i and document d_j to exploit the recursive relationship of their difficulty.

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

Proceedings of the Fifteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-15), pages 150–155

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 $f : \mathcal{W} \cup \mathcal{D} \rightarrow Y$. 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.

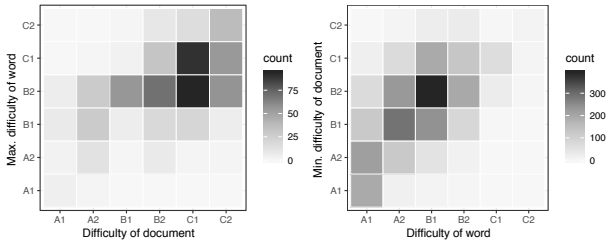

Figure 2: Recursive relationship of word/document difficulty. Word difficulty is correlated to the *minimum* difficulty of the document where that word appears, and document difficulty is correlated to the *maximum* difficulty of a word in that document.

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:

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

where σ is the ReLU function[3], $\tilde{A} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ 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]:

$$Z_w = H_N W_w + b_w \quad (2)$$
$$Z_d = H_N W_d + b_d \quad (3)$$

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)

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

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):

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

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:

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

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.

[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.

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.

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

Method	Word Acc	Word Corr	Document Acc	Document Corr
HAN	-	-	0.367	0.498
LR (regr)	0.409	0.534	0.480	0.657
LR (cls+m)	0.440	0.514	0.765	0.723
LR (cls+w)	0.440	0.540	0.765	0.880
GBDT	0.432	0.376	0.765	0.833
GCN (regr)	0.434	0.579	0.643	0.849
GCN (cls+m)	**0.476**	0.536	**0.796**	0.878
GCN (cls+w)	**0.476**	**0.592**	**0.796**	**0.891**

Table 2: Difficulty estimation results in accuracy (Acc) and correlation (Corr) on classification outputs converted to continuous values by taking the max (cls+m) or weighted sum (cls+w) and regression (regr) variants for the logistic regression (LR) and GCN.

Features	Word Acc	Word Corr	Document Acc	Document Corr
All	0.476	0.592	0.796	0.891
−word freq.	0.476	0.591	0.796	0.899
−doc length	0.481	0.601	0.796	0.890
−GloVe	0.463	0.545	0.714	0.878
−BERT	0.450	0.547	0.684	0.830
None	0.440	0.436	0.520	0.669

Table 3: Ablation study on the features used. "None" is when applying GCN without any features ($X = I$ i.e., one-hot encoding per node), which solely relies on the word-document structure of the graph.

Evaluation We use accuracy and Spearman's rank correlation as the metrics. When calculating the correlation for a classification model, we convert the discrete outputs into continuous values in two ways: (1) convert the CEFR label with the maximum probability into corresponding β in Section 2, (cls+m), or (2) take a sum of all β in six labels weighted by their probabilities (cls+w).

4.1 Results

Table 2 shows the test accuracy and correlation results. GCNs show increase in both document accuracy and word accuracy compared to the baseline. We infer that this is because GCN is good at capturing the relationship between words and documents. For example, the labeled training documents include an A1 document and that contains the word "bicycle," and the difficulty label of the document is explicitly propagated to the "bicycle" word node, whereas the logistic regression baseline mistakenly predicts as A2-level, since it relies solely on the input features to capture its similarities.

4.2 Ablation Study on Features

Table 3 shows the ablation study on the features explained in Section 4. By comparing Table 2 and Table 3, which are experimented on the same datasets, GCN without using any traditional or embedding features ("None") shows comparative results to some baselines, especially on word-level accuracy. Therefore, the structure of the word-document graph provides effective and complementary signal for readability estimation.

Overall, the BERT embedding is a powerful feature for predicting document readability on Cambridge Readabilty Dataset. Ablating the BERT embeddings (Table 3) significantly decreases the document accuracy (-0.112) which is consistent with the previous work (Martinc et al., 2019; Deutsch et al., 2020) that BERT being one of the best-performing method for predicting document readability on one of the datasets they used, and HAN performing relatively low due to not using the BERT embeddings.

4.3 Training on Less Labeled Data

To analyze whether GCN is robust when training dataset is small, we compare the baseline and GCN by varying the amount of labeled training data. In Figure 3, we observe consistent improvement in GCN over the baseline especially in word accuracy. This outcome suggests that the performance of GCN stays robust even with smaller training data by exploiting the signals gained from the recursive word-document relationship and their structure. Another trend observed in Figure 3 is the larger gap in word accuracy compared to document accuracy when the training data is small likely due to GCN explicitly using context given by word-document edges.

5 Conclusion

In this paper, we proposed a GCN-based method to jointly estimate the readability on both words and documents. We experimentally showed that GCN achieves higher accuracy by capturing the recursive difficulty relationship between words and documents, even when using a smaller amount of labeled data. GCNs are a versatile framework that allows inclusion of diverse types of nodes, such as

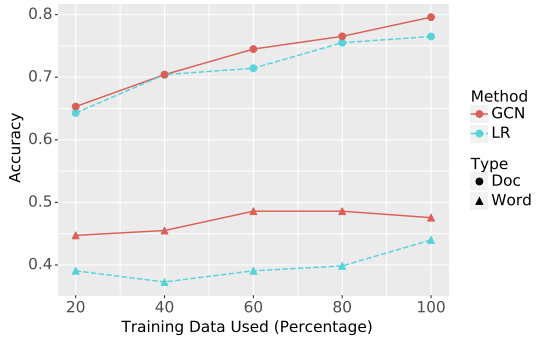

Figure 3: Word and document accuracy with different amount of training data used.

subwords, paragraphs, and even grammatical concepts. We leave this investigation as future work.

Acknowledgements

The authors would like to thank Adam Wiemerslage, Michael J. Paul, and anonymous reviewers for their detailed and constructive feedback. We also thank Kathleen Hall for her help with annotation.

References

David Alfter and Elena Volodina. 2018. Towards single word lexical complexity prediction. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*.

Annette Capel. 2010. A1–B2 vocabulary: insights and issues arising from the English Profile Wordlists project. *English Profile Journal*, 1.

Annette Capel. 2012. Completing the english vocabulary profile: C1 and C2 vocabulary. *English Profile Journal*, 3.

Council of Europe. 2001. *Common European Framework of Reference for Languages: Learning, Teaching, Assessment*. Press Syndicate of the University of Cambridge.

Brent Culligan. 2015. A comparison of three test formats to assess word difficulty. *Language Testing*, 32(4):503–520.

Tovly Deutsch, Masoud Jasbi, and Stuart Shieber. 2020. Linguistic features for readability assessment. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Lijun Feng, Martin Jansche, Matt Huenerfauth, and Noémie Elhadad. 2010. A comparison of features for automatic readability assessment.

Thomas François, Nùria Gala, Patrick Watrin, and Cédrick Fairon. 2014. FLELex: a graded lexical resource for French foreign learners. In *Proceedings of the Language Resources and Evaluation Conference*.

Thomas François, Elena Volodina, Ildikó Pilán, and Anaïs Tack. 2016. SVALex: a CEFR-graded lexical resource for Swedish foreign and second language learners. In *Proceedings of the Language Resources and Evaluation Conference*.

Filip Ginter, Jan Hajič, Juhani Luotolahti, Milan Straka, and Daniel Zeman. 2017. CoNLL 2017 shared task - automatically annotated raw texts and word embeddings. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Michael Heilman, Le Zhao, Juan Pino, and Maxine Eskenazi. 2008. Retrieval of reading materials for vocabulary and reading practice. In *Proceedings of the Third Workshop on Innovative Use of NLP for Building Educational Applications*, pages 80–88.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of the International Conference on Learning Representations*.

Edward Loper and Steven Bird. 2002. NLTK: The natural language toolkit. In *Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*.

Frederic M. Lord. 1980. *Applications of Item Response Theory To Practical Testing Problems*. Lawrence Erlbaum Associates.

Matej Martinc, Senja Pollak, and Marko Robnik-Sikonja. 2019. Supervised and unsupervised neural approaches to text readability. *CoRR*, abs/1907.11779.

Masashi Negishi, Tomoko Takada, and Yukio Tono. 2013. A progress report on the development of the CEFR-J. In *Exploring language frameworks: Proceedings of the ALTE Kraków Conference*, pages 135–163.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of Empirical Methods in Natural Language Processing*.

Burr Settles, Geoffrey T. LaFlair, and Masato Hagiwara. 2020. Machine learning–driven language assessment. *Transactions of the Association for Computational Linguistics*, 8:247–263.

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*.

Sowmya Vajjala and Taraka Rama. 2018. Experiments with universal CEFR classification. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*.

Shuhan Wang and Erik Andersen. 2016. Grammatical templates: Improving text difficulty evaluation for language learners.

Felix Wu, Tianyi Zhang, Amauri Holanda de Souza Jr., Christopher Fifty, Tao Yu, and Kilian Q. Weinberger. 2019. Simplifying graph convolutional networks. In *Proceedings of the International Conference of Machine Learning*.

Menglin Xia, Ekaterina Kochmar, and Ted Briscoe. 2016. Text readability assessment for second language learners. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Association for the Advancement of Artificial Intelligence*.

Seid Muhie Yimam, Chris Biemann, Shervin Malmasi, Gustavo Paetzold, Lucia Specia, Sanja Štajner, Anaïs Tack, and Marcos Zampieri. 2018. A report on the complex word identification shared task 2018. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*.

A Hyperparameter Details

We conduct random hyperparameter search with 200 samples in the following ranges: $\alpha \in \{0.1, 0.2, ..., 0.9\}$, the learning rate from $\{1, 2, 5, 10, 20, 50, 100\} \times 10^{-4}$, dropout probability from $\{0.1, 0.2, ..., 0.5\}$, the number of epochs from $\{250, 500, 1000, 1500, 2000\}$, the number of hidden units $h_n \in \{32, 64, 128, 256, 512, 1024\}$, the number of hidden layers from $\{1, 2, 3\}$, and the PMI window width from $\{\text{disabled}, 5, 10, 15, 20\}$.

We now describe the selected best combination of hyperparameters for each setting. For GCN in the classification setting, the selected hyperparameters for document difficulty estimation are:

- α: 0.3
- Learning rate: $5 \cdot 10^{-4}$
- Dropout probability: 0.5
- The number of epochs: 500
- The number of hidden units h_n: 512
- The number of hidden layers N: 2
- PMI window width: 5

and for word difficulty estimation, the selected hyperparameters are:

- α: 0.2
- Learning rate: $5 \cdot 10^{-3}$
- Dropout probability: 0.2
- The number of epochs: 250
- The number of hidden units h_n: 64
- The number of hidden layers N: 1
- PMI window width: disabled

For GCN in the regression setting, the selected hyperparameters for document difficulty estimation are:

- α: 0.4
- Learning rate: $2 \cdot 10^{-4}$
- Dropout probability: 0.3
- The number of epochs: 1500
- The number of hidden units h_n: 128
- The number of hidden layers N: 2
- PMI window width: 5

and for word difficulty estimation, the selected hyperparameters are:

- α: 0.2
- Learning rate: $1 \cdot 10^{-3}$
- Dropout probability: 0.1
- The number of epochs: 500
- The number of hidden units h_n: 512
- The number of hidden layers N: 2
- PMI window width: disabled