# Word Embedding-Based Text Complexity Analysis

Kahyun Choi[1][0000−0003−4854−7104]

Indiana University Bloomington
{choika}@iu.edu

**Abstract.** Text complexity metrics serve crucial roles in quantifying the readability level of important documents, leading to ensuring public safety, enhancing educational outcomes, and more. Pointwise mutual information (PMI) has been widely used to measure text complexity by capturing the statistical co-occurrence patterns between word pairs, assuming their semantic significance. However, we observed that word embeddings are similar to PMI in that both are based on co-occurrence in large corpora. Yet, word embeddings are superior in terms of faster calculations and more generalizable semantic proximity measures. Given this, we propose a novel text complexity metric that leverages the power of word embeddings to measure the semantic distance between words in a document. We empirically validate our approach by analyzing the OneStopEnglish dataset, which contains news articles annotated with expert-labeled readability scores. Our experiments reveal that the proposed word embedding-based metric demonstrates a stronger correlation with ground-truth readability levels than conventional PMI-based metrics. This study serves as a cornerstone for future research aiming to incorporate context-dependent embeddings and extends applicability to various text types.

**Keywords:** Readability · Text complexity · Word embedding · Pointwise mutual information.

## 1 Introduction

Computational analysis of text complexity has been actively explored over a century, which produced many readability metrics and algorithms [2]. Readability metrics have served a wide range of purposes; it is critical to ensure the safety of people by providing the appropriate levels of military manuals or medical documents [2][17]; students can also benefit from the right level of reading materials for a more engaging reading experience and a better learning outcome [6][12]; many writers use readability metrics from online text analytic services, such as `https://readable.com`, `https://textinspector.com`, and `https://app.grammarly.com`, to increase their readerships by providing more readable content.

Computational metrics for text complexity have a long history of evolution. Traditional readability metrics, for example, Flesch reading ease, Flesch-Kincaid

grade, Gunning Fog index, and Coleman-Liau index, were introduced decades ago [2]. Their basic principle is that short, simple, and familiar words and sentences make text easy to read, and vice versa. The final scores are based on straightforward linguistic statistics, such as the number of characters, syllables, words, familiar words, and sentences. More recently, sophisticated word-level features have been developed to quantify concreteness and lexical cohesion, and structural features of words, e.g., syntactic complexity [11]. In addition, advanced natural language processing (NLP) technology provides a new perspective to text complexity metrics by using big text corpora and data-driven machine learning techniques [5].

In this vein, Flor et al. advanced the field by introducing a text complexity metric based on pointwise mutual information (PMI), a measure rooted in the co-occurrence of words [4]. Their examples highlighted how PMI can effectively capture semantic distances between words, thereby contributing to the overall complexity of the text. For instance, in a sentence like "The dog barked and wagged its tail," the PMI score would be relatively high (PMI=5.5) due to the frequent co-occurrence of word pairs such as "dog" and "bark" or "wag" and "tail." Conversely, a sentence like 'Green ideas sleep furiously' would yield a lower PMI score (PMI=2.2), attributed to the rare co-occurrence of pairs like "green" and "idea" or "sleep" and "furiously."

Motivated by these innovations, we recognize that both word embeddings and PMI are essentially derived from co-occurrence information [13][7][10]. However, word embeddings trained on large datasets generally capture the average contextual meaning of each word, offering great generalizability in comparison tasks. On the other hand, the pairwise scores provided by PMI estimate the relationships between word pairs, which do not generalize as much as the robustly learned discriminative word-specific representations [9]. With this understanding, we propose an advanced text complexity metric based on word embedding models.

We empirically demonstrate the merit of the proposed method. To begin with, we compare PMI and the proposed word embedding-based text complexity metrics to estimate the complexity of documents. We use OneStopEnglish [16] to evaluate the estimation by comparing it to the human-labeled, thus ground-truth readability scores. The results verify that the proposed method shows a statistically more meaningful relationship with the human-labeled complexity of the text than the PMI metrics.

## 2 The Proposed Text Complexity Metric

Flor et al. reported that a complex document tends to have words that are semantically farther from each other [4]. The semantic distance of words is the opposite concept of word similarity, which correlates with the co-occurrence of words in a context window. To be specific, words in a less complex text, such as "The dog barked and wagged its tail," tend to co-occur in the same sentence

or adjacent sentences, while the words in a more complex text, such as "Green ideas sleep furiously," rarely co-occur.

PMI can capture this concept in a statistical way by comparing the joint probability (i.e., the co-occurrence frequency) of the two words $p(x, y)$ and their *a priori* expected co-occurrence probability based on the product of their global word frequencies $p(x)p(y)$:

$$PMI(x; y) = \log \frac{p(x, y)}{p(x)p(y)}. \tag{1}$$

Hence, for example, if two words are rarely used in the corpus, i.e., both $p(x)$ and $p(y)$ are low, while they happen to co-occur frequently, i.e., $p(x, y)$ is relatively high, it means that the two words are semantically associated.

Flor et al.'s proposed method is based on the average *normalized* pointwise mutual information (NPMI) scores between all pairs of words in each document. Unlike the basic PMI, NPMI values are bounded between $-1$ and 1: $-1$ indicates that the two terms never co-occur together, while 1 indicates that the two terms always co-occur in the corpus. They went one step further and used positive normalized PMI (PNPMI), which assigns 0 to any negative NPMI score, and showed that PNPMI was a promising complexity metric.

$$\begin{aligned} NPMI(x; y) &= \frac{\log \frac{p(x,y)}{p(x)p(y)}}{-\log p(x, y)}, \\ PNPMI(x; y) &= \begin{cases} NPMI(x; y) & \text{if } NPMI(x; y) > 0 \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{2}$$

However, word embedding models such as Word2Vec have been reported to outperform more traditional count-based models such as PMI on measuring word similarity [8]. Word embedding is a multidimensional vector representation of words, derived by a shallow neural network that learns semantic relationships between words from a large corpus. Word2Vec was introduced in 2013, followed by other embedding methods, such as GloVe and FastText [10][14][1]. The main characteristics of these methods are that for a pair of words $x$ and $y$, the encoder function $f$ learns their $D$-dimensional embedding vectors $\mathbf{z}_x$ and $\mathbf{z}_y$,

$$\mathbf{z}_x \leftarrow f(x), \quad \mathbf{z}_y \leftarrow f(y), \tag{3}$$

whose distance is more likely to be high if the pair's semantic difference is high, and vice versa. While the semantic distance of two words is hard to represent as a function, the embedding vectors often preserve it in the simple Euclidean space. For example, a cosine distance could be used to quantify the semantic distance of $x$ and $y$:

$$D(x; y) = 1 - \frac{\mathbf{z}_x^\top \mathbf{z}_y}{|\mathbf{z}_x||\mathbf{z}_y|}. \tag{4}$$

Training of the encoding function $f$ utilizes various word pairs that share similar conceptual meanings. In addition, it is typical to use "negative sampling"

to expose the embedding to those learned from semantically far words as well, making the learned word representation more discriminative and robust. Therefore, it is expected that a word embedding $\mathbf{z}_x$ must contain its relationship to other words. This holistic representation is helpful when the word embeddings are compared against each other because the average semantic meanings the word carries are taken into account. On the other hand, a PMI (or its variants') value records narrower meanings of the word compared only to its counterpart. For example, since the word embeddings encode the semantics of words, a new pairwise relationship can be more robustly computed by comparing their embedding vectors, even though the pair did not appear in the corpus–PMI variants, however, consider them dissimilar words.

Finally, the proposed word embedding (WE)-based text complexity metric computes pairwise cosine distance values of all possible pairs and then calculates its average:

$$WE = \frac{1}{|\mathcal{V}|} \sum_{x,y \in \mathcal{V}} D(x; y), \tag{5}$$

where $|\mathcal{V}|$ stands for the number of all words that appear in the document.

Likewise, the similarity defined in the word embedding space is conceptually similar to PMI in that word embeddings are an implicit factorization of a word-context matrix, which can be computed based on PMI [9]. However, word embedding is superior in word similarity tasks, such as automatically evaluating the coherence of topics generated by topic modeling algorithms [3]. Hence, we expect word embedding models would be equivalent to or better than PMI variants in measuring the complexity of documents.

Note that, in the rest of the paper, we treat the PMI-based complexity measures by inverting the NPMI or PNPMI values, i.e., $-NPMI(x; y)$ and $1 - PNPMI(x; y)$, as the original scores are meant to measure the similarity of the word pairs, not their distance.

## 3 Experimental Studies

In this section, we present a comparative analysis between the proposed word embedding-based complexity metric and existing metrics based on pointwise mutual information (PMI), using the OneStopEnglish dataset [16].

### The Dataset

*Corpus with readability level annotation:* We use the OneStopEnglish corpus to assess the proposed text complexity metric as the dataset provides human experts' annotation of complexity levels on various news articles. The OneStopEnglish corpus comprises 189 sets of texts with three reading levels for each topic. The corpus was curated by `onestopenglish.com`, which is a worldwide English education service with plenty of resources for English language teachers. Participating teachers there rewrote each news article from the Guardian newspaper

**Table 1.** Example sentences for three reading levels [16].

| Level | Example Text |
|---|---|
| Advanced (Adv) | Amsterdam still looks liberal to tourists, who were recently assured by the Labour Mayor that the city's marijuana-selling coffee shops would stay open despite a new national law tackling drug tourism. But the Dutch capital may lose its reputation for tolerance over plans to dispatch nuisance neighbours to scum villages made from shipping containers. |
| Intermediate (Int) | To tourists, Amsterdam still seems very liberal. Recently the city's Mayor assured them that the city's marijuana-selling coffee shops would stay open despite a new national law to prevent drug tourism. But the Dutch capitals plans to send nuisance neighbours to scum villages made from shipping containers may damage its reputation for tolerance. |
| Elementary (Ele) | To tourists, Amsterdam still seems very liberal. Recently the city's Mayor told them that the coffee shops that sell marijuana would stay open, although there is a new national law to stop drug tourism. But the Dutch capital has a plan to send antisocial neighbours to scum villages made from shipping containers, and so maybe now people won't think it is a liberal city any more. |

**Table 2.** Some statistics about OneStopEnglish corpus [16]

| Feature | ADV | INT | ELE |
|---|---|---|---|
| avg num. words | 820.49 | 676.59 | 533.17 |
| FKGL | 9.5 | 8.2 | 6.4 |
| TTR | 0.56 | 0.432 | 0.42 |
| avg num. NP | 6.08 | 5.52 | 4.92 |
| avg num. VP | 4.49 | 4.03 | 3.49 |
| avg num. PP | 2.72 | 2.30 | 1.82 |

in three different versions: Advanced (Adv), Intermediate (Int), and Elementary (Ele). Table 1 shows three different versions of the same article example. It is known that advanced texts have positive correlations with various statistical and linguistic features, such as the number of words, Flesch-Kincaid grade level (FKGL), type-token ratio (TTR), and the average number of noun phrases (NP), verb phrases (VP), and preposition phrases (PP) [16]. We select this dataset to test the effectiveness of readability metrics because its reliability levels highly correlate with all the features (see Table 2).

*Word embedding models:* We use eight different pretrained word embedding models supported by the toolbox, Gensim[1], to compute different types of word embeddings. The pretrained models are based on three word embedding algorithms, including Word2Vec [10], GloVe [13], and FastText [7]. The models were trained on the following large-scale sources: Wikipedia, Twitter, and internet

[1] `https://github.com/RaRe-Technologies/gensim-data#models`

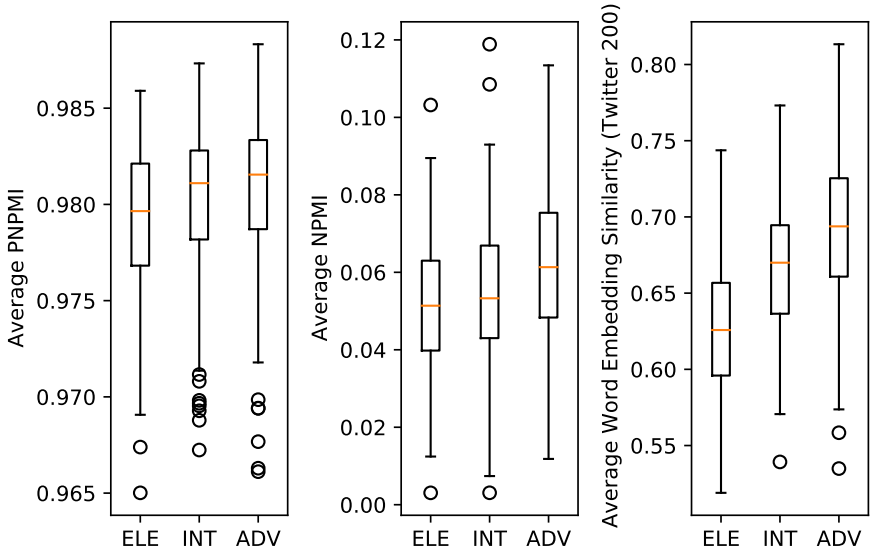


**Fig. 1.** Comparison of two PMI variants and one of the proposed WE similarity metrics. More widely spread boxes in the WE results indicate the proposed metric provides a stronger association with the three complexity levels (i.e., ELE, INT, and ADV).

news, with varying co-occurrence statistics. Among these, `glove-twitter` and `glove-wiki-gigaword` come in multiple versions, differentiated by their vector dimensions $D$. A large $D$ typically allows more representative vectors at the cost of a potential loss of linearity in the learned vector space. To investigate the impact of vector size, we experimented with three different dimensions: $D = 50$, 100, and 200.

### Experimental Setup

*Data pre-processing:* We employ SpaCy for tokenization and part-of-speech (POS) tagging of the OneStopEnglish corpus. We consider only content words in our analysis, as we aim to measure the complexity arising from the semantic distances between words, and content words carry greater weight in determining meaning. To calculate the WE-based complexity score for each document, we aggregate and average the cosine distances of individual pairs of words. Meanwhile, we calculate PNPMI and NPMI for each document using the Palmetto API, an open-source tool that computes NPMI based on word co-occurrence in the English Wikipedia [15].

*Comparison between the proposed WE complexity metrics and PMI variants:* The experiments consist of two steps. In the first experiment, we conducted a

**Table 3.** Comparison of two PMI variants and eight proposed WE similarity metrics using 50 content words from each document. Each `[higher reading level]-[lower reading level]` entry reports the percentage of source articles where a metric returns a higher complexity score (lower similarity score) for the more advanced version of the article. `avg-gap` is the average of the percentages of the three pairs. Also, for each metric, the correlation coefficient score between the complexity scores and the reading levels is reported.

| Models | `int-ele` | `adv-int` | `adv-ele` | `avg-gap` | Correlation Coefficient |
|---|---|---|---|---|---|
| `glove-twitter-50` | 90% | 81% | 96% | 89% | 0.51 |
| `glove-twitter-100` | 92% | 86% | 97% | 91% | 0.51 |
| `glove-twitter-200` | 93% | 86% | 96% | 92% | 0.52 |
| `glove-wiki-gigaword-50` | 88% | 79% | 95% | 87% | 0.42 |
| `glove-wiki-gigaword-100` | 89% | 81% | 96% | 89% | 0.45 |
| `glove-wiki-gigaword-200` | 92% | 83% | 96% | 90% | 0.46 |
| `word2vec-google-news-300` | 91% | 74% | 93% | 86% | 0.42 |
| `fasttext-wiki-news-subwords-300` | 78% | 59% | 78% | 72% | 0.20 |
| NPMI | 56% | 67% | 72% | 65% | 0.22 |
| PNPMI | 68% | 59% | 68% | 65% | 0.17 |

comparison between two PMI variants, including NPMI and PNPMI, and our proposed WE complexity metric. We use the first 50 content words per document because otherwise, the number of word pairs grows intractably large for the relatively slow palmetto API. The first experiment confirmed a significant enough gap between the PMI and word embedding groups to establish the superiority of our proposed WE metric.

*Comparison among the proposed metrics derived by different WE models:* The second experiment follows to determine the best configuration among the eight pre-trained word embedding models. For this experiment, we consider all words in the documents, rather than limiting to the first 50, as computing WE vectors and their pairwise similarities is sufficiently fast.

### Experimental Results

*Comparison between the proposed WE complexity metrics and PMI variants:* First, we compare the WE complexity metrics and NPMI and PNPMI using box plots. We report the box plot of `glove-twitter-200` as the representative of the proposed method because all the WE complexity metrics except `fasttext-wiki-news-subwords-300` show similar box plots. Figure 1 shows that the gaps between different reading levels are much wider in the proposed WE group than in the PMI groups. For a fair comparison to the slow NPMI and PNPMI calculation, we only use the first 50 words for the WE scores, too. These box plots show the overall superiority of the proposed WE scores. However, since a box plot aggregates all documents that belong to the same reading

**Table 4.** Comparison among the proposed metrics derived by different WE models using all words from each document instead of the first 50. Otherwise, the setup is identical to Table 3's.

| Models | int-ele | adv-int | adv-ele | avg-gap | Correlation Coefficient |
|---|---|---|---|---|---|
| glove-twitter-50 | 98% | 89% | 99% | 95% | 0.602 |
| glove-twitter-100 | 98% | 92% | 99% | 96% | 0.594 |
| glove-twitter-200 | 98% | 93% | 99% | 96% | 0.597 |
| glove-wiki-gigaword-50 | 98% | 90% | 98% | 96% | 0.494 |
| glove-wiki-gigaword-100 | 98% | 92% | 99% | 96% | 0.534 |
| glove-wiki-gigaword-200 | 98% | 92% | 98% | 96% | 0.543 |
| word2vec-google-news-300 | 98% | 87% | 99% | 95% | 0.542 |
| fasttext-wiki-news-subwords-300 | 90% | 55% | 87% | 77% | 0.243 |

level, the graphs do not show how well a complexity metric distinguishes different reading-level versions derived from the same original news article.

Table 3 shows how well each metric distinguishes different reading levels of the same source article. Once again, only 50 content words from each document are used. For example, `int-ele` is the percentage of source articles where the intermediate version has a higher score than the elementary version, i.e., 100% means a perfect discrimination performance. First, we observe that the average gap of those three comparisons, `int-ele`, `adv-int`, and `adv-ele`, summarizes that all WE-derived complexity metrics (the top 8 rows, ranging between 72 and 92%) are significantly better than the PMI variants (the bottom two rows, with an accuracy of 65%). Furthermore, the absolute correlation coefficient of NPMI and PNPMI, -0.22 and -0.17, is significantly lower than those of the word embedding models, between 0.42 and 0.52 (except for the fast-text case as an outlier, 0.20).

Likewise, the word embedding models outperform the PMI group significantly in differentiating documents with different reading levels both per source article and regardless of source articles.

*Comparison among the proposed metrics derived by different WE models* : We also compare the eight WE configurations more thoroughly. In Table 4, we repeat the same experiment done in Table 3, but on all words in the documents instead of only the first 50. Indeed, using all words led to higher scores and correlations across all pairs and pre-trained models. However, the increase in the number of words did not result in a significant change in their rankings. Overall, while `fasttext-wiki-news-subwords-300` shows the lowest distinguishing power across the three pairs, the rest pre-trained models perform similarly in widening the gaps. For example, in all cases, 98% of the time, the proposed WE-based metrics succeed in distinguishing elementary-level versions from their corresponding intermediate versions, except for `fasttext-wiki-news-subwords-300`. When it comes to distinguishing the intermediate and advanced levels documents, `glove-twitter-200` achieves the highest score, 93%. Finally, except for

`fasttext-wiki-news-subwords-300`, with more than 98% of accuracy, all WE configurations separate the advanced and elementary versions.

We compare three different vector sizes, $D = 50, 100$, and 200, for the two WE configurations: `glove-twitter` and `glove-wiki-gigaword`, to determine the relationship between the vector size of the word embedding and the distinguishing power. Table 4 shows that there is no meaningful correlation between the two factors. Among the pre-trained word embedding models, `glove-twitter` models show higher distinguishing power in terms of correlation coefficient scores. Although their correlation coefficient scores are almost equivalent, `glove-twitter-50` showed the highest score, 0.602.

## 4 Conclusion and Future Work

This study proposed a new text complexity metric that measures the average pairwise distance between words, relying on the assumption that more complex documents tend to have words that are semantically far from each other, as studied in Flor et al.'s work using PMI-based word similarity metrics. This study adopted the metric after modification but with greater generalizability, which was achieved by redefining the semantic distance between the two words in the word embedding space. Thanks to the representational power of word embeddings, the proposed metric showed superior performance on readability level estimation tasks, We investigated the complexity pattern of the OneStopEnglish dataset, which offers three readability levels for the same article. Our proposed method demonstrated a stronger statistical correlation with the dataset's expert-labeled readability scores than existing PMI-based approaches. In future work, we will explore other complexity metrics based on context-dependent word embeddings, and see the relationships to extended types of text.

## References

1. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics **5**, 135–146 (2017)
2. DuBay, W.H.: The principles of readability. Impact Information (2004)
3. Fang, A., Macdonald, C., Ounis, I., Habel, P.: Using word embedding to evaluate the coherence of topics from twitter data. In: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. pp. 1057–1060 (2016)
4. Flor, M., Klebanov, B.B., Sheehan, K.M.: Lexical tightness and text complexity. In: Proceedings of the Workshop on Natural Language Processing for Improving Textual Accessibility. pp. 29–38 (2013)
5. François, T., Miltsakaki, E.: Do nlp and machine learning improve traditional readability formulas? In: Proceedings of the First Workshop on Predicting and Improving Text Readability for target reader populations. pp. 49–57 (2012)
6. Hiebert, E.H.: Readability and the Common Core's staircase of text complexity. Santa Cruz, CA: TextProject Inc (2012)

7. Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., Mikolov, T.: Fasttext. zip: Compressing text classification models. arXiv preprint arXiv:1612.03651 (2016)
8. Levy, O., Goldberg, Y.: Neural word embedding as implicit matrix factorization. Advances in neural information processing systems **27**, 2177–2185 (2014)
9. Levy, O., Goldberg, Y., Dagan, I.: Improving distributional similarity with lessons learned from word embeddings. Transactions of the Association for Computational Linguistics **3**, 211–225 (2015)
10. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
11. Napolitano, D., Sheehan, K.M., Mundkowsky, R.: Online readability and text complexity analysis with textevaluator. In: Proceedings of the 2015 conference of the north american chapter of the association for computational linguistics: Demonstrations. pp. 96–100 (2015)
12. Nelson, J., Perfetti, C., Liben, D., Liben, M.: Measures of text difficulty: Testing their predictive value for grade levels and student performance. Council of Chief State School Officers, Washington, DC (2012)
13. Pennington, J., Socher, R., Manning, C.: GloVe: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543. Association for Computational Linguistics, Doha, Qatar (Oct 2014). https://doi.org/10.3115/v1/D14-1162, https://aclanthology.org/D14-1162
14. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
15. Röder, M., Both, A., Hinneburg, A.: Exploring the space of topic coherence measures. In: Proceedings of the eighth ACM international conference on Web search and data mining. pp. 399–408 (2015)
16. Vajjala, S., Lučić, I.: Onestopenglish corpus: A new corpus for automatic readability assessment and text simplification. In: Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications. pp. 297–304 (2018)
17. Zheng, J., Yu, H.: Assessing the readability of medical documents: a ranking approach. JMIR medical informatics **6**(1), e8611 (2018)