Comparative analysis of word embeddings in assessing semantic similarity of complex sentences

DHIVYA CHANDRASEKARAN, AND VIJAY MAGO
Lakehead University, Thunder Bay, Ontario, Canada P7B 5E1
Corresponding author: Dhivya Chandrasekaran (e-mail: dchandra@lakeheadu.ca)
This work was supported by the Ontario Council on Articulation and Transfer (ONCAT)

ABSTRACT Semantic textual similarity is one of the open research challenges in the field of Natural Language Processing. Extensive research has been carried out in this field and near-perfect results are achieved by recent transformer-based models in existing benchmark datasets like the STS dataset and the SICK dataset. In this paper, we study the sentences in these datasets and analyze the sensitivity of various word embeddings with respect to the complexity of the sentences. In this article, we build a complex sentence dataset comprising of 50 sentence pairs with associated semantic similarity values provided by 15 human annotators. Readability analysis is performed to highlight the increase in complexity of the sentences in the existing benchmark datasets and those in the proposed dataset. Further, we perform a comparative analysis of the performance of various word embeddings and language models on the existing benchmark datasets and the proposed dataset. The results show the increase in complexity of the sentences has a significant impact on the performance of the embedding models resulting in a 10-20% decrease in Pearson’s and Spearman’s correlation.

INDEX TERMS Language models, Natural language processing, Readability analysis, Semantic Similarity, Word embeddings

I. INTRODUCTION

ONE of the core components of Natural Language Processing (NLP) is assessing the semantic similarity between text data. The versatility of natural languages has made it a challenging task for researchers to capture the semantics of text data using numerical representations. Measuring the semantics of text data is essential in various NLP tasks like text summarization [2], topic modelling [3], text simplification [4], machine translation [5], question answering tasks [6], information retrieval [7] and so on. Extensive research has been carried out in the past decade in the field of semantic similarity to construct vector representations that preserve the syntactic and semantic properties of words [8]. Word embeddings like word2vec [9] and GloVe [10] exploit the principle of the distributional hypothesis [11] i.e., “similar words occur in similar context”. These methods use the advancements in deep learning techniques to capture the semantics of the words using large text corpora. Recent language models like BERT [12], RoBERTa [13], and ALBERT [14] use transformers to build vector representations of text data from underlying corpora by traversing through the corpora in both directions. Over the years various benchmark datasets have been used for comparing the performance of models in measuring semantic similarity between text data. Two of the most popular datasets are the STS benchmark dataset [15] and the SICK dataset [16] on which the BERT models have achieved near-perfect results [12]. Analyzing the readability of the sentences in these datasets, we find that the sentences in these datasets have a low readability index which is a measure of complexity of sentences [17]. However, various real world applications of semantic similarity involve more complex sentences to be analysed [18]. In this paper, a new dataset with sentences of a higher degree of complexity than the existing datasets is proposed. The dataset comprises 50 pairs of complex sentences, with corresponding similarity scores provided by 15 human annotators. A comparative analysis of various existing text embeddings is performed on two existing benchmark datasets and the proposed complex sentence dataset. The results indicate that the performance of the embedding models decreases significantly with the increase in complexity of the sentences. To the best of our knowledge, this research work is the first attempt to analyze
the impact of complexity of sentences on the performance of text embedding models.

The remaining of the paper is organized as follows. Section II of this paper provides a brief description of the existing research works carried out in the field of semantic similarity. Section III describes two of the existing benchmark datasets and five different word embeddings chosen for the comparative analysis. Section IV discusses in detail the methodology adopted to construct the proposed dataset and provides a detailed description of the readability analysis that compares the complexity of sentences in the existing datasets to the sentences in the proposed dataset. Section V provides a comparative study of the performance of various word embeddings and provides an insight into the inferences made that would guide the future scope of this research.

II. RELATED WORK

Similarity between text data does not always attribute to the lexical or syntactic similarity between text. While two sentences that contain exactly the same words may mean something completely different, it is possible that sentences with different sets of words provide the same meaning. Hence while assessing the similarity between text data, it is important to understand the meaning conveyed by the text. The similarity between the meaning of the text is known as semantic textual similarity (STS). For the past three decades, various semantic similarity methods have been proposed to measure semantic similarity. These methods are widely classified as knowledge-based methods and corpus-based methods. The knowledge-based methods use structurally strong ontologies like Wordnet \[19\], DBPedia \[20\], Wikipedia \[21\], Wiktionary \[22\] etc. These ontologies are often used like graphs and various edge counting methods, consider the words in the taxonomy as nodes and calculate the distance between the words using the edges between them. The greater the distance between the words the lower their similarity value \[21\]. However, these methods assume that the length of these edges to be similar which is not always the case. Another type of knowledge-based approach called the feature-based methods, assess the similarity based on features of the words, like their dictionary definition, neighboring concepts, etc. derived from the underlying ontologies \[22\]. Knowledge-based methods are computationally simple and are efficient in distinguishing the different meanings of words solving the problem of ambiguity with concepts like polysemy and synonymy. However they are heavily dependent on the underlying taxonomies, they do not account for the versatility of natural language, and structured taxonomies for languages other than English are not common \[8\]. Corpus-based semantic similarity methods use statistical principles to capture the contextual semantics of data using large underlying corpora. The principle of distributional hypothesis states that words with similar meanings occur together in documents and this principle forms the basis of most corpus-based methods, while these methods do not take into account the actual meaning of individual words. Word vectors, also called word embeddings, are constructed using corpus-based methods and the similarity is measured based on the angle or distance between these vectors. The dimensionality of these embeddings depends on the size of the corpus. While using significantly large corpora various dimensionality reduction techniques like Singular Value Decomposition (SVD), Principal Component Analysis (PCA), and filtering techniques are used to achieve computational efficiency. These word embeddings are the fundamental components of recent techniques that use the advancements in deep neural networks to achieve a significant increase in performance in semantic similarity tasks. word2vec proposed by Milokov et al. \[9\] and GloVe vectors proposed by Pennington et al. \[10\] have proven to be major breakthroughs in the field of semantic similarity and they are two of the most widely used word embeddings to date. In 2019, Delvin et al. \[12\] proposed the Bidirectional Encoder Representation from Transformers (BERT) language model which used transformers to build word embeddings, which were further used for various downstream NLP applications. Variations of the BERT models like ALBERT \[14\] and RoBERTa \[13\] were also published in 2019 and have outperformed the existing semantic similarity models achieving state-of-the-art results \[23\]. Raffel et al. \[24\] proposed the T5: text-to-text transformer model which used the principle of transfer learning and was trained on a custom-built corpus called “Colossal Clean Crawled Corpus” or C4. This model tied for first place with ALBERT by achieving a Pearson’s correlation of 0.925 on the STS dataset. In the following section, we describe in detail the two most widely used benchmark datasets for assessing the performance of semantic similarity methods, and five of the most popular word embeddings that we have chosen for comparison in this paper.

III. DATASETS AND WORD EMBEDDING MODELS

A. SEMANTIC SIMILARITY DATASETS

The first and the most widely used word-to-word similarity dataset, the R&G dataset \[25\], was proposed by Rubenstein and Goodenough in 1965 with 65 English noun pairs annotated by 51 native English speakers with similarity values ranging between 0 and 4. Some of the prominent datasets used are compiled in Table \[7\] in Appendix \[C\]. While many datasets were published over the years as the benchmark for models measuring sentence-level semantic similarity, the SICK dataset, and the STS datasets are those that gained significance. For our analyses and comparison, we choose these two datasets owing to their wide usage and popularity.

1) SICK Dataset \[16\]

Marelli et al. \[16\] compiled the SICK dataset for sentence-level semantic similarity/relatedness in 2014 composed of 10,000 sentence pairs obtained from the ImageFlickr 8 and MSR-Video descriptions dataset. The sentence pairs were
derived from image descriptions by various annotators. 750 random sentence pairs from the two datasets were selected, followed by three steps to obtain the final SICK dataset: sentence normalization, sentence expansion, and sentence pairing. Initially, all the sentences were normalized by removing unwanted syntactic or semantic phenomena. This process was carried out by two different annotators and checked for compatibility. In instances of contradiction, a third annotator made the choice by analyzing the two alternatives, and if both were correct a random choice was made. From the normalized sentences, 500 pairs were chosen for expansion. The process of expansion involved generating three different versions of the normalized sentence pairs - a similar sentence using meaning preserving transformation, a completely opposite version using negative transformation, and a sentence with the same words but different meanings using random shuffling. FIGURE 1 shows an example as presented by the authors. Finally, each normalized sentence was paired with all its expanded versions, along with some random pairing. A survey was then conducted on Amazon Mechanical Turk where the workers were requested to rank the similarity/relatedness over a scale of 1 to 5 with 1 representing that the sentences were highly dissimilar and 5 representing that the sentences were highly similar. Ten unique responses for each sentence pair were collected and the average of the ten ratings was assigned as the gold standard. Each sentence pair is associated with a relatedness score and a text entailment relation as well. The three entailment relations are NEUTRAL, ENTAILMENT and CONTRADICTION.

2) STS Dataset [15]
In order to encourage research in the field of semantic similarity, semantic textual similarity tasks called SemEval have been conducted since 2012. The organizers of the SemEval tasks collected sentences from a wide variety of sources and compiled them to form a benchmark dataset against which the performance of the models submitted by the participants in the task was measured. While the dataset contains different tracks with sentences from different languages, we focus on the English component of the dataset. The English component of the dataset contains 8,295 sentence pairs of which 5,720 are provided as training samples and the remaining 2,575 sentences from the test set. The dataset is built over the years and contains the sentences that were used from 2012 to 2017. Table 1 provides a breakdown of the source of the sentences in the dataset and the year they were appended to form the current version of the dataset. The sentences were annotated using Amazon Mechanical Turk. The quality of the annotators was assessed using the ‘masters’ provided by the platform and five unique annotations were obtained for each sentence pair. The similarity values ranged between the values 0 and 5, with 5 indicating that the sentences are completely similar and 0 indicating the sentences are completely dissimilar. The final similarity score was obtained by taking an average of the five unique responses from the annotators.

TABLE 1: STS English language training dataset (2012-2017) [26]

| Year | Dataset | Pairs | Source |
|------|---------|-------|--------|
| 2012 | MSRPar  | 1500  | newswire |
| 2012 | MSR-vid | 1500  | videos |
| 2012 | OnWN    | 750   | glosses |
| 2012 | SMTNews | 750   | WMT eval. |
| 2012 | SMTeuroparl | 750 | WMT eval. |
| 2013 | HDL     | 750   | newswire |
| 2013 | FNWN    | 561   | glosses |
| 2013 | OnWN    | 561   | glosses |
| 2013 | SMT     | 561   | MT eval. |
| 2014 | HDL     | 750   | newswire headlines |
| 2014 | OnWN    | 750   | glosses |
| 2014 | Deft-forum | 450  | forum posts |
| 2014 | Deft-news | 300  | news summary |
| 2014 | Images  | 750   | image descriptions |
| 2014 | Tweet-news | 750 | tweet-news pairs |
| 2015 | HDL     | 750   | newswire headlines |
| 2015 | Images  | 375   | Q & A forum answers |
| 2015 | Ans.-student | 375 | committed belief |
| 2015 | Ans.-forum | 375 | Q & A forum answers |
| 2016 | HDL     | 249   | newswire headlines |
| 2016 | Plagiarism | 230  | short-answers plag. |
| 2016 | post-editing | 244  | MT posteds |
| 2016 | As.-Ans | 254   | Q & A forum answers |
| 2016 | Quest.-Quest | 209  | Q & A forum questions |
| 2017 | Trail   | 23    | Mixed STS 2016 |

B. WORD-EMBEDDING MODELS
Distributed semantic vector representations of words called word embeddings have gained significant attention in the past decade, and a wide variety of word embeddings have been proposed over the years [27]. Word embeddings are constructed by analyzing the distribution of words in any text data and are well known to capture the semantics of the words thus making them a significant component of a wide variety of semantic similarity algorithms. The word2vec model uses neural networks to construct word embeddings and it has been one of the most widely used word-embeddings [27]. GloVe vectors employ word co-occurrence matrices to identify the distribution of words, which is then statistically used to build word vectors that capture the semantics of the target word with respect to its neighboring words. Pre-trained word embeddings provided by recent transformer based models achieved state-of-the-art results in a wide range of NLP tasks, including semantic similarity. In this section, we discuss in detail five of the popular word embeddings that are publicly available.

1) word2vec [9]
Mikolov et al. proposed a word embedding called word2vec in 2013, using a simple neural network that converted the given input word to a dense vector representation. Two different models of word2vec were proposed namely, the Skip-gram model and the Continuous Bag of Words (CBOW) model. In the skip-gram model, the neural network is optimized to
predict a target word given its context words, whereas in the CBOW model, the neural network predicts the neighboring words given a target word. The value vector in the hidden layer of the optimized neural network is used as the vector representation of the word. The number of neurons in the hidden layer of the neural network determines the dimension of the word vector. The models are trained using Google News corpus, which contains 1 million words. The model produced state-of-the-art results in 2013 and is used widely among researchers owing to the simplicity in the construction of the neural network. The major claim of the authors was that when simple algebraic operations were performed on these vectors the results were closely related to human understanding. For example, as shown in FIGURE 2, the difference between $V_k$ (vector representation for the word ‘king’) and $V_m$ (vector representation for the word ‘man’) added to $V_w$ (vector representation for the word ‘women’) provides a vector that is close to vector $V_q$ (vector representation for the word ‘queen’). This can be mathematically represented as,

$$V_q \simeq (V_k - V_m) + V_w \quad (1)$$

2) GloVe [10]

Researchers at Stanford University proposed a vector representation for words using the word-to-word co-occurrence matrices. Given a corpus, a global co-occurrence matrix is built where each row and column represents the words in the corpus. The underlying principle for the construction of these vectors is that similar words occur together. The model proposed uses ‘log-bilinear regression’ to create a word vector space with substructures that provide meaningful word-vector representations. GloVe vectors were trained on five different corpora like a common web crawled corpus and a Wikipedia data dump resulting in 400,000 unique words to form the co-occurrence matrix. Pretrained word vectors of 3 different dimensions (50, 100, and 300) were released by the authors and they claimed that GloVe vectors outperformed word2vec and achieved the state-of-the-art results in 2014.

3) BERT [12]

In 2019, the BERT transformer model surpassed the state-of-the-art results in 11 different NLP tasks, including semantic similarity. BERT uses the transformer model proposed by Vaswani et al. [28]. The BERT models follow two distinct steps to adapt to specific NLP tasks namely, pretraining and fine-tuning. The transformer contains an encoder and decoder module, the encoder containing 6 identical layers stacked above each other. Each layer consists of sublayers comprising of a multi-head attention mechanism followed by a fully connected feed-forward neural network. The decoder is similar to the encoder, with one additional sub-layer of multi-head attention, which captures the attention weights in the output of the encoder. The model is pretrained using the Book corpus [29] and Wikipedia dump comprising of nearly 3300 million words. Pre-training is carried out with the help of two tasks, namely, ‘Masked Language Model (MLM)’ and ‘Next Sentence Prediction (NSP)’. In the first task, random words in the corpus are masked and the model is optimized to predict the masked tokens. In the second task, the model is optimized to predict whether or not a sentence follows another given sentence. The BERT models thus produce bidirectional representations of words taking into consideration the context of the word in both directions. In general, the BERT model is fine-tuned using labeled training data to accommodate a specific NLP task. The model was fine-tuned with STS dataset and achieved state-of-the-art results.
results.

4) RoBERTa [13] proposed a robustly optimized version of BERT, by replicating the work of Delvin et al. [12] and adding to it an improved training procedure. They added more training data and trained for a longer period of time, achieving state-of-the-art results, which proved the BERT architecture was equipped to perform better than many later models, but it was under-trained. While BERT was trained on the Book Corpus and Wikipedia corpus, the RoBERTa model was trained on four different corpora namely, the Book Corpus, the Common Crawl News dataset, OpenWebText dataset, and the Stories dataset. RoBERTa uses variations of the pretraining tasks used by the BERT model. It uses both static and dynamic masking, and by performing dynamic mask the training data is duplicated ten times thus enabling the model to encounter each masked token four times over the forty epoch training. The authors study the effect of the ‘Next Sentence Prediction’ task by replacing it with the prediction of subsequent segments or sentences in a document and prove that the performance increased by removing the NSP task. The model outperforms the BERT model and achieves state-of-the-art results in 11 NLP tasks including semantic similarity.

5) ALBERT [14]
One of the major challenges in the BERT model is the time and resource requirement to pre-train a complex transformer model. Lan et al. [14] proposed a Lite version of BERT by employing two different parameter reduction techniques to aid in scaling the pre-trained models namely, Factorized Embedding Parameterization (FEP) and Cross-layer Parameter Sharing (CPS). Using FEP, the authors split the vector embedding matrix of the vocabulary into two smaller matrices thus making the size of the vector independent of the hidden layer of the model. Using CPS enables the sharing of parameters across layers thus preventing the increase in the number of parameters as the depth of the layers increase. ALBERT also replaces one of the pretraining tasks, next sentence prediction, in BERT with inter-sentence coherence loss that focuses on the coherence between two consecutive sentences. ALBERT has outperformed all the existing models and holds the highest Pearson’s correlation in the STS dataset.

A. HUMAN ANNOTATION

1) Annotation
The sentence pairs were marked with a ‘binary annotation’ as ‘Similar’ and ‘Dissimilar’ based on whether they are the definition of the same topic to ascertain the ground truth. Then two separate surveys were conducted to obtain the human annotation of the similarity value between the sentence pairs. The initial survey was conducted among 5 graduate students pursuing a Masters’ in Computer Science. The students were requested to mark the similarity between the provided sentence pairs over a range of values between 0 and 5, where 0 indicates that the sentences are completely dissimilar and 5 indicates that the sentences are completely similar. In order to capture the ratings from a larger audience, the survey was further extended to Amazon Mechanical Turk (MTurk) requesting 10 unique responses for each sentence pair. The survey environments of both surveys are depicted in FIGURE 5 and FIGURE 6 in Appendix B.

| SNo | Topic              | SNo | Topic     |
|-----|--------------------|-----|-----------|
| 1   | Computer science   | 10  | Pseudo code                     |
| 2   | Computer program   | 11  | Programming language            |
| 3   | Algorithm          | 12  | Data analytics                   |
| 4   | Data structures    | 13  | Computer Security                |
| 5   | Artificial Intelligence | 14 | Computer virus                  |
| 6   | Computer programming | 15 | Cloud computing                 |
| 7   | Operating systems  | 16  | Server                           |
| 8   | Database           | 17  | Firewall                         |
| 9   | Computer Architecture | 18 | Outlier                          |
TABLE 3: Distribution of complex sentences and complex sentence pairs in the existing benchmark datasets. *Complex sentence pairs = both sentences in the pair have grade-level >12.

|                | No. of unique sentences | No. of Complex sentences  | % of complex sentences | Total Sentence pairs | Complex Sentence pairs* |
|----------------|-------------------------|---------------------------|------------------------|----------------------|------------------------|
| STS            | 2549                    | 326                       | 12.79                  | 1379                 | 36                     |
| SICK           | 6076                    | 17                        | 0.28                   | 9840                 | 0                      |

The formula below:

$$S = \frac{\sum_{i=0}^{5}(w_i \times s_i)}{\sum_{i=0}^{5} w_i}$$  \hspace{1cm} (2)

where,

$$s_i$$ represent the values from 0 to 5 respectively and

$$w_i$$ represent the corresponding weights and are calculated as,

$$w_i = \left( \frac{\text{No. of responses with } i \text{ similarity score}}{\text{Total number of responses}} \right)$$  \hspace{1cm} (3)

3) Limitations:

The important limitation of this research can be attributed to the fewer number of sentence pairs compared to the large datasets used in the comparison. The major factor for the fewer number of sentences was the cost associated with the annotation of the sentence pairs. The STS dataset and the SICK dataset spent 0.02 USD and 0.01 USD per annotation, whereas the cost per annotation in the proposed dataset is 0.65 USD. The cost spent ensured the quality of annotators by selecting well-established annotators with US graduate-level education and a proven track record of quality work in their previous participations on the Amazon MTurk platform. TABLE 3 shows that only 13% of the sentences in STS and less than 0.5% of the sentences in the SICK dataset are complex sentences. When considering sentence pairs in which both sentences are complex (grade level greater than 12) there are only 36 sentence pairs in the STS dataset and none in the SICK dataset. This clearly shows that although the proposed dataset has only 50 sentence pairs it is still greater than the number of complex sentence pairs in the existing benchmark datasets.

B. READABILITY ANALYSIS

Readability indices are used by researchers to measure the complexity of text data mostly in text simplification tasks \[30, 17, 31, 32, 33, 34\]. In order to prove that the sentences chosen for building this dataset are more complex than the existing benchmark datasets, a comparative readability analysis is conducted between the two existing benchmark datasets and the proposed dataset. The below-mentioned readability grade-level scores indicate the grade level of education required by the reader to comprehend the given text which in turn reflects the complexity of the sentences. For example, a readability index of 10.4 indicates that a student of grade 10 would be able to read the given text. Various readability indices used and the formula for determining them are provided below.
Comparison of mean values of readability indices (Random generator seed = 5)

Comparison of mean values of readability indices (Random generator seed = 8)

Comparison of mean values of readability indices (Random generator seed = 48)

FIGURE 4: Comparison of mean readability indices among three given datasets.

- **Flesch-Kincaid Grade Level** [35]
  \[
  = 0.39 \left( \frac{\text{totalwords}}{\text{totalsentences}} \right) + 11.8 \left( \frac{\text{totalsyllables}}{\text{totalwords}} \right) - 15.59 \]  
  (4)

- **Coleman-Liau Index** [36]
  \[
  = 0.588L - 0.296S - 15.8 \]  
  (5)

where,

- \( L \) denotes the number of characters per 100 words and
- \( S \) denotes the number of sentences per 100 words.

- **Automated readability Index** [35]
  \[
  = 4.71 \left( \frac{\text{characters}}{\text{words}} \right) + 0.5 \left( \frac{\text{words}}{\text{sentences}} \right) - 21.43 \]  
  (6)
Linsear Write
For each sample of 100 words,

\[
r = \frac{1 \times (\text{Easywords}) + 3 \times (\text{Hardwords})}{\text{No. of sentences in sample}}
\]  
\[
(7)
\]

where,

\[
\text{Easywords} = \text{words with less than 2 syllables and}
\]

\[
\text{Hardwords} = \text{words with more than 3 syllables.}
\]

Gunning fog index [37]
\[
= 0.4 \left[ \left( \frac{\text{words}}{\text{sentences}} \right) + 100 \left( \frac{\text{complex words}}{\text{words}} \right) \right]
\]
\[
(9)
\]

where,

\[
\text{complex words} = \text{words consisting more than or equal to three syllables}
\]

Text Standard:
An aggregated score based on all the above readability indices.

The STS training dataset contains 10,818 unique sentences and the SICK dataset contains 6,076 unique sentences. On analyzing the complexity of these sentences using the above-mentioned readability indices we find that 70% of sentences in the STS dataset and 90% of sentences in the SICK dataset have an aggregate readability score below 10, while only 25% of the sentences in the proposed dataset are below the index 10. This clearly indicates that the two existing datasets have predominantly simpler sentences. Table 8 in Appendix D shows a few sample sentences from STS dataset for respective grade levels. In order to compare the complexity of the datasets, we select the readability indices of 52 random sentences from the two benchmark datasets and 52 sentences from the proposed dataset. This process is repeated with three different seeds for random selection. FIGURE 4 shows the results of the comparison between the mean value of six different readability indices among the three datasets repeated thrice and we can clearly see a significant increase in the complexity of the sentences. The results show us that while sentences in the STS dataset and SICK dataset can be interpreted by 6th graders and 4th graders respectively, the proposed dataset requires the knowledge of a 12th grader to comprehend the meaning of the provided sentences clearly indicating the increase in complexity in the sentences of the proposed dataset.

V. COMPARATIVE ANALYSIS
A. EXPERIMENTAL SETUP
We perform a comparative analysis between the five chosen word embeddings, to assess the impact of the complexity of the sentences. We compare their performance across the two benchmark datasets and the proposed dataset. For word embedding models the sentence vectors are formed by simply taking the mean of the word vectors. In transformer-based models, a mean pooling is added to the model to form the sentence vectors. Initially, we replicate the results provided by the authors of each model using the STS dataset and use the same models to measure the similarity scores for both SICK dataset and the proposed complex sentence dataset.

In the first model, the pre-trained word2vec model, trained on the Google News dataset containing 300-dimensional vectors with a vocabulary of 3M words provided by gensim python library is used for building the word vectors. Then, the 300 dimensional GloVe vectors pre-trained on a Wikipedia dump corpus with 6B tokens provided by the gensim library are used to build the word vectors. However, since the proposed sentences are definitions of computer science related topics, we use a specific corpus - the computer science corpus to train the word2vec and GloVe models initialized with the pretrained weights using transfer learning. The PetSca[5] tool that traverses through the Wikipedia categories and subcategories based on provided keywords is used for building the corpus. Using ‘computer_science’ as the category and a depth ‘2’ which indicates the level of subcategories to be included, the ‘computer science corpus’ containing 4M tokens is used to train both the word2vec and GloVe models.

Since the different BERT models do not provide explicit word embeddings, we use the Sentence-BERT framework proposed by Reimer’s et al. [38] to compare their performance. The various BERT models selected for comparison and their configuration are listed above in TABLE 4. We use the SentenceTransformer python library to initialize the model with the weights of pretrained BERT models followed by a mean pooling layer to form the sentence vectors. In order to factor in the effect of ‘fine-tuning’ one of the prominent characteristics of transformer-based models, we finetune the BERT models, with the AllNLI dataset. To estimate the impact of supervised learning in the quality of the word vectors provided by transformer-based models we experiment with the BERT models fine-tuned with both the AllNLI dataset and the SICK dataset on all three datasets included in the comparison. While the T5 model proposed by Raffel et al. [24] is one of the models that achieved the best performance in the STS dataset, due to computational constraints we were not able to replicate the model hence it is not included in our comparison.

Table 4: Configuration of BERT, RoBERT, and ALBERT models compared in this paper.

| Model Name         | Transformer Blocks | Hidden Layers | Attention heads | Total Parameters |
|--------------------|--------------------|--------------|-----------------|-----------------|
| BERT_BASE          | 24                 | 768          | 12              | 110M            |
| BERT_LARGE         | 24                 | 1024         | 16              | 340M            |
| RoBERT_BASE        | 24                 | 1024         | 16              | 340M            |
| RoBERT_LARGE       | 24                 | 2048         | 32              | 60M             |
| ALBERT_LARGE       | 12                 | 4096         | 64              | 235M            |
| ALBERT_XLARGE      |                    |              |                 |                 |

5 https://petsca.wmflabs.org/
TABLE 5: Pearson’s correlation and Spearman’s correlation percentages of the word embedding models in STS Benchmark Dataset, SICK dataset, and Proposed dataset.

| Supervision | Model                          | Pearson’s Correlation                     | Spearman’s Correlation                      |
|-------------|--------------------------------|--------------------------------------------|---------------------------------------------|
|             |                                | STS | SICK | Complex Sentences | STS | SICK | Complex Sentences |
| Unsupervised| word2vec                       | 62.61 | 72.67 | 48.30 | 58.71 | 62.13 | 49.04 |
|             | GloVe                          | 45.42 | 63.53 | 43.90 | 46.88 | 55.59 | 50.14 |
|             | word2vec + CS Corpus           | 56.77 | 67.75 | 52.35 | 53.27 | 59.38 | 46.79 |
|             | GloVe + CS Corpus              | 41.09 | 60.44 | 48.02 | 42.85 | 54.20 | 42.52 |
| Partially supervised| BERT large + NLI + Mean Pooling | 76.17 | 73.65 | 54.78 | 79.19 | 73.72 | 54.47 |
|             | BERT base + NLI + Mean Pooling | 74.11 | 72.95 | 50.91 | 76.97 | 72.89 | 47.73 |
|             | RoBERTa large + NLI + Mean Pooling | 76.26 | 74.35 | 54.54 | 78.69 | 74.01 | 46.36 |
|             | RoBERTa base + NLI + Mean Pooling | 74.58 | 76.05 | 37.74 | 77.09 | 74.44 | 39.48 |
|             | ALBERT xxlarge + NLI + Mean Pooling | 78.56 | 77.78 | 45.44 | 79.55 | 75.37 | 41.33 |
|             | ALBERT xlarge + NLI + Mean Pooling | 77.68 | 73.90 | 54.90 | 80.12 | 74.78 | 57.68 |
| Supervised  | BERT large + STSB+ Mean Pooling | 84.63 | 80.97 | 64.69 | 85.25 | 78.48 | 60.48 |
|             | BERT base + STSB+ Mean Pooling | 84.18 | 82.16 | 67.79 | 85.04 | 78.43 | 64.02 |
|             | RoBERTa large + STSB+ Mean Pooling | 85.54 | 82.32 | 71.98 | 86.42 | 78.40 | 66.78 |
|             | RoBERTa base + STSB+ Mean Pooling | 84.26 | 81.78 | 50.57 | 85.26 | 77.43 | 46.46 |
|             | ALBERT xxlarge + STSB + Mean Pooling | 92.58 | 85.33 | 73.12 | 90.22 | 80.03 | 67.54 |
|             | ALBERT xlarge + STSB + Mean Pooling | 91.59 | 83.28 | 71.45 | 90.73 | 81.52 | 66.97 |

B. RESULTS AND DISCUSSION

The correlation between the similarity scores provided by human annotators and the similarity scores calculated by the models is used as the measure to estimate the performance of the word embedding models. Since both Pearson’s correlation and Spearman’s correlation are used by the authors of the models chosen for comparison, we depict our results using both correlations. The results are categorized into three different sections based on the level of supervision used for the models. The first section records the performance of the unsupervised word embedding models, the word2vec and GloVe, and their variations trained on the ‘computer science corpus’. The second section shows the performance of the two variants of BERT, RoBERTa, and ALBERT models, fine-tuned using the AllNLI dataset (Natural Language Inference dataset closely similar to semantic similarity tasks), we consider this as partial supervision since the text entailment tasks are similar to semantic similarity tasks, but the model has not been trained on sentences from any of the three datasets used for comparison. The third section comprises the results of the same models trained (fine-tuned) using both AllNLI and the STS dataset. The results are provided in TABLE 5.

The performance of the word-embeddings decreases considerably when tested on the proposed dataset, indicating that the increase in complexity of the text has a significant and inverse effect on the performance of widely used text embedding models. While the ALBERT-xxlarge model achieves the best performance in the proposed dataset with 73.67% Pearson’s Correlation and 67.54% Spearman’s correlation, it is clear that these results are sub-par in comparison to the 92.58% correlation achieved in the existing benchmark dataset. It is important to note that, though the sentences in the proposed dataset are definitions of topics from a particular domain, they are derived from sources that are commonly used by everyone. Hence these sentences are comparatively simpler than the sentences in scientific or academic articles. Hence, it is evident that there is an imminent need to explore avenues to improve the quality of these word embeddings to capture the semantics in complex documents. We also see that the BERT models fine-tuned to a specific NLP task like semantic similarity do not perform as effectively in the proposed dataset indicating the impact of the training data used in the fine-tuning process. Researchers have highlighted the need for building better datasets to ensure the robustness of the embedding models. Rogers et al. [39] mentions that “As with any optimization method, if there is a shortcut in the data, we have no reason to expect BERT to not learn it. But harder datasets that cannot be resolved with shallow heuristics are unlikely to emerge if their development is not as valued as modeling work.”. In the future, understanding the need for versatile and harder datasets we intend to repeat the process and add more sentences to the dataset thus building a large complex sentence dataset that can, in turn, be used by transformer-based models for fine-tuning.

VI. CONCLUSION

Measuring semantic similarity between text data has been one of the most challenging tasks in the field of Natural Language Processing. Various word embedding models have been proposed over the years to capture the semantics of the words in numeric representations. In this article, a new dataset with sentences that are more complex than the existing benchmark datasets is built, and a read-ability analysis is performed to prove the increase in complexity. A comparative analysis of various word embedding and language models is performed across the two existing benchmark datasets i.e., the STS dataset and SICK dataset, and the proposed dataset. The results show that while the models achieve near-perfect results in the existing benchmark datasets the correlation drops significantly in the proposed dataset, indicating the impact of the increase in complexity of the sentences on the performance of the existing models.
ACKNOWLEDGMENT

The authors would like to extend our gratitude to the research team in the DaTALab at Lakehead University for their support, Punardeep Sikka and Arunim Garg for their feedback and revisions on this publication. We would also like to thank Lakehead University, CASES and the Ontario Council for Articulation and Transfer, without their support this research would not have been possible.

REFERENCES

[1] D. Chandrasekaran, “Exploiting semantic similarity models to automate transfer credit assessment in academic mobility,” 2021.
[2] M. Mohamed and M. Oussalah, “SRL-ESA-TextSum: A text summarization approach based on semantic role labeling and explicit semantic analysis,” Information Processing & Management, vol. 56, no. 4, pp. 1356–1372, 2019.
[3] Y. Liu, Z. Liu, T.-S. Chua, and M. Sun, “Topical word embeddings,” in Twenty-ninth AAAI conference on artificial intelligence. Citeseer, 2015.
[4] E. Sulem, O. Abend, and A. Rappoport, “Semantic structural evaluation for text simplification,” in Proceedings of NAACL-HLT, 2018, pp. 685–696.
[5] J. Wieting, T. Berg-Kirkpatrick, K. Gimpel, and G. Neubig, “Beyond bleu: Training neural machine translation with semantic similarity,” in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 4344–4355.
[6] A. Bordes, S. Chopra, and J. Weston, “Question answering with subgraph embeddings,” in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 615–620.
[7] S. Kim, N. Fiorini, W. J. Wilbur, and Z. Lu, “Bringing the gap: Incorporating a semantic similarity measure for effectively mapping PubMed queries to documents,” Journal of biomedical informatics, vol. 75, pp. 122–127, 2017.
[8] D. Chandrasekaran and V. Mago, “Evolution of Semantic Similarity-A Survey,” ACM Computing Surveys (CSUR), vol. 54, no. 2, pp. 1–37, 2021.
[9] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” arXiv preprint arXiv:1301.3781, 2013.
[10] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.
[11] J. Gorman and J. R. Curran, “Scaling distributional similarity to large corpora,” in Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, 2006, pp. 361–368.
[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, pp. 4171–4186.
[13] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “RoBERTa: A robustly optimized BERT pretraining approach,” arXiv preprint arXiv:1907.11692, 2019.
[14] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, “Albert: A lite bert for self-supervised learning of language representations,” in International Conference on Learning Representations, 2019.
[15] Y. Shao, “Haci at semeval-2017 task 1: Use convolutional neural network to evaluate semantic textual similarity,” in Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), 2017, pp. 130–133.
[16] M. Marelli, S. Menini, M. Baroni, L. Bentivogli, R. Bernardi, R. Zamparelli et al., “A SICK cure for the evaluation of compositional distributed semantic models,” in LREC, 2014, pp. 216–223.
[17] J. Al Qundus, A. Paschke, S. Gupta, A. M. Alzoubay, and M. Youssef, “Exploring the impact of short-text complexity and structure on its quality in social media,” Journal of Enterprise Information Management, 2020.
[18] A. Heppner, A. Pawar, D. Kivi, and V. Mago, “Automating articulation: Applying natural language processing to post-secondary credit transfer,” IEEE Access, vol. 7, pp. 48 295–48 306, 2019.
[19] G. A. Miller, “Wordnet: A lexical database for English,” Communications of the ACM, vol. 38, no. 11, pp. 39–41, 1995.
This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
### APPENDIX A COMPLEX SENTENCE DATASET

| SNo | Sentence 1                                                                 | Sentence 2                                                                 | Final Similarity out of 5 |
|-----|---------------------------------------------------------------------------|---------------------------------------------------------------------------|--------------------------|
| 1   | The study of computation and information.                                  | The study of manipulating, managing, transforming and encoding information. | 3.60                     |
| 2   | The study of computation and information.                                  | A branch of science that deals with the theory of computation or the design of computers | 3.67                     |
| 3   | A program can be a plan of how to do something                            | A branch of science that deals with the theory of computation or the design of computers | 0.87                     |
| 4   | A finite sequence of well-defined, computer-implementable instructions, typically to solve a class of problems or to perform a computation | A specific set of instructions or steps on how to complete a task.         | 4.00                     |
| 5   | A finite sequence of well-defined, computer-implementable instructions, typically to solve a class of problems or to perform a computation | A procedure for solving a mathematical problem in a finite number of steps that frequently involves repetition of an operation | 3.20                     |
| 6   | A data organization, management, and storage format that enables efficient access and modification. | A specific set of instructions or steps on how to complete a task.         | 0.87                     |
| 7   | A set of instructions used to control the behavior of a machine.          | A program can be a plan of how to do something                              | 2.13                     |
| 8   | A set of instructions used to control the behavior of a machine.          | A sequence of coded instructions that can be inserted into a mechanism     | 3.27                     |
| 9   | The study of manipulating, managing, transforming and encoding information. | A sequence of coded instructions that can be inserted into a mechanism     | 1.87                     |
| 10  | A data organization, management, and storage format that enables efficient access and modification. | The organization and implementation of values and information.             | 2.47                     |
| 11  | A procedure for solving a mathematical problem in a finite number of steps that frequently involves repetition of an operation | Various methods or formats for organizing data in a computer.               | 0.60                     |
| 12  | The organization and implementation of values and information.            | Various methods or formats for organizing data in a computer.               | 2.53                     |
| 13  | The intelligence demonstrated by machines, unlike the natural intelligence displayed by humans and animals. | The ability of a computer program or a machine to think and learn.         | 3.67                     |
| 14  | The intelligence demonstrated by machines, unlike the natural intelligence displayed by humans and animals. | A branch of computer science dealing with the simulation of intelligent behavior in computers | 3.07                     |
| 15  | The ability of a computer program or a machine to think and learn.       | A branch of computer science dealing with the simulation of intelligent behavior in computers | 3.67                     |
| 16  | The process of designing and building an executable computer program to accomplish a specific computing result. | A software that controls the operation of a computer and directs the processing of programs | 2.40                     |
| 17  | The process of designing and building an executable computer program to accomplish a specific computing result. | The process of preparing an instructional program for a device              | 3.00                     |
| 18  | The process of telling a computer to do certain things by giving it instructions. | The process of preparing an instructional program for a device              | 3.40                     |
| 19  | A system software that manages computer hardware, software resources, and provides common services for computer programs. | A group of computer programs, including device drivers, kernels, and other software that lets people interact with a computer. | 2.80                     |
| 20  | A system software that manages computer hardware, software resources, and provides common services for computer programs. | The process of telling a computer to do certain things by giving it instructions. | 1.73                     |
| 21  | A group of computer programs, including device drivers, kernels, and other software that lets people interact with a computer. | A software that controls the operation of a computer and directs the processing of programs | 3.27                     |
| 22  | An organized collection of data, generally stored and accessed electronically from a computer system. | A system for storing and taking care of data                               | 3.67                     |
| 23  | An organized collection of data, generally stored and accessed electronically from a computer system. | A set of rules and methods that describe the functionality, organization, and implementation of computer systems. | 1.60                     |
| 24  | A system for storing and taking care of data.                             | A large collection of data organized especially for rapid search and retrieval | 2.87                     |
| 25  | A set of rules and methods that describe the functionality, organization, and implementation of computer systems. | The conceptual design and fundamental operational structure of a computer system. | 3.27                     |
| 26  | A large collection of data organized especially for rapid search and retrieval | The manner in which the components of a computer or computer system are organized and integrated | 1.13                     |
| 27  | The conceptual design and fundamental operational structure of a computer system. | The manner in which the components of a computer or computer system are organized and integrated | 3.93                     |
| 28  | An informal high-level description of the operating principle of a computer program or other algorithm. | A description of the source code of a computer program or an algorithm in a language easily understood by humans. | 3.93                     |
| 29  | An informal high-level description of the operating principle of a computer program or other algorithm. | A formal language comprising a set of instructions that produce various kinds of output. | 2.07                     |
| 30  | A description of the source code of a computer program or an algorithm in a language easily understood by humans. | A program code unrelated to the hardware of a particular computer and requiring conversion to the code used by the computer before the program can be used. | 2.80                     |
| SNo | Sentence_1                                                                 | Sentence_2                                                                 | Final Similarity |
|-----|---------------------------------------------------------------------------|---------------------------------------------------------------------------|-----------------|
| 31  | a formal language comprising a set of instructions that produce various kinds of output. | a type of written language that tells computers what to do in order to work. | 3.60            |
| 32  | A program code unrelated to the hardware of a particular computer and requiring conversion to the code used by the computer before the program can be used. | various high-level languages used for computer programs                    | 2.33            |
| 33  | a type of written language that tells computers what to do in order to work. | various high-level languages used for computer programs                    | 3.80            |
| 34  | The discovery, interpretation, and communication of meaningful patterns in data. | The process of discovering, interpreting, and communicating significant patterns in data to get meaningful information. | 4.27            |
| 35  | The discovery, interpretation, and communication of meaningful patterns in data. | A branch of information technology which is intended to protect computers. | 0.60            |
| 36  | The protection of computer systems and networks from the theft of or damage to their hardware, software, or electronic data. | A branch of information technology which is intended to protect computers. | 3.73            |
| 37  | The protection of computer systems and networks from the theft of or damage to their hardware, software, or electronic data. | The process of discovering, interpreting, and communicating significant patterns in data to get meaningful information. | 0.33            |
| 38  | A type of computer program that, when executed, replicates itself by modifying other computer programs and inserting its own code. | A program that is able to copy itself when it is run and can also execute instructions that cause harm. | 4.27            |
| 39  | The practice of storing regularly used computer data on multiple servers that can be accessed through the Internet | A computer program that is usually disguised as an innocuous program or file, that often produces copies of itself and inserts them into other programs performing a malicious action | 0.13            |
| 40  | The on-demand availability of computer system resources, especially data storage and computing power, without direct active management by the user. | Computing services provided by a company or place outside of where they are being used. | 2.73            |
| 41  | The on-demand availability of computer system resources, especially data storage (cloud storage) and computing power, without direct active management by the user. | The practice of storing regularly used computer data on multiple servers that can be accessed through the Internet | 2.60            |
| 42  | Computing services provided by a company or place outside of where they are being used. | A type of computer program that, when executed, replicates itself by modifying other computer programs and inserting its own code. | 0.47            |
| 43  | A program that is able to copy itself when it is run and can also execute instructions that cause harm. | A computer program that is usually disguised as an innocuous program or file, that often produces copies of itself and inserts them into other programs performing a malicious action | 4.33            |
| 44  | A piece of computer hardware or software that provides functionality for other programs or devices, called "clients". | A computer that serves many kinds of information to a user or client machine. | 3.60            |
| 45  | A piece of computer hardware or software that provides functionality for other programs or devices, called "clients". | A computer in a network that is used to provide services to other computers in the network | 2.87            |
| 46  | A network security system that monitors and controls incoming and outgoing network traffic based on predetermined security rules. | A piece of software that monitors the network traffic between the inside and outside. | 4.20            |
| 47  | A network security system that monitors and controls incoming and outgoing network traffic based on predetermined security rules. | A computer hardware or software that prevents unauthorized access to private data by outside computer users | 3.40            |
| 48  | A computer that serves many kinds of information to a user or client machine. | A piece of software that monitors the network traffic between the inside and outside. | 1.20            |
| 49  | A computer in a network that is used to provide services to other computers in the network | A computer hardware or software that prevents unauthorized access to private data by outside computer users | 0.87            |
| 50  | A data point that differs significantly from other observations. | A statistical observation that is markedly different in value from the others of the sample | 4.13            |

**TABLE 6: Complex Sentence Dataset**
APPENDIX B SURVEY ENVIRONMENTS

FIGURE 5: Survey using Google forms

FIGURE 6: Survey using Amazon Mechanical Turk

APPENDIX C SEMANTIC SIMILARITY DATASETS

TABLE 7: Popular Benchmark datasets for Semantic similarity

| Dataset Name | Word/Sentence pairs | Similarity score range | Year | Reference |
|--------------|----------------------|------------------------|------|-----------|
| R&G          | 65                   | 0-4                    | 1965 | [25]      |
| M&C          | 30                   | 0-4                    | 1991 | [30]      |
| WS353        | 353                  | 0-10                   | 2002 | [41]      |
| LiSent       | 65                   | 0-4                    | 2007 | [42]      |
| SRS          | 30                   | 0-4                    | 2007 | [43]      |
| WS353-Sim    | 203                  | 0-10                   | 2009 | [44]      |
| STS2012      | 5250                 | 0-5                    | 2012 | [45]      |
| STS2013      | 2250                 | 0-5                    | 2013 | [46]      |
| WF300        | 300                  | 0-1                    | 2013 | [47]      |
| STS2014      | 3750                 | 0-5                    | 2014 | [48]      |
| SL7576       | 7576                 | 1-5                    | 2014 | [49]      |
| SimLex-999   | 999                  | 0-10                   | 2014 | [50]      |
| SICK         | 10000                | 1-5                    | 2014 | [51]      |
| STS2015      | 3000                 | 0-5                    | 2015 | [52]      |
| SimVerb      | 3500                 | 0-10                   | 2016 | [53]      |
| STS2016      | 1186                 | 0-5                    | 2016 | [54]      |
| WiC          | 5428                 | NA                     | 2019 | [55]      |
APPENDIX D COMPLEXITY GRADE LEVEL AND EXAMPLE SENTENCES FROM STS DATASET

TABLE 8: Sample sentences for various grade levels from STS dataset

| Grade level | Example                                                                 |
|-------------|-------------------------------------------------------------------------|
| 2           | A plane is taking off                                                   |
| 3           | A man pours oil in a pot                                                |
| 4           | A man is playing on a guitar and singing                                |
| 5           | A woman picks up and holds a baby kangaroo                              |
| 6           | Two men are packing suitcases into the trunk of a car                   |
| 7           | Suicide attack kills eight in Baghdad                                   |
| 8           | A woman is riding a motorized scooter down a road.                      |
| 9           | Russia praises idea to render rights assistance to Ukraine             |
| 10          | Israel confirms strike on Syria targeted weapons                        |
| 11          | Ankeet Chavan granted conditional bail for marriage                    |
| 12          | France warns of extremists benefiting from Egypt violence              |
| 13          | Redford touts Alberta economy ahead of vote on leadership              |
| 14          | Jury nullification has nothing to do with the Constitutionality of laws.|
| 17          | Page said it’s also possible that genes on the Y chromosome may influence gender-specific differences in disease susceptibility. |

***