applied sciences

MDPI

Article

From General Language Understanding to Noisy Text Comprehension

Buddhika Kasthuriarachchy [1,*], Madhu Chetty [1], Adrian Shatte [1] and Darren Walls [2]

[1] School of Engineering, Information Technology and Physical Sciences, Federation University Australia, Churchill, VIC 3842, Australia; madhu.chetty@federation.edu.au (M.C.); a.shatte@federation.edu.au (A.S.)
[2] Global Hosts Pty Ltd., Melbourne, VIC 3463, Australia; darren@sportshosts.com
* Correspondence: b.kasthuriarachchy@federation.edu.au

Abstract: Obtaining meaning-rich representations of social media inputs, such as Tweets (unstructured and noisy text), from general-purpose pre-trained language models has become challenging, as these inputs typically deviate from mainstream English usage. The proposed research establishes effective methods for improving the comprehension of noisy texts. For this, we propose a new generic methodology to derive a diverse set of sentence vectors combining and extracting various linguistic characteristics from latent representations of multi-layer, pre-trained language models. Further, we clearly establish how BERT, a state-of-the-art pre-trained language model, comprehends the linguistic attributes of Tweets to identify appropriate sentence representations. Five new probing tasks are developed for Tweets, which can serve as benchmark probing tasks to study noisy text comprehension. Experiments are carried out for classification accuracy by deriving the sentence vectors from GloVe-based pre-trained models and Sentence-BERT, and by using different hidden layers from the BERT model. We show that the initial and middle layers of BERT have better capability for capturing the key linguistic characteristics of noisy texts than its latter layers. With complex predictive models, we further show that the sentence vector length has lesser importance to capture linguistic information, and the proposed sentence vectors for noisy texts perform better than the existing state-of-the-art sentence vectors.

Keywords: sentence representation; probing tasks; language understanding; noisy text

check for updates

Citation: Kasthuriarachchy, B.; Chetty, M.; Shatte, A.; Walls, D. From General Language Understanding to Noisy Text Comprehension. *Appl. Sci.* **2021**, *11*, 7814. https://doi.org/10.3390/app11177814

Academic Editor: Maxim Mozgovoy

Received: 30 June 2021
Accepted: 23 August 2021
Published: 25 August 2021

1. Introduction

Natural Language Processing (NLP) and its subfield, Natural Language Understanding (NLU), primarily focuses on the well-known complex problem of machine reading comprehension. Among several challenges facing NLU, the representation of sentences incorporating all their linguistic elements is considered to be highly complex. Due to the benefit of accurate sentence representations, e.g., sentence classification, text summarization, and machine translation, it has become necessary to explore new NLU methods that incorporate linguistic components, such as syntax and semantics, to improve accuracy. While a plethora of techniques have already been proposed, representing sentences as vectors of real numbers in high dimensional continuous space is still attracting attention [1,2].

For vector representation, both word and sentence embeddings have influenced the representation, following the rapid rise of Word2Vec [3]. Recently, unsupervised, pre-trained language models, such as Bidirectional Encoder Representations from Transformers (BERT) [4], were successful in achieving state-of-the-art results in various NLP tasks, e.g., at the sentence level, thereby introducing a major paradigm shift in sentence representations. It may be noted that unlike shallow word vector models (i.e., Word2Vec [3] and Global Vectors for Word Representation (GloVe) [5]), deep models, such as BERT, are contextual.

Widespread use cases, such as sentiment analysis and intent analysis, mandate sophisticated sentence representations since these models essentially involve the identification of intricate linguistic patterns [6,7]. With the increasing proliferation of social media data, such as Tweets, it has further become inevitable to represent noisy texts as vectors to improve the model performance. For this reason, the BERT model is extensively used with Tweets to achieve state-of-the-art accuracy [8–11].

However, the application of pre-trained language models, such as BERT, in such scenarios is not easy because Tweets follow a different distribution [12,13] than the training inputs. While the BERT model is pre-trained on BookCorpus and English Wikipedia, the Tweets exhibit a significant deviation from this mainstream English language usage. Further, such challenges become extremely overwhelming, as Tweets cover different domains (e.g., day-to-day activities, sports, politics, and science); hence, they are significantly different. For these reasons, the language representation should clearly express non-task-specific general-purpose priors to develop artificially intelligent systems [14].

Although BERT is a general-purpose language model, the reason behind its overall success is not understood clearly. Goldberg [15] and Jawahar et al. [16] made efforts to understand BERT's ability to learn the structure and syntax of the English language. It was observed that different layers and regions of BERT capture different traits of the English language. However, it was not reported how these findings can enhance the quality of word or sentence embeddings. Indeed, Kumar et al. [17] demonstrated a drastic fall in BERT's performance with an increase in noise level. Apart from this, there was also the recent emergence of various pre-trained language models comprising multi-layer architectures [18]. Thus, a technique based on the latent representations of multi-layer models is vital for optimizing the vector representations to be used for use cases involving unstructured and noisy texts.

To address these research gaps, we use BERT as the multi-layer pre-trained language model and Tweets to represent noisy texts. We propose a systematic approach to derive a diverse set of sentence vectors combining and extracting various linguistic characteristics. For this, we have developed new probing datasets, using noisy texts based on the definition of specific probing tasks in [19] to analyze BERT's behavior across different linguistic territories centered on noisy texts. We derive generalizable sentence representations for noisy texts, comprising the most important linguistic characteristics to capture the meaning of a sentence. More specifically, our key contributions for enabling BERT in deriving meaning-rich sentence representation from the noisy text are as follows:

- New *noisy* probing datasets: This new dataset can serve as benchmark datasets for future researchers to study the linguistic characteristics of unstructured and noisy texts. These datasets are available in the public domain (https://bit.ly/3rK0g7P) and available on request.
- New methodology: this allows studying the linguistic comprehension of multi-layer language models.
- Generic technique: used for sentence vector generation, using a pre-trained multi-layer language model.

The rest of this paper is organized as follows. Section 2 provides relevant background information related to BERT's language understanding ability and probing tasks. Section 3 discusses the probing dataset generation approach and the strategy to generate sentence embeddings. Section 4 presents various experimental results across different probing tasks. The results are analyzed and discussed in Sections 5 and 6, respectively. Finally, Section 7 presents the conclusion.

2. Background

2.1. Pre-Trained Language Models

Recently, word embedding [20] has become popular as a de facto starting point for representing the meaning of words. However, static methods, such as Word2Vec [3], GloVe [5], and FastText [21] generally generate fixed word representations in a vocabulary. Hence, these techniques cannot easily be adapted to identify the contextual meaning of a word. Recent discoveries of dynamic, pre-trained language representations, such as ELMo, a deep contextualized word representation [22], and BERT [4] produce dynamic representations of a word based on its context. The BERT architecture includes a multi-layer bidirectional Transformer [23] and an attention mechanism that learns contextual relations between words (or sub-words) in a text. The Transformer consists of two separate mechanisms—an encoder that processes the input, and a decoder that generates a prediction for the task. BERT, which is trained bidirectionally on a large corpus of unlabeled text, including the entirety of Wikipedia and BookCorpus, allows its models to understand the meaning of a language more correctly.

Further, several other Transformer-based language models perform well at a broader range of tasks beyond document classification, such as commonsense reasoning, semantic similarity, and reading comprehension. Transformer-XL [24], a Transformer-based auto-regressive model, enables capturing longer-term dependencies in a sentence and achieves better performance on NLP tasks for both short and long sequences. Generative Pre-trained Transformer 3 (GPT-3) [25], the third generation language prediction model in the GPT-n series created by OpenAI, is an auto-regressive Transformer model that performs reasonably well on unseen NLP tasks.

These recent models capture many facets of language relevant for downstream tasks, such as long-term dependencies, hierarchical relations, and context, to provide state-of-the-art performance [15,26]. Further, previous research [20,27,28] demonstrated that deep learning models with complex architectures that leverage the contextual meaning of the words can significantly improve the learning abilities.

2.2. *Language Understanding with BERT*

Goldberg [15] assesses the extent to which the BERT model captures the syntactic structure of a sentence, using three stimuli tasks related to subject–verb agreement. Though the results are not directly comparable with previous work, due to BERT's bidirectional nature, the results suggest that purely attention-based BERT models are likely capable of capturing syntactic information at least as well as the sequence models, and probably better.

Jawahar et al. [16] performed a series of experiments, using conventional and standard English sentences extracted from books, to identify the linguistic information learned by BERT. These experiments were based on the probing datasets developed by [19], using the Toronto BookCorpus dataset [29], which was one of the two data sources used to train the BERT model. They showed that BERT's intermediate layers encode a rich set of linguistic characteristics, with surface features at the bottom, syntactic features in the middle, and semantic features at the top. This indicates that specific regions or layers of BERT are better suited for comprehending different aspects of the English language.

Similarly, Liu et al. [30] examined the linguistic knowledge captured by contextual word representations derived from different layers of large-scale neural language models. They showed that the frozen contextual representations are competitive with state-of-the-art, task-specific models in many cases but fail on tasks requiring fine-grained linguistic knowledge. These studies focused only on structured and clean English sentences. They paid little attention to combining the layer representations based on linguistic knowledge to derive a meaning-rich sentence vector. Tenny et al. [31] introduced "edge probing" tasks, covering syntax, semantic meaning and dependency relations phenomena to study how contextual representations encode sentence structures. Their results using BERT and a few other pre-trained language models concluded that these models encode syntactic phenomena strongly but demonstrate comparable minor improvements on semantic tasks, compared to a non-contextual baseline. However, they worked only with the top layer activations of the BERT model. Further, Hewitt and Manning [32] showed that the contextual word representations provided by pre-trained language models, such as BERT, embed syntax trees in their vector representations. Nevertheless, they focused mainly on the syntactic structure.

On the other hand, Clark et al. [26] analyzed BERT's attention mechanism and showed that a specific set of attention heads correspond well to linguistic notions of syntax and coreference. Further, they demonstrated the ability of BERT's attention heads to capture important syntactic information, using an attention-based probing classifier.

However, Wang et al. [33] more recently concluded that the popular complex pre-trained language models do not necessarily translate noisy text to better representations. Further, they highlighted that more exploration is needed in this area.

2.3. Probing Tasks

Shi et al. [34] and Adi et al. [35] introduced general prediction tasks to understand the language information captured by sentence vectors. Shi et al. [34] investigated whether Neural Machine Translation (NMT) systems learn source language syntax as a by-product of training by analyzing the syntactic structure as a by-product of training. Adi et al. [35] proposed a framework that facilitates a better understanding of the encoded representations, using tasks to predict a sentence's length, detect a change in word orders, and identify the words in a sentence.

Extending the work of [19,34,35] has introduced ten classification problems known as *probing tasks*. As we know, a probing task is a text classification problem that focuses on a grouping of sentences based on simple linguistic characteristics of sentences. The performance of this classification model depends on the richness of the linguistic information packed into a sentence representation. Further, these probing tasks are assigned to three groups: surface information, syntactic information, and semantic information, based on the primary linguistic feature required to perform the task effectively. The surface information tasks can rely only on surface properties (e.g., sentence length) to perform the classification successfully, and no linguistic knowledge is required. The tasks grouped under syntactic information are sensitive to a sentence's syntactic properties (e.g., depth of the syntactic tree). In contrast, semantic information-related tasks require some understanding of the meaning of a sentence and the semantic structure.

3. Methodology

This section introduces our methodology for leveraging probing tasks to efficiently validate BERT's ability to capture linguistic information and to derive meaning-rich sentence representations for noisy and unstructured text.

We propose a systematic approach to study the linguistic behaviors of multi-layer pre-trained language models by dividing the layers into multiple regions. Hence, in our methodology, we introduce a novel technique to generate sentence embeddings by bisecting BERT into three regions (Figure 1) and then combining the hidden layers and token vectors, using two pooling operations. This allows us to analyze a diverse set of sentence vectors and their ability to capture linguistic information representing different linguistic domains. Next, we discuss our approach to generate probing datasets covering five probing tasks under noisy text conditions. These noisy probing datasets are crucial in determining each sentence vector's ability to capture necessary linguistic patterns to classify sentences to the target classes of each probing task. This framework can be easily extended to study the language comprehension capabilities of similar multi-layer language models.

The details of the methodology and its components are presented below.

3.1. Sentence Vector Generation

Our proposed methodology uses pre-trained language models to generate sentence representations. We use the "$\text{BERT}_{\text{BASE}}$-uncased" model [4] to obtain word embeddings from different hidden layers to produce sentence vectors. This allows for exploration of the linguistic features of unstructured and noisy text, such as Tweets, as learned by different hidden layers of the BERT model. Further, to link BERT's learning ability with specific linguistic components, inspired by the work of Jawahar et al. [16], we divide BERT's hidden layers into three regions as shown in Figure 1. Jawahar et al. [16] showed that BERT's hidden layers encode a rich hierarchy of linguistic information, with surface features at the bottom, syntactic features in the middle and semantic features at the top. These linguistic components are crucial to represent the meaning of a sentence. Hence, in our methodology, we propose a novel technique to generate region-wise sentence embeddings by bisecting BERT into three regions.

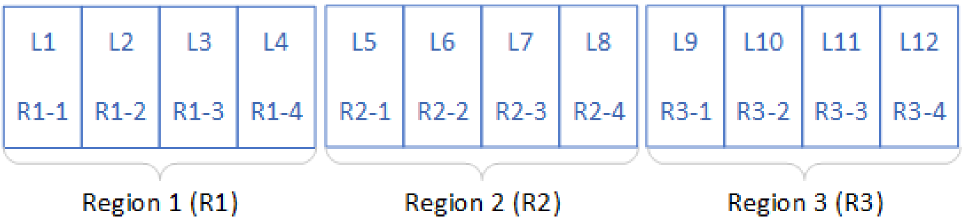

Figure 1. The twelve layers (L1 to L12) of the BERT$_{\text{BASE}}$ model are partitioned into 3 regions. Rn-i represents the ith layer in the nth region.

Further, apart from this, we use pre-trained Word2Vec [3] and Stanford's GloVe [5] models to derive sentence vectors. In contrast to BERT, these models, although shallow and non-contextual, offer 10 to 100 times more vocabulary, thereby providing a vibrant vocabulary. For this reason, it is possible that this may outweigh the benefits of a context-aware pre-trained model with a minimal vocabulary (e.g., BERT), especially for noisy data [36]. Moreover, we employ a word vector trained with the GloVe algorithm, using two billion Tweets to evaluate the impact of Twitter-specific pre-trained language models.

The following section explains the strategy to generate multiple sentence embeddings, using the pre-trained BERT$_{\text{BASE}}$-uncased model. It may be noted that for the remaining paper, the term BERT is used to represent BERT$_{\text{BASE}}$-uncased.

3.1.1. Sentence Representations Using Multi-Layer Pre-Trained Language Models

An input sentence is represented as a set of input tokens $T = [t_0, t_1, \ldots, t_n]$, where t$_0$ is the special $[CLS]$ token that needs to be prepended for the out-of-the-box pooling schema to work. BERT produces a set of hidden layer activations $H^0, H^{(1)}, \ldots, H^{(L)}$, where $H^{(l)} = [h_0^{(l)}, h_1^{(l)}, \ldots, h_n^{(l)}]$ are the activation vectors of the lth hidden layer. We have ignored the H_0, which consists of non-contextual word-piece embeddings, to generate sentence representations.

To generate a sentence representation based on multiple hidden layers, we propose to generate token representation vector w_i for each token t_i in T, using a *layer pooling* strategy. A layer pooling strategy combines different representations of the same token across multiple hidden layers. For this, three layer pooling strategies are studied: (i) SUM-layer-strategy, (ii) MEAN-layer-strategy, and (iii) CONCAT-layer-strategy. The SUM-layer-strategy and the MEAN-layer-strategy calculate the sum and mean of all the activation vectors $h_i \in \mathbb{R}^d$ of the selected hidden layers, respectively, producing $w_i \in \mathbb{R}^d$, where d is the size of the hidden vector h. Thus, for each sentence, the Mean-layer-strategy and SUM-layer-strategy produce a matrix $W \in \mathbb{R}^{n \times d}$. On the other hand, the CONCAT-layer-strategy concatenates the corresponding hidden activation vectors h_i in the order of the layer numbers to generate $w_i \in \mathbb{R}^{kd}$, where k is the number of BERT layers selected to generate the sentence representation. The CONCAT-layer-strategy produces a sentence representation $W \in \mathbb{R}^{n \times nd}$.

Then, to derive the sentence vector $S = [s_1, s_2, \ldots, s_{||w_i||}]$, we apply multiple *token pooling* strategies for the sentence representation W (obtained after applying the layer pooling strategy), where each token representation w_i is a row. A token pooling strategy merges all the token embeddings of a sentence into a singe vector. For this, we study two token pooling operations: (i) MEAN-token-strategy, and (ii) MAX-token-strategy. MEAN-token-strategy and MAX-token-strategy are calculated as $s_j = \underset{1 \le j \le n}{\mathbb{E}} W_{ij}$ and $s_j = \max\limits_{1 \le j \le n} W_{ij}$, respectively. Further, the MEAN–MAX-token-strategy we propose concatenates the MEAN-token-strategy output vector and the MAX-token-strategy output to derive a sentence vector twice the size of w_i.

As shown in Figure 1, for each region Rn ($n \in 1,2,3$), different combinations of four layers are considered to generate sentence embeddings. We apply the layer pooling and token pooling strategy combinations listed in Table 1 across each BERT region Rn to systematically generate a diverse set of sentence embeddings, using the pre-trained BERT model.

Table 1. Strategy to generate sentence embeddings from each region (ref. Figure 1) of the BERT model. Rn-i represents the ith layer in the nth region. We combine each layer pooling strategy with every token pooling strategy across identified layers to generate multiple sentence embeddings. Layer pooling is not applicable for the sentence embeddings generated using a single vector.

Layers	No. of Layers	Layer Pooling	Token Pooling
Rn-1	1	—	mean, max
Rn-2	1	—	mean, max
Rn-3	1	—	mean, max
Rn-4	1	—	mean, max
Rn-1, Rn-2	2	sum, mean, concat	mean, max
Rn-3, Rn-4	2	sum, mean, concat	mean, max
Rn-1 to Rn-4	4	sum, mean, concat	mean, max

3.1.2. Sentence-BERT

Our experiments also utilize the state-of-the-art sentence embedding model, Sentence-BERT (SBERT) [37], which uses Siamese and triplet network structures to derive semantically meaningful sentence vectors from the pre-trained BERT model. We propose to use a pre-trained model optimized for Semantic Textual Similarity (STS), as this model is recommended for general-purpose use. SBERT uses a mean pooling strategy to derive sentence vectors from word embeddings.

3.1.3. Static Embeddings

We propose two shallow pre-trained models, namely Word2Vec [3] and GloVe [5], to generate sentence vectors for unstructured and noisy sentences, as these language models are rich in vocabulary, compared to BERT. It is known that any social media data, such as Tweets, often lack grammatical structure and can contain misspelled words and acronyms. Hence, a language model (e.g., Word2Vec and GloVe) that ensures a lower percentage of out-of-vocabulary (OOV) words may provide better sentence representations than a deep pre-trained model with a smaller vocabulary [36].

We use the MEAN-token-strategy to derive sentence embeddings, using Word2Vec and GloVe.

3.2. Noisy Probing Datasets

Probing datasets have a crucial role in the proposed study, as they validate the model's ability to comprehend linguistic characteristics. Studies reported earlier (e.g., [19]) have focused only on language comprehension of structured and grammatical sentences. Hence, the existing probing datasets [19] contain structured and grammatical sentences and rely on the pre-trained Probabilistic Context-Free Grammar (PCFG) model [38] and part-of-speech, constituency, and dependency parsing information provided by the Stanford Parser. Although the PCFG model reported close to 87% accuracy for regular English sentences, it is poorly suited for noisy texts [39,40]. Further, the available Twitter-specific dependency parsers reported a low overall accuracy level with further reductions if the test set topics differed from the training dataset. Thus, the use of automatic part-of-speech or automatic dependency parsing as suggested by [19] is not a feasible option for noisy probing datasets. Hence, we propose to use a noisy dataset manually annotated with the required linguistic labels to generate quality probing datasets.

For this, we use "Tweebank v2", a collection of English Tweets [41], annotated in Universal Dependencies [42], as it can be exploited to generate the required noisy probing datasets. Authors of [41] followed a rigorous two-stage process to develop 3550 manually labeled Tweets. They automatically annotated the Tweets, using a parser trained on a sample set of Tweets manually annotated in the first stage. In the second stage, they manually corrected the parsed data. These high-quality labels are crucial to developing gold standard probing datasets for noisy text data. However, this research [41] did not focus on specific aspects of linguistics, such as dependency parsing information. Due to the unavailability of these linguistic labels, we are focusing only on a selected subset of probing tasks out of the ten probing tasks proposed by [19]. Nevertheless, the selected probing tasks continue to cover the three important linguistic categories (i.e., surface, syntactic and semantic), thereby enabling us to analyze the richness of the sentence vectors across all three levels of linguistic information and ensuring the quality of the findings. Further, we introduce additional criteria explained below to adapt the dataset to noisy conditions. The probing tasks that are focused on in this study are explained in the following sections.

Sentence length

In this classification task, the goal is to predict the sentence length in 8 possible bins (0–7) based on their lengths: 0 (5–8), 1 (9–12), 2 (13–16), 3 (17–20), 4 (21–25), 5 (26–29), 6 (30–33), and 7 (34–70). These bins are the same as those proposed earlier [35]. This task is referred to as "SentLen" in the paper.

Word content

We consider a 10-class classification task with ten words as targets, considering the available manually annotated instances. The aim is to predict which of the target words appears in the given sentence. Words that are not part of the vocabulary are split by BERT into subwords and characters. In this case, word embeddings might not reflect the best meaning of the word. Hence, we propose to use only the words that appear in the BERT vocabulary as target words. We construct the data by picking the first ten lower-cased words occurring in the corpus vocabulary ordered by frequency and having a length of at least four characters, as this is a noisy dataset this improves the reliability of the dataset. Each sentence contains a single target word, and the word occurs precisely once in the sentence. The task is referred to as "WC" in the paper.

Bigram shift

The purpose of the Bigram Shift task is to test whether an encoder is sensitive to legal word orders. Two adjacent words in a Tweet are inverted, and the classifier performs a binary classification to identify inverted and non-inverted Tweets. The task is referred to as "BShift" in the paper.

Tree depth

The Tree Depth task evaluates the encoded sentence's ability to understand the hierarchical structure by allowing the classification model to predict the depth of the longest path from the root to any leaf in the Tweet's parser tree. The dataset contains six different classes (two to seven) based on the tree depth. The task is referred to as "TreeDepth" in the paper.

Semantic odd man out

The Tweets are modified by replacing a random noun or a verb o with another noun or verb r. The task of the classifier is to identify whether the sentence gets modified due to this change. The task is called "SOMO" in the paper.

These five probing tasks, covering the three key linguistic information levels, are presented in Table 2.

Table 2. Grouping of probing tasks.

Group	Probing Tasks
Surface information	SentLen, WC
Syntactic information	BShift, TreeDepth
Semantic information	SOMO

3.3. Sentence Vector Evaluation Framework

The most commonly used approach to generate sentence vectors is to average the BERT output layer (BERT embeddings) or to use the output of the first token (the $[CLS]$ token). We extend the common sentence vector generation with our sentence embedding generation technique and combine it with the new probing datasets to develop a sentence vector evaluation framework, as shown in Figure 2. This framework enables us to assess the ability of various sentence vectors to capture linguistic information that can be useful for various downstream tasks. Probing datasets consist of the noisy datasets we developed, using manually annotated Tweets. As discussed in Section 3.1.1, the Embedding Generator generates a diverse set of sentence vectors based on the BERT model while generating sentence vectors using various other pre-trained models. Next, sentence vectors are forwarded to a classification model. We propose to use a Logistic Regression (LR) model and a Multi-Layer Perceptron (MLP) model to analyze the relationship between different sentence vectors and the shallowness or the deepness of the network.

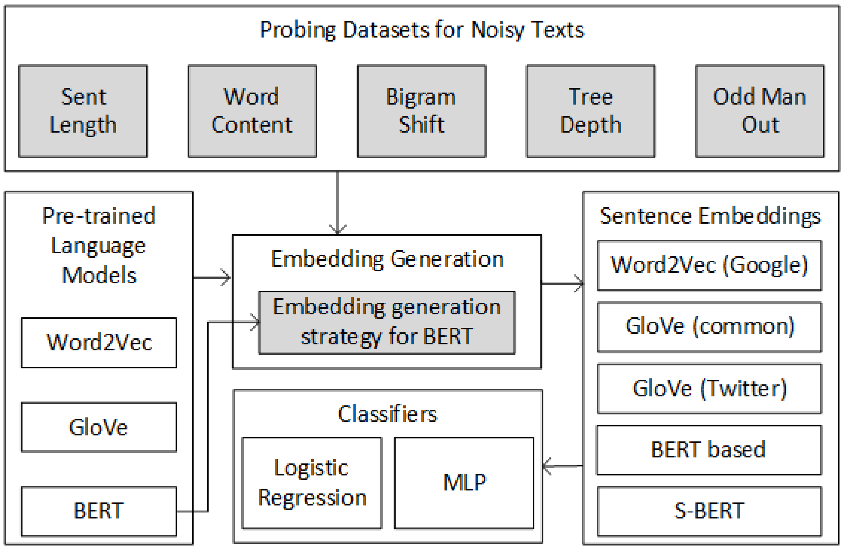

Figure 2. Sentence embedding evaluation framework. For the proposed work, we study BERT for embedding generation. This framework can be easily extended to study other pre-trained language models.

4. Experiments

4.1. Dataset Development

As discussed in Section 3.2, we have developed five different probing datasets for these different probing tasks. The probing datasets are developed based on the Tweebank v2 dataset (https://github.com/Oneplus/Tweebank, accessed on 10 August 2020) developed by [41]. Tweebank v2, a collection of English Tweets annotated in Universal Dependencies [42], is useful since it can be exploited for training NLP systems to enhance their performance on social media texts. Tweebank v2 dataset contains 3550 Tweets, which includes tokenization, part-of-speech-tagging, and labeled Universal Dependencies. This dataset is split into train, development, and test sets as shown in Table 3.

Table 3. Statistics of Tweebank v2.

Split	Tweets	Tokens
train	1639	24,753
development	710	11,742
test	1201	19,112

We use these tokenization, part-of-speech tagging and labeled dependencies to generate five probing datasets as discussed in Section 3.2. Table 4 shows the distribution of Tweets for training, validation and tests in each of the probing datasets. Our splits are based on the original splits of the Tweebank v2 dataset.

Table 4. Probing task datasets.

Dataset	Train	Validation	Test
SentLen	1530	684	1141
WC	439	186	328
BShift	1639	710	1201
TreeDepth	1071	433	757
SOMO	1003	444	727

4.2. Sentence Embedding Generation

As shown in Table 5, we leverage a few commonly used pre-trained language models and the Sentence-BERT embeddings model under each of the base language models discussed in Section 3.1. For training, standard sentences from the Google News dataset and Wikipedia were used for "GoogleNews" and the "glove_6b" pre-trained models while $\text{BERT}_{\text{BASE}}$ model was trained using BookCorpus and Wikipedia data. Similarly, the SBERT-NLI-base sentence transformer was trained on the SNLI [43] dataset, whereas the "glove_twitter" language model was trained with a large number of Tweets.

Table 5. Pre-trained Language Models.

Base Model	Pre-Trained Model	Vocab. Size	Dimension
Word2Vec	GoogleNews	3 M	300
GloVe	glove_6b	400 K	300
GloVe	glove_twitter	1.2 M	200
BERT	BERT-base	30,522	768
S-BERT	SBERT-NLI-base	30,522	768

4.3. Probing Task Classification

We use SentEval toolkit [44] to evaluate different sentence encoders. As in [45], we use a deeper network—MLP and a Logistic Regression classifier—to make the findings more practical while reducing the undesirable side effects, such as preference for embeddings of a larger size. We use the classifier and the validator provided with the SentEval toolkit (https://github.com/facebookresearch/SentEval/, accessed on 12 August 2020) [44] after modifying it to accommodate the proposed sentence embeddings. Following Conneau et al. [44], we use the parameters, shown in Table 6, for Logistic Regression and MLP. However, to cope with the computational constraints, we modify the value of the "batch_size" parameter to 32.

Table 6. Parameters for the Classifiers.

Parameter	LR	MLP
nhid	0	200
optimizer	'adam'	'adam'
batch_size	32	32
tenacity	5	5
epoch_size	4	4

5. Results

This section first analyses the effectiveness of the proposed pooling strategies: layer pooling and token pooling. Next, we analyze the distribution of the language understanding (surface, syntactic and semantic) across the various regions of the BERT model proposed for this study. Finally, we analyze the performance of the sentence vectors generated by combining these findings along with the existing sentence vector generation mechanisms, including the state-of-the-art techniques.

Pooling strategy analysis:

For this study, we consider sentence embeddings derived using all four layers of each BERT region. Table 7 shows the resulting sentence vector sizes for each combination of layer and token pooling strategies when applied to four hidden layers of BERT. The CONCAT-layer-strategy and MEAN–MAX-token-strategy significantly increase the resulting sentence vector size, by four times and two times, respectively. From the results shown in Table 8, we note that the Logistic Regression model achieves the best results with sentence vectors of size 6144, whereas the MLP model achieves the best results, in most cases, with 1536 vector size. From this, it becomes evident that simpler models, such as Logistic Regression, require huge sentence vectors to identify linguistic patterns, while complex models can achieve improved results with significantly lower-sized sentence vectors.

Table 7. Sentence vector sizes derived using different pooling strategies with four hidden layers.

Layer Pooling	Token Pooling	Embedding Size
concat	max	3072
	mean	3072
	mean_max	6144
mean	max	768
	mean	768
	mean_max	1536
sum	max	768
	mean	768
	mean_max	1536

Similarly, Table 9 shows that the Logistic Regression model achieves, in most cases, the best accuracy with the CONCAT-layer-strategy. However, one of the syntactic information groups' tasks and the semantic information task obtains the best results with the MEAN-layer-strategy. On the other hand, the MLP model performs satisfactorily with the MEAN-layer-strategy and SUM-layer-strategy. Both logistic regression and the MLP models prefer the MEAN-MAX-token-strategy or MEAN-token-strategy, while MAX-token-strategy performs poorly across all the performing tasks.

Table 8. Average classification accuracy for different sentence vector sizes derived with four hidden layers using different pooling strategies.

Vector Size	**Region**	**Logistic Regression**					**MLP**				
		SL	**WC**	**BS**	**TD**	**SM**	**SL**	**WC**	**BS**	**TD**	**SM**
768	1	53.29	70.96	67.36	51.29	68.95	61.11	73.48	68.13	55.65	69.26
	2	52.54	58.69	71.47	51.75	70.29	59.91	60.98	71.38	57.17	70.94
	3	47.40	46.11	71.13	50.93	70.77	55.37	49.47	71.75	55.58	73.14
1536	1	56.75	80.18	69.45	55.42	70.15	**68.41**	84.15	68.99	56.61	70.09
	2	56.18	70.58	73.77	53.57	72.70	65.65	71.19	**74.19**	**58.45**	71.73
	3	48.95	54.27	70.15	51.12	**72.97**	58.07	56.56	71.44	56.28	**74.69**
3072	1	54.74	73.48	69.28	51.79	68.23	63.19	74.70	69.15	56.54	59.22
	2	53.42	61.74	72.53	53.10	70.77	61.49	65.40	72.40	58.39	70.70
	3	48.42	50.31	72.36	51.92	72.56	58.33	52.90	71.44	56.01	72.28
6144	1	**59.33**	**82.62**	69.61	**56.01**	70.29	65.03	**86.59**	71.61	56.94	72.35
	2	56.70	69.21	**75.60**	53.50	72.35	64.59	72.26	72.69	58.39	72.21
	3	51.45	55.79	71.61	51.65	72.90	61.35	56.71	72.61	58.39	71.94

Table 9. Mean classification accuracy for sentence vectors derived using different pooling strategies (SL: SentLen, WC: WC, BS: BShift, TD: TreeDepth, SM: SOMO).

Pooling		**Logistic Regression**					**MLP**				
Layer	**Token**	**SL**	**WC**	**BS**	**TD**	**SM**	**SL**	**WC**	**BS**	**TD**	**SM**
concat	max	50.51	55.99	69.50	52.53	69.33	57.35	58.03	69.38	57.11	63.00
	mean	53.87	67.68	**73.27**	52.00	71.71	64.65	70.63	72.61	56.85	71.80
	mean_max	**55.83**	**69.21**	72.27	53.72	71.85	63.66	**71.85**	72.30	57.91	72.17
mean	max	48.85	51.32	67.97	51.78	69.42	56.77	55.08	67.89	56.50	71.43
	mean	54.28	66.67	72.72	52.62	**72.81**	63.02	69.41	72.02	**58.96**	71.34
	mean_max	54.63	68.50	71.77	**54.07**	72.26	63.02	70.33	71.74	56.80	**72.31**
sum	max	47.65	50.00	66.81	49.54	65.75	53.87	50.20	68.83	55.57	70.43
	mean	53.52	66.36	72.44	51.34	72.03	61.53	70.53	**72.94**	53.50	71.25
	mean_max	53.28	68.19	70.47	52.66	71.62	**65.06**	70.94	71.33	57.42	72.03

In the rest of the analyses, the results derived with the MEAN-layer-strategy and MEAN-token-strategy using the MLP classifier are used. This enables easy comparisons of the BERT based sentence embeddings with vectors derived from static pre-trained models by calculating the average of the word embeddings. Further, Sentence-BERT internally uses the mean of the token embeddings to generate sentence embeddings.

Region-wise analysis:

Figure 3 shows a heat map of the accuracies (darker colors equate to higher accuracy) of each probing task with sentence vectors generated using each hidden layer of the BERT model. The SentLen and the WC tasks in the Surface Information group achieves better accuracy with sentence vectors derived from hidden layers in the first region (R1), and the performance gradually decreases as we move toward the last layers of the BERT model. On the other hand, higher accuracies are obtained for the syntactic information tasks—BShift and TreeDepth—with the sentence vectors generated using the hidden layers from the second region (R2). The initial layers of the R2 show the most contribution to the accuracy, while the hidden layers from the R1 contribute poorly to the syntactic information group tasks. Further, the hidden layers that contribute to increasing the sentence vectors' richness for the semantic information task are found at the border of R2 and R3.

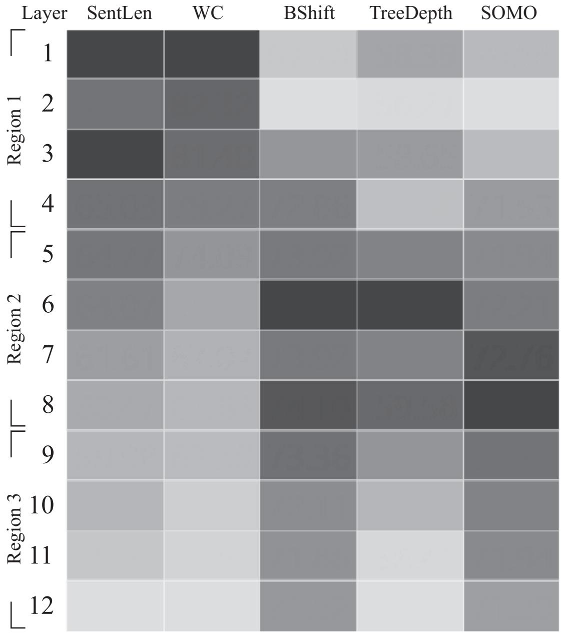

Figure 3. Heat map of probing task accuracy.

Overall, in the context of noisy texts, the hidden layers in the region R1 contain most of the linguistic characteristics required to address probing tasks in the surface group. In contrast, the syntactic and semantic group tasks are able to identify necessary linguistic patterns from R1 and R2. Nevertheless, the sentence vectors' performance derived from hidden layers in the last region (R3) ranges from low to marginal, indicating their inability to capture linguistic information from noisy texts.

Overall accuracy:

Table 10 presents the classification accuracies for probing tasks with sentence vectors derived from GloVe-based pre-trained models, Sentence-BERT and using different hidden layers from the BERT$_{\text{BASE}}$-uncased model. In the context of BERT-based sentence vectors, we have considered sentence vectors derived based on the last hidden layer, the last four hidden layers, and all 12 layers. Devlin et al. [4] achieved comparable results for feature-based by using those layers as input to an artificial recurrent neural network. Based on our findings, we propose two separate approaches for noisy texts. The first is based on BERT's first hidden layer, while the second combines the first hidden layer of each BERT region, i.e., layers 1, 5 and 9 (1-5-9).

The MLP model achieves the best accuracy for all the probing tasks, except for the SOMO task, which is in the semantic information group. The Logistic Regression model has reached the surface information probing tasks' best results with the BERT-based sentence vectors derived only by using the first hidden layer. However, Logistic Regression performs better for the syntactic and semantic information probing tasks with sentence vectors generated using all 12 hidden layers of the BERT model.

On the other hand, the best results for the MLP model are mostly achieved with the sentence vectors derived using the 1-5-9 hidden layers. Only the semantic information task achieves the best accuracy with all 12 hidden layers. The WC probing task performs well with the first hidden layer, and the second-best accuracy is obtained with the 1-5-9 hidden layers.

Table 10. Classification accuracies for different sentence vectors (SL—SentLen, WC—WC, BS—BShift, TD—TreeDepth, SM—SOMO).

Model	**Name**	**Logistic Regression**					**MLP**				
		SL	**WC**	**BS**	**TD**	**SM**	**SL**	**WC**	**BS**	**TD**	**SM**
S-BERT	None	39.53	29.57	66.53	49.27	68.36	44.79	29.27	67.36	49.27	65.47
BERT	CLS	33.83	25	65.95	40.29	65.06	38.39	25.61	66.28	48.22	65.2
	Last layer	45.75	44.51	70.27	53.24	71.66	51.8	46.65	71.52	56.01	71.39
	Last four	50.57	52.44	72.77	51.92	73.59	56.79	55.79	72.94	56.54	72.9
	All	55.57	67.68	**73.61**	**54.69**	**74.83**	66.87	71.04	73.77	59.18	**73.18**
	First layer	**58.63**	**82.01**	67.94	52.44	70.01	66.96	**85.67**	67.78	58.39	70.56
	1-5-9	56.35	70.12	72.52	54.43	72.9	**68.8**	74.09	**73.94**	**59.84**	71.53
Glove	glove_6b	22.7	64.02	59.87	38.57	58.18	24.45	64.02	59.2	38.44	57.08
	glove_twitter	26.38	66.77	61.53	40.29	64.65	25.42	69.82	61.37	42.01	63.96
Word2Vec	word2vec_neg	25.24	62.2	61.62	36.2	67.4	26.47	70.43	62.7	40.16	66.16

6. Discussion

The experimental results related to the comparison of BERT sentence vectors with respect to GloVe and Word2Vec is given in Table 10. It can be observed that the BERT sentence vectors performed exceptionally well on all the probing tasks and performed better than GloVe and Word2Vec, despite these two representations having a rich vocabulary. Specifically, the GloVe model, despite being trained on a large corpus of Tweets, performed poorly. This overall performance observed for noisy texts is in agreement with the superior performance reported earlier [30–32] of contextual representations derived using BERT over non-contextual baselines on standard English sentences. Further, the sentence vectors derived from BERT's hidden layers achieved significantly better results over the state-of-the-art Sentence-BERT model. This underpins the importance of combining useful linguistic components to derive superiors sentence representations.

However, as we can see from Figure 3, the latter hidden layers of BERT performed poorly in capturing linguistic information compared to the shallow layers. We observe that the unstructured nature of the Tweets benefits more from the initial layers that capture shallow information than the last layers, which capture more complex hidden information. Since the results reported by authors of the BERT model [4], the top layers of the BERT model have been commonly used to derive sentence vectors for NLP tasks with both standard and noisy texts [7,31,36,46]. Nevertheless, our results confirm that the initial layers of the pre-trained BERT model are more efficient at comprehending noisy text. Further, the earlier layers of each region are observed to contribute more significantly toward encoding specific linguistic components.

Further, as we can see from Table 7, the experiments relating to the length of the sentence vector also revealed that the simpler predictive models perform better with large sentence vectors, while complex models are observed to prefer significantly smaller vectors. This underpins the fact that the complex models are better at identifying intricate patterns from compressed vectors that contain rich information. However, simpler models need higher dimensions of sentence vectors to achieve better results.

The methodology presented to systematically analyze the knowledge distribution within a multi-layer pre-trained language model, while generating sentence vectors, can capture various linguistic characteristics. This technique, being generic, can be directly applied to most multi-layer pre-trained language models to understand the linguistic properties captured by latent representations. The method leads to devising similar sentence embedding strategies to generate sentence embeddings from other Transformer-based models, such as Transformer-XL and GPT-3 models. The new probing datasets and the proposed framework can be used to study the ability of these models to comprehend natural language. Moreover, the noisy probing datasets generated in this study can lead to further research in NLU by providing additional datasets that cover the domain of noisy data.

It is also significant that future research should focus on the understanding of preprocessing Tweets to reduce the noise level of the linguistic knowledge distribution and the derived sentence representations. Moreover, the same probing dataset could be used to examine the relationship between the BERT's attention layers and the meaning-rich sentence embeddings. This could help to derive more meaning-rich sentence vectors.

7. Conclusions

The research work reported in this paper demonstrates that the general language understanding of pre-trained language models, such as BERT, can be effectively exploited to comprehend noisy texts. Further, the proposed methodology can effectively generate sentence vectors encoding different linguistic aspects, using latent representations of multilayer pre-trained language models. We observe that the shallow layers of the BERT model are better at capturing linguistic information of noisy and unstructured texts than the deeper layers for general English sentences [16]. Further, it can be noted that simple predictive models prefer large sentence vectors, while complex models are more successful with significantly smaller sentence vectors. It is worthwhile noting that the first layer or a combination of BERT layers from each region can be used to derive generalizable sentence vectors for noisy and unstructured texts.

We believe that our new noisy probing datasets can serve as benchmark datasets for future researchers to study the linguistic characteristics of unstructured and noisy texts. Currently, work is in progress on developing new and larger probing datasets for noisy texts, covering all 10 probing tasks.

Author Contributions: Conceptualization, B.K., M.C., A.S. and D.W.; data curation, B.K.; formal analysis, B.K.; funding acquisition, D.W.; investigation, B.K.; methodology, B.K., M.C. and A.S.; project administration, M.C.; resources, M.C.; software, B.K.; supervision, M.C., A.S. and D.W.; validation, B.K.; visualization, B.K.; writing—original draft, B.K.; writing—review and editing, M.C. and A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by Global Hosts Pty Ltd. trading as SportsHosts, a Melbourne-based company.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: We publish a new dataset that can serve as benchmark datasets for future researchers to study the linguistic characteristics of unstructured and noisy texts. These datasets are available in the public domain (https://bit.ly/3rK0g7P) and available on request.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
BERT	Bidirectional Encoder Representations from Transformers
NLP	Natural Language Processing
NLU	Natural Language Understanding
NMT	Neural Machine Translation
PCFG	Probabilistic Context-free Grammar

References

1. Kiros, R.; Zhu, Y.; Salakhutdinov, R.R.; Zemel, R.; Urtasun, R.; Torralba, A.; Fidler, S. Skip-Thought Vectors. In Proceedings of the 28th International Conference on Neural Information Processing Systems, NIPS'15, Montreal, QC, Canada, 7–12 December 2015; Volume 2, pp. 3294–3302.
2. Conneau, A.; Kiela, D.; Schwenk, H.; Barrault, L.; Bordes, A. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; pp. 670–680. [CrossRef]
3. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems, NIPS'13, Lake Tahoe, NV, USA, 5–10 December 2013; Volume 2, pp. 3111–3119.
4. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2019**, arXiv:1810.04805.
5. 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), Doha, Qatar, 25–29 October 2014; pp. 1532–1543. [CrossRef]
6. Coban, O.; Ozyer, G.T. Word2vec and Clustering based Twitter Sentiment Analysis. In Proceedings of the 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), Malatya, Turkey, 28–30 September 2018; pp. 1–5.
7. Kasthuriarachchy, B.; Chetty, M.; Karmakr, G.; Walls, D. Pre-Trained Language Models With Limited Data For Intent Classification. In Proceedings of the International Joint Conference on Neural Network (IJCNN), Glasgow, UK, 19–24 July 2020.
8. Grzeça, M.; Becker, K.; Galante, R. Drink2Vec: Improving the classification of alcohol-related tweets using distributional semantics and external contextual enrichment. *Inf. Process. Manag.* **2020**, *57*, 102369. [CrossRef]
9. Harb, J.G.; Ebeling, R.; Becker, K. A framework to analyze the emotional reactions to mass violent events on Twitter and influential factors. *Inf. Process. Manag.* **2020**, *57*, 102372. [CrossRef]
10. Ren, Z.; Shen, Q.; Diao, X.; Xu, H. A sentiment-aware deep learning approach for personality detection from text. *Inf. Process. Manag.* **2021**, *58*, 102532. [CrossRef]
11. González, J.Á.; Hurtado, L.F.; Pla, F. Transformer based contextualization of pre-trained word embeddings for irony detection in Twitter. *Inf. Process. Manag.* **2020**, *57*, 102262. [CrossRef]
12. Jacob, P.; Uitdenbogerd, A. Readability of Twitter Tweets for Second Language Learners. In Proceedings of the 17th Annual Workshop of the Australasian Language Technology Association, Sydney, Australia, 4–6 December 2019; pp. 19–27.
13. Boot, A.B.; Tjong Kim Sang, E.; Dijkstra, K.; Zwaan, R.A. How character limit affects language usage in tweets. *Palgrave Commun.* **2019**, *5*, 76. [CrossRef]
14. Bengio, Y.; Courville, A.; Vincent, P. Representation Learning: A Review and New Perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1798–1828. [CrossRef] [PubMed]
15. Goldberg, Y. Assessing BERT's Syntactic Abilities. *arXiv* **2019**, arXiv:1901.05287.
16. Jawahar, G.; Sagot, B.; Seddah, D. What does BERT learn about the structure of language? In Proceedings of the ACL 2019—57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019.
17. Kumar, A.; Makhija, P.; Gupta, A. Noisy Text Data: Achilles' Heel of BERT. In Proceedings of the 2020 EMNLP Workshop W-NUT: The Sixth Workshop on Noisy User-Generated Text, Online, 19 November 2020; pp. 16–21. [CrossRef]
18. Qiu, X.; Sun, T.; Xu, Y.; Shao, Y.; Dai, N.; Huang, X. Pre-trained models for natural language processing: A survey. *Sci. China Technol. Sci.* **2020**, *63*, 1872–1897. [CrossRef]
19. Conneau, A.; Kruszewski, G.; Lample, G.; Barrault, L.; Baroni, M. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia, 15–20 July 2018; pp. 2126–2136.
20. Peters, M.; Neumann, M.; Zettlemoyer, L.; Yih, W.T. Dissecting Contextual Word Embeddings: Architecture and Representation. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 1499–1509.
21. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching Word Vectors with Subword Information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146. [CrossRef]
22. Peters, M.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep Contextualized Word Representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, LA, USA, 1–6 June 2018; Volume 1, pp. 2227–2237.
23. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.U.; Polosukhin, I. Attention is All you Need. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17), Long Beach, California, USA, 4–9 December 2017; pp. 6000–6010.
24. Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J.G.; Le, Q.V.; Salakhutdinov, R. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, 28 July–2 August 2019; pp. 2978–2988. [CrossRef]
25. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. In Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS'20), Virtual, 6–12 December 2020; pp. 1877–1901.

26. Clark, K.; Khandelwal, U.; Levy, O.; Manning, C.D. What Does BERT Look at? An Analysis of BERT's Attention. In Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, Florence, Italy, 1 August 2019; pp. 276–286. [CrossRef]
27. Wang, A.; Pruksachatkun, Y.; Nangia, N.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; Bowman, S. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. In Proceedings of the 33rd International Conference on Neural Information Processing Systems (NIPS'19), Vancouver, BC, Canada, 8–14 December 2019; pp. 3261–3275.
28. Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; Bowman, S. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, Brussels, Belgium, 1 November 2018; pp. 353–355. [CrossRef]
29. Zhu, Y.; Kiros, R.; Zemel, R.; Salakhutdinov, R.; Urtasun, R.; Torralba, A.; Fidler, S. Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 19–27.
30. Liu, N.F.; Gardner, M.; Belinkov, Y.; Peters, M.E.; Smith, N.A. Linguistic Knowledge and Transferability of Contextual Representations. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, pp. 1073–1094. [CrossRef]
31. Tenney, I.; Xia, P.; Chen, B.; Wang, A.; Poliak, A.; McCoy, R.T.; Kim, N.; Bowman, S.R.; Das, D.; Pavlick, E. What do you learn from context? Probing for sentence structure in contextualized word representations. *arXiv* **2019**, arXiv:1905.06316.
32. Hewitt, J.; Manning, C.D. A Structural Probe for Finding Syntax in Word Representations. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, pp. 4129–4138. [CrossRef]
33. Wang, L.; Gao, C.; Wei, J.; Ma, W.; Liu, R.; Vosoughi, S. An Empirical Survey of Unsupervised Text Representation Methods on Twitter Data. In Proceedings of the Sixth Workshop on Noisy User-Generated Text (W-NUT 2020), Online, 19 November 2020; pp. 209–214. [CrossRef]
34. Shi, X.; Padhi, I.; Knight, K. Does String-Based Neural MT Learn Source Syntax? In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 1526–1534. [CrossRef]
35. Adi, Y.; Kermany, E.; Belinkov, Y.; Lavi, O.; Goldberg, Y. Fine-grained Analysis of Sentence Embeddings Using Auxiliary Prediction Tasks. In Proceedings of the ICLR Conference Track, Toulon, France, 24–26 April 2017.
36. Khatri, A.; P, P. Sarcasm Detection in Tweets with BERT and GloVe Embeddings. In Proceedings of the Second Workshop on Figurative Language Processing, Online, 9 July 2020; pp. 56–60. [CrossRef]
37. Reimers, N.; Gurevych, I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 3982–3992.
38. Klein, D.; Manning, C.D. Accurate Unlexicalized Parsing. In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, Sapporo, Japan, 7–12 July 2003; pp. 423–430. [CrossRef]
39. Kong, L.; Schneider, N.; Swayamdipta, S.; Bhatia, A.; Dyer, C.; Smith, N.A. A Dependency Parser for Tweets. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1001–1012. [CrossRef]
40. Foster, J.; Wagner, J.; van Genabith, J. Adapting a WSJ-Trained Parser to Grammatically Noisy Text. In Proceedings of the ACL-08: HLT, Short Papers, Columbus, OH, USA, 15–20 June 2008; pp. 221–224.
41. Liu, Y.; Zhu, Y.; Che, W.; Qin, B.; Schneider, N.; Smith, N.A. Parsing Tweets into Universal Dependencies. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, LA, USA, 1–6 June 2018; Volume 1, pp. 965–975.
42. Nivre, J.; de Marneffe, M.C.; Ginter, F.; Goldberg, Y.; Hajič, J.; Manning, C.D.; McDonald, R.; Petrov, S.; Pyysalo, S.; Silveira, N.; et al. Universal Dependencies v1: A Multilingual Treebank Collection. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), Portorož, Slovenia, 23–28 May 2016; pp. 1659–1666.
43. Bowman, S.R.; Angeli, G.; Potts, C.; Manning, C.D. A large annotated corpus for learning natural language inference. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 632–642.
44. Conneau, A.; Kiela, D. SentEval: An Evaluation Toolkit for Universal Sentence Representations. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, 7–12 May 2018.
45. Eger, S.; Rücklé, A.; Gurevych, I. Pitfalls in the Evaluation of Sentence Embeddings. In Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019), Florence, Italy, 2 August 2019; pp. 55–60. [CrossRef]
46. Gao, Z.; Feng, A.; Song, X.; Wu, X. Target-Dependent Sentiment Classification With BERT. *IEEE Access* **2019**, *7*, 154290–154299. [CrossRef]