Assessment of Word-Level Neural Language Models for Sentence Completion †

Heewoong Park ♦ and Jonghun Park *

Department of Industrial Engineering & Center for Superintelligence, Seoul National University, Seoul 08826, Korea; hee188@snu.ac.kr
* Correspondence: jonghun@snu.ac.kr; Tel.: +82-2-880-7174
† This paper is an extended version of our paper published in the proceedings of 2018 IEEE International Congress on Information Science and Technology (CiSt).

Received: 24 January 2020; Accepted: 12 February 2020; Published: 16 February 2020

Abstract: The task of sentence completion, which aims to infer the missing text of a given sentence, was carried out to assess the reading comprehension level of machines as well as humans. In this work, we conducted a comprehensive study of various approaches for the sentence completion based on neural language models, which have been advanced in recent years. First, we revisited the recurrent neural network language model (RNN LM), achieving highly competitive results with an appropriate network structure and hyper-parameters. This paper presents a bidirectional version of RNN LM, which surpassed the previous best results on Microsoft Research (MSR) Sentence Completion Challenge and the Scholastic Aptitude Test (SAT) sentence completion questions. In parallel with directly applying RNN LM to sentence completion, we also employed a supervised learning framework that fine-tunes a large pre-trained transformer-based LM with a few sentence-completion examples. By fine-tuning a pre-trained BERT model, this work established state-of-the-art results on the MSR and SAT sets. Furthermore, we performed similar experimentation on newly collected cloze-style questions in the Korean language. The experimental results reveal that simply applying the multilingual BERT models for the Korean dataset was not satisfactory, which leaves room for further research.

Keywords: BERT; bidirectional RNN; cloze test; Korean dataset; machine comprehension; neural language model; sentence completion

1. Introduction

In the research domain of machine reading comprehension (MRC), a cloze-style task whose objective is to restore the removed portion of text has been widely used to evaluate a machine’s level of understanding [1–3]. Sentence completion is a specific type of cloze-style task whose goal is to choose a correct word or phrase from the provided list of candidates to fill in the blank in a question sentence. Despite its simplicity, this class of questions can assess diverse abilities including linguistic proficiency, common knowledge, and logical reasoning at different levels.

To date, several publications have evaluated reading comprehension models through sentence completion tests. As an earlier comparative study, Zweig et al. [4] tested various methods based on different language models (LMs) and topic models against the Microsoft Research (MSR) Sentence Completion Challenge [5] and the Scholastic Aptitude Test (SAT) sentence completion questions, while these two datasets have become standard benchmark test sets for subsequent studies. Mikolov et al. [6] achieved an improved accuracy on the MSR by using the combination of skip-gram and recurrent neural network (RNN) LMs for sentence completion. Subsequently, deep neural models have received continuous attention for sentence completion, while Tang [7] and Woods [8] attained comparable...
results with classical non-neural feature based methods. In [9], the authors introduced a neural model named context2vec, which embeds a target word by considering the surrounding sentential context, demonstrating its usefulness in sentence completion in addition to word sense disambiguation and lexical substitution. Tran et al. [10] established the state-of-the-art results on the MSR set with Recurrent Memory Network (RMN), which stacked memory network blocks on RNN for language modeling.

Recently, Park et al. [11] revisited the word-level RNN LM based approach for sentence completion. Motivated by the empirical fact that the performance of the RNN LM highly depends on the number of nodes and optimization parameters [12,13], Park et al. demonstrated that their implementation of RNN LM surpassed the state-of-the-art models on the MSR set despite its simple architecture. Furthermore, they proposed a bidirectional version, which delivered additional performance gains by exploiting future context information. The authors also validated the RNN LMs against the SAT dataset, and they achieved higher accuracy than the other previously published results.

This work extends the study of Park et al. [11] with extensive experiments on various sentence completion methods based on neural LMs. To clarify which modification of the RNN LM mainly brings the performance gain, we added more experimental results for different choices of the network. Furthermore, this paper introduces and compares three criteria for selecting the answer based on a trained LM for sentence completion.

This study also includes a supervised learning approach that directly receives supervision from sentence completion questions. Specifically, we employed a task transfer framework that pre-trains an LM with a large text corpus and adapts it for sentence completion by modifying the network structure slightly and learning from a few questions. This framework has been emerging as a new paradigm in natural language understanding owing to its great success on many datasets [14–17]. In this work, we mainly follow the approach of Devlin et al. [16], while comparing pre-trained networks of BERT [16] and GPT2 [18], both of which are based on transformer architecture [19].

Another contribution of this paper is that we collected cloze-style questions written in Korean and evaluated the methods mentioned above with this dataset. There are few non-English datasets in MRC [20,21], which hinders the verification of the effectiveness of models in cases of other languages. The new dataset consists of 1823 multiple-choice questions from the Test of Proficiency in Korean (TOPIK). We conducted the performance analysis of our RNN LM and multilingual BERT models on this dataset.

In summary, our contributions are following:

- We demonstrate that, when properly trained, simple RNN LMs are highly competitive for the sentence completion. Our word RNNs achieved results beyond the previous best reported on the MSR and SAT datasets.
- We verify that the transfer learning approach that pre-trains a transformer-based LM from large data and fine-tunes the model for the target task is also viable for the sentence completion. Our experiments compared various pre-trained networks along with different settings for fine-tuning, showing that the performance varied significantly with different networks, and we were able to obtain state-of-the-art results for both datasets under certain configurations.
- The new cloze-style dataset written in Korean was collected from the government’s official examinations. Experimental results show that the models that were effective for the English datasets underperformed on the Korean dataset, leaving space for further investigation.
- The PyTorch implementation code (https://github.com/hevery/sentence-completion) for experimentation is made available to encourage subsequent studies on neural approaches in machine comprehension.

The remainder of this paper is organized as follows. In the next section, we begin with discussions on related work. In Section 3, we delineate word-level RNN LMs, formalize how to apply these LMs to sentence completion, and describe a fine-tuning approach that employs transformer-based models. Sections 4 and 5 show the results of experiments on the MSR and SAT sets, respectively. Then,
we present the new dataset and performance analysis on this set in Section 6. Finally, the paper is concluded in the last section.

2. Related Work

The machine comprehension of text or MRC, in which a machine is expected to answer a question in the form of natural language given a relevant source text, has become a major research topic in both academia and industry [22]. A natural language question that requests information such as WH-questions can be converted to the cloze-style counterpart in most cases by a simple algorithm that constructs a declarative sentence with the answer and masks it. Owing to its wide coverage and ease of generation, cloze-style datasets have been curated for different text domains and applications, including children’s book test [3], CNN and daily mail reading comprehension [1], summary cloze [23], and story cloze [24]. For this class of task, LM-based approaches have often served as solid baselines when the masked answers are from the daily vocabulary such as verbs and prepositions in children’s book test [3] or everyday life narratives [24,25]. Moreover, thanks to the gradual development of training techniques for neural LM, its performance has been improved [12,13]. Establishing a simple but effective baseline provides one of the key foundations for this kind of research, and this work attempts to make contributions along this line for the sentence completion task with state-of-the-art LMs.

Meanwhile, building a general-purpose feature extractor for text data has been a long-standing goal, as ImageNet-pretrained classifiers often play such a role for image data [26]. Among the notable breakthroughs were ELMo [27] and ULMFit [14] that learn an LSTM-based LM from large-scale plain text data and use the hidden representations as contextualized word embeddings of an input word sequence. Subsequently, a transformer-based LM equipped with a self-attention mechanism significantly reduced the LM perplexity and the error metrics of downstream tasks [15], while a bidirectional transformer-based model called BERT showed impressive results [16]. Numerous studies are still ongoing to further improve this transfer learning approach, including adding translation task objectives to provide high-quality supervision in pre-training [28], permutation-based language modeling to reduce the pretrain–finetune discrepancy [17], and robustly optimized pre-training [29]. However, the transfer learning approach has not been tested against the sentence completion datasets, and we aim to provide a variety of experimental results based on it.

Another interesting observation on transformer-based LMs is that multilingual models which were pre-trained from multiple monolingual corpora were able to generalize information across different languages [30]. Wu and Dredze [31] confirmed that a multilingual BERT model performed well uniformly across languages in document classification, named entity recognition, and part-of-speech tagging, when fine-tuned with a small amount of target language supervision for the downstream task. The more surprising finding of their work is that, even in a zero-shot cross-lingual setting, where the multilingual model is fine-tuned with task-specific data from one language and tested in another language, the multilingual BERT achieved comparable performance with the other models that require some cross-lingual supervision. A recent study further suggested that shared subword embeddings were not necessary, and monolingual models were also able to learn some universal linguistic abstractions [32]. Motivated by the above empirical evidence, we conjecture that multilingual BERT models might also perform well on the Korean cloze questions and accordingly considered them in our experiments.

3. Methods

3.1. Word-Level RNN LM

RNN based word-level LMs [33] that exploit the advantage of RNN in modeling sequential data have been widely applied in natural language processing [22]. The goal of word-level language modeling is to estimate the probability of the next word based on a previous text. Let $w_1, w_2, \ldots, w_n$ denote a word sequence. Then, a typical word-level RNN LM approximates the conditional probability
of the \( t \)th position, \( p(w_t|w_1, w_2, ..., w_{t-1}) \), by encoding the context words with RNN layers and decoding the output to the probability vector. In the following, we describe in turn a word-level RNN LM used in this work (referred to as word RNN henceforth), its bidirectional version, and another variant that adopts a training strategy called masked LM [16].

The unidirectional word RNN transforms each input word to a learnable embedding vector of size \( d_{emb} \). Then, recurrent layers take the word embedding and the previous hidden states where the past text has been encoded. We choose Long Short-Term Memory (LSTM) for the recurrent cell type and set the number of recurrent layers, \( l \) to 2, following the authors of [12,34]. A fully-connected layer reduces the dimension of the output hidden vector of the topmost recurrent layer from \( d_{hid} \) to \( d_{out} \). After the linearly projected vector is multiplied by an output embedding matrix, the softmax operation produces the likelihood of the next word, \( P_{LM}(w_t|w_1, w_2, ..., w_{t-1}) \). The training objective for the network is to minimize the negative log-likelihood of target words, also known as categorical cross-entropy loss. Dropout is applied between recurrent layers [34].

For the bidirectional version, we design it to infer a target word based on not only the previous words but also the subsequent words, which means the output of the network estimates \( p(w_t|w_1, w_2, ..., w_{t-1}, w_{t+1}, ..., w_n) \). Although this formulation cannot be applied for the language modeling in which a model is required to generate one word at a time from scratch, it fits in the sentence completion, which determines the missing word based on both sides. Some probabilistic interpretation of bidirectional modeling for sequences was suggested by Berglund et al. [35], who showed the effectiveness of bidirectional reconstruction on inferring omitted symbolic tokens such as characters in text and notes in mid-encrypted music. It is noteworthy that our bidirectional model fuses forward and backward hidden layers before the softmax operation rather than just ensembles the output distributions of forward and backward LMs as in [14,27].

To be specific, our bidirectional word RNN (Figure 1) has separate word embedding lookup tables for forward and backward directions, similar to context2vec [9]. Each word embedding is connected to either forward or backward directional LSTM layers. The network then aggregates the output of the bidirectional hidden layers into a final output vector. The aggregation consists of conducting a separate linear projection for each hidden layer vector and then adding the two projected vectors. This way of aggregation was slightly better than concatenation followed by a single linear projection in our preliminary experiments. The main difference between our bidirectional word RNN and context2vec is that our model computes the negative likelihood loss rather than an approximation of pointwise mutual information between the target word and the context during training.

Lastly, we present a variant based on masked LM training, which is a crucial strategy to pre-train bidirectional Transformers for language understanding [16]. We apply this to our bidirectional word RNN to see if the training strategy has a synergy with RNN-based architecture in place of the transformer-based one. Instead of aggregating the one-step-ahead hidden state from the forward layer and the one-step-behind hidden state from the backward layer, the masked LM version aggregates the current step hidden states from both directions. As an illustrative example, the bidirectional word RNN aggregates the forward hidden representation of ‘was’ and the backward hidden representation of ‘his’ for predicting ‘she’ in Figure 1, while the masked LM fuses the both direction hidden states at the position of ‘she’ to reconstruct it. For the masked LM training, 15% of the input tokens were replaced with noise data, and the network learned to reconstruct these tokens with the cross-entropy loss. Among the 15%, 80% of them were masked with [MASK] tokens, 10% of them were substituted with random words, and the others were kept unchanged, as suggested in [16].

### 3.2. LM-Based Scoring

This subsection describes three scoring strategies with respect to how to apply a trained word RNN to sentence completion. Suppose the \( t \)th position is blank, \( w_1, w_2, ..., w_{t-1}, w_{t+1}, ..., w_n \) are context words, and \( c^1, c^2, ..., c^m \) are candidate choices for the blank. Blank scoring strategy selects the choice word that maximizes the conditional likelihood, \( P_{LM}(c|w_1, w_2, ..., w_{t-1}) \) for a typical unidirectional LM or
\( P_{\text{biLM}}(c|w_1, w_2, ..., w_{t-1}, w_{t+1}, ..., w_n) \) for our bidirectional model, where \( c \in \{ c^1, c^2, ..., c^m \} \). This strategy is equivalent to picking the choice that minimizes the cross-entropy loss on the blank. In the case that a choice text spans more than a word, the strategy computes the score by summing the loss over the blank span and multiplying by \(-1\).

In the unidirectional case, however, the blank scoring strategy does not reflect any contextual information from future words following the blank. To compensate for this defect, we adopt the full scoring strategy that minimizes the cross-entropy loss on the entire sentence instead of the blank. Let \( w'_1, w'_2, ..., w'_{n_l} \) be the word sequence obtained by replacing the blank with a choice word \( c^l \), or phrase composed of multiple words \( c^l_1, c^l_2, ..., c^l_{n_l} \) where \( n_l = n - 1 + r_l \). Then, the strategy computes the score as \( \sum_{i=2}^{n_l} \log P_{\text{LM}}(w'_i|w'_1, w'_2, ..., w'_{i-1}) \) for the unidirectional LM or \( \sum_{i=2}^{n_l} \log P_{\text{biLM}}(w'_i|w'_1, w'_2, ..., w'_{i-1}, w'_{i+1}, ..., w'_n) \) for the bidirectional one, where \( j = 1, 2, ..., m \). Even in the bidirectional case, we observed that the full scoring strategy performed better than the former, although the blank scoring strategy also considers the future context.

The third strategy, referred to as partial scoring strategy [36], computes the score as the likelihood of the subsequent text conditioned on the choice word and its preceding text. For a unidirectional LM, the score of \( c \) is defined as \( \sum_{t=1}^{n} \log P_{\text{LM}}(w_t|w_1, w_2, ..., w_{t-1}) \), where the \( t \)th position is blank. Since \( P_{\text{LM}}(w_t|w_1, w_2, ..., w_{t-1}) \) for \( i = 1, ..., t - 1 \) does not depend on \( c \), the strategy is equivalent to picking the choice that minimizes \( \sum_{i=1}^{n} \log P_{\text{LM}}(w_i|w_1, w_2, ..., w_{t-1}, c, ..., w_{t-1}) \), which equals the full score minus the blank score. From this formulation, the partial score of \( c \) for the bidirectional model can be naturally derived as \( \sum_{t=1}^{n} \log P_{\text{biLM}}(w_t|w_1, w_2, ..., w_{t-1}, c, w_{t+1}, ..., w_n) + \sum_{t=1}^{n} \log P_{\text{biLM}}(w_t|w_1, w_2, ..., w_{t-1}, c, w_{t+1}, ..., w_n) \). While Trinh and Le [36] reported that the partial scoring strategy outperformed the full scoring strategy for pronoun disambiguation problems, we observed that it did not hold for the sentence completion.

For networks based on masked LM training, we can apply the blank scoring strategy by defining the score of choice \( c \) as \( P_{\text{mask}}(c|w_1, w_2, ..., w_{t-1}, [\text{MASK}], w_{t+1}, ..., w_n) \), where \([\text{MASK}]\) is inserted in the

---

**Figure 1.** Bidirectional word RNN (modified from Figure 1 in [11]).
blank. If a choice text consists of multiple words, the corresponding number of [MASK] tokens are inserted. Since it is not straightforward to apply either the partial or full scoring strategy to masked LM based networks, we did not take them into account.

3.3. Fine-Tuning Pre-Trained LM for Sentence Completion

We introduce a supervised learning approach for the case in which a few sentence-completion examples are available for model training. To mitigate this data insufficiency, which often occurs in fully-supervised settings due to the difficulty in collecting high-quality MRC data, recently researchers have proposed a task transfer framework that pre-trains an LM with a large amount of plain text, slightly modifies the network structure, and then fine-tunes it with a small dataset of the target task. For multiple-choice classification, the authors of [15,16] introduced a transferring method that adds a new simple linear regression layer to a pre-trained LM at the first [15] or the last [16] time step of the topmost hidden layers; obtains the scalar output values of input sequences corresponding to choices; and performs the softmax operation to those values to produce a probability vector for answers. We employ it for multi-choice questions for the sentence completion in a straightforward manner.

For construction of model input, we fill the blank with a choice word as in the full scoring strategy and insert a special [CLS] token at the first time step at which the output vector of the topmost hidden layer is fed into the simple linear regression layer. In contrast to Radford et al. [15] and Devlin et al. [16], the delimiter tokens are not inserted between question and choice texts, since the text filled with the correct answer forms a syntactically and semantically correct sentence. For a similar reason, while BERT LM also takes as input segment identifiers, which indicate if the corresponding token belongs to the question or the choice text, we feed the same segment identifier for the entire sequence into the model. An experiment in the following section shows that this simple method was sufficient for the sentence completion.

4. MSR Sentence Completion

First, we considered the MSR Challenge dataset [5] to evaluate the aforementioned methods. The MSR sentence completion set includes 1040 questions whose source sentences are from five Sherlock Holmes novels. Human workers were engaged in constructing the question sentences with five candidate choices for each question to ensure that the questions require semantic knowledge as well as logical inference [5]. We trained the neural LMs with the official training corpus for the challenge to compare with previous work, and then further investigated how much the performance can be improved with a larger model that was pre-trained with external data.

4.1. Results with the Official Training Data

For experiments in this subsection, we used 522 novels from Project Gutenberg written in the nineteenth-century, which is the specified standard training corpus of the MSR Challenge. We preprocessed the data through sentence splitting, followed by word tokenization and lower-casing to feed word sequences into the word RNNs. Sentences consisting of more than eighty or less than ten words were filtered out. As a result, the vocabulary was composed of 64K words after converting any word with fewer than six occurrences in the corpus to [UNK] token.

To be specific about the implementation, $d_{emb}$, $d_{hid}$, and $d_{out}$ were set to 200, 600, and 400, respectively. We set the forward and backward embedding sizes of the bidirectional word RNN to be equal. We trained all networks for ten epochs with stochastic gradient descent while applying 10% dropout. Twenty sentences of the same length comprised a mini-batch. Gradients were clipped at the maximum norm of 5.0 and normalized by the mini-batch size [34]. The learning rate was started at 0.5 and decreased by half at the beginning of each epoch after the fifth epoch. We determined these settings by inspecting the performance on the first 520 questions of the MSR set, referred to as development set, following prior work [9,37]. Accordingly, we refer to the remaining MSR questions as test set.
First, we investigated the performance of different language modeling and scoring strategies (Table 1). We optimized the networks repeatedly with five different random seeds and considered the average accuracy of those networks as a model accuracy. As expected, the full scoring was far more beneficial than the blank scoring in the unidirectional cases. Likewise, the accuracy of the bidirectional LM increased when using the full scoring compared to the blank scoring. In contrast to Trinh and Le [36], the partial scoring strategy was inferior to the full scoring strategy. This can be explained by the fact that a special word related to the answer appears in the following text of the blank in all questions of Levesque et al. [38], whereas the presence of a special word is not ensured in the MSR questions. In addition, the masked LM was inferior to the others, which differs from Devlin et al. [16]. It would seem that the masked LM requires more training data and prefers the transformer architecture rather than RNN layers. The performance gaps are visualized in Figure 2, where the error bars signify the minimum and maximum accuracies obtained by five networks with different random seeds.

Table 1. Performance comparison of word RNNs for different scoring strategies on the MSR set. We trained each model five times with different random seeds and report the average accuracy with the standard deviation in parentheses.

| LM Formulation    | Accuracy [%]      |
|-------------------|-------------------|
|                   | Blank | Full | Partial |
| Unidirectional LM | 50.5 (0.4) | 69.4 (0.8) | 56.4 (0.9) |
| Bidirectional LM  | 69.8 (0.5) | 72.3 (1.1) | 63.7 (1.4) |
| Masked LM         | N/A   | 58.2 (1.6) | N/A |

Figure 2. Performance comparison for different LM formulation and scoring strategies: (a) the full scoring was applied except for the masked LM; and (b) the bidirectional model was used. The x-axis represents the number of epochs and the y-axis represents the validation accuracy for the MSR dataset.

To justify the choice of the network size, we plot the validation results for the different numbers of nodes in Figure 3. The unidirectional word RNN was evaluated on the MSR validation set with different $d_{emb}$, $d_{hid}$, and $d_{out}$ values. Figure 3 verifies that the choice of $d_{hid}$ greatly affected the performance, while smaller embedding sizes than $d_{hid}$ led to better results possibly due to a regularization effect. We further experimented with more numbers of nodes, but the accuracy increase was negligible.
Table 2 compares the accuracy results of previous work with ours. An ensemble of differently initialized RNNs having the same dimensions was defined by selecting the option that minimizes the average loss obtained from the RNNs. All of our results in Table 2 were obtained by using the full scoring strategy. To present the approximate network sizes, we specify the dimensions in the first column as \((d_{emb} - d_{hid} - d_{out})\).

Table 2. Accuracy results (%) for the MSR Challenge dataset. The accuracies in italic were from their corresponding publications. We present the accuracy results of the development and test sets using a slash if they were measured separately. We report the average accuracy with the standard deviation if a model was optimized repeatedly.

| Model \((d_{emb} - d_{hid} - d_{out})\) | Accuracy |
|--------------------------------------|----------|
| RNNLMs [39]                         | 55.4     |
| Skip-gram [6]                       | 48.0     |
| Skip-gram + RNNLMs [6]              | 59.2/58.7 |
| NPMI + Co-occ. Freq. + LSA + CBOW + CSKIP [7] | 48     |
| PMI using Unigrams + Bigrams + Trigrams [8] | 61.44   |
| Context2vec (300-600-600) [9]       | 66.2/64.0 |
| LSTM (256-256-256) [10]             | 56.0     |
| Unidirectional-RM (512-512-512) [10] | 69.2     |
| Bidirectional-RM (512-512-512) [10] | 67.0     |
| Unidirectional word RNN (200-600-400) | 69.1 (0.9)/69.6 (0.8) |
| Bidirectional word RNN (200-600-400) | 72.5 (1.4)/72.0 (2.0) |
| Unidirectional word RNN ensemble     | 72.0/71.5 |
| Bidirectional word RNN ensemble      | 74.1/74.6 |

Interestingly, even the performance of our unidirectional word RNN was on par with or slightly above the previous best accuracy results on the MSR dataset. As examined in Figure 3, most accuracy gains compared to the LSTM of Tran et al. [10] were attributed to the appropriate embedding and hidden sizes, and miscellaneous training settings contributed to additional gains. When compared to the RNN [10], which contains a memory block consisting of two lookup tables for additional word embeddings and calculates an attention distribution over context words at every timestep, our RNN is simple and computationally inexpensive. In addition, our bidirectional version achieved the best result among the individual models while the ensemble of those five networks further improved the performance by leveraging the stochastic nature of deep learning. As mentioned above, adding more nodes to the unidirectional word RNN (200-600-400) was not helpful. This implies that the benefit
of bidirectional modeling did not come from mere increase of the number of learnable weights but from incorporation of future context information. These results disagree with Tran et al. [10], whose bidirectional model compared unfavorably with the unidirectional model.

4.2. Results with External Data

While the official training corpus is highly similar and relevant to the question sentences in terms of linguistic styles and time periods of writing, the limited data may hinder the learning of a deep neural network. We applied various pre-trained models that learn from large external data for the MSR questions (Table 3). LM1B represents the best single model of Józefowicz et al. [12], who experimented variants of LSTM-based LMs on the 1B word benchmark [40], which consists of news text of about one billion words. We also include variants of BERT and GPT2 models which have shown compelling results on many language understanding tasks with fine-tuning on small data or zero-shot task transfer. We obtained the pre-trained weights of LM1B and the others from their publicly available repositories (https://github.com/tensorflow/models/tree/master/research/lm_1b, https://huggingface.co/transformers/pretrained_models.html). According to the results of the scoring strategies (see Table 1), we applied the full scoring except for the BERT family for which we deployed the blank scoring strategy since BERT adopts masked language modeling. We adhere to this policy for further experiments unless noted otherwise.

Table 3. Accuracy results (%) of pre-trained models without fine-tuning for the MSR questions. The best model trained with the official dataset is presented at the first row for comparison.

| Model                                      | Dev. | Test |
|--------------------------------------------|------|------|
| Bidirectional word RNN ensemble           | 74.1 | 74.6 |
| LM1B                                       | 70.2 | 67.7 |
| BERT-base-uncased                          | 55.38| 58.46|
| BERT-base-cased                            | 60.58| 60.19|
| BERT-base-multilingual-uncased            | 22.50| 20.58|
| BERT-base-multilingual-cased              | 41.54| 41.92|
| BERT-large-uncased                         | 54.23| 55.00|
| BERT-large-cased                           | 52.69| 49.23|
| BERT-large-uncased-wwm                     | 77.69| 77.12|
| BERT-large-cased-wwm                       | 75.19| 77.12|
| GPT2                                       | 46.73| 44.04|
| GPT2-medium                                | 54.62| 52.69|

Although LM1B yielded better accuracy than the models of previous publications, its accuracy could not reach those of our word RNNs, and even its size was substantially larger than ours. Among BERT models, the performance differences between the case-sensitive ones (indicated as ‘-cased’) and the corresponding case-insensitive ones (indicated as ‘-uncased’) were inconsistent across different training settings. The multilingual BERTs that were trained from Wikipedia dumps of about one hundred languages were inferior to the monolingual ones, without fine-tuning to the target language. The authors of BERT have warned that lower-casing non-Latin alphabets could result in somewhat unwanted outcomes (https://github.com/google-research/bert), which might be the reason for the poor accuracy of BERT-base-multilingual-uncased. Despite having three times more learnable parameters, BERT-large models performed worse than BERT-base models without whole-word-masking. The whole-word-masking technique masks all-at-once the group of tokenized wordpieces [41] corresponding to a word during masked LM training (see details in https://github.com/google-research/bert). With this modification, whole-word-masking BERTs (indicated as ‘-wwm’) outperformed the BERT-large models without whole-word-masking and even surpassed the best ensemble model that was trained from the official training data. On the other hand,
GPT2 models, designed to be used with minimal adaptation, were less effective than BERT-base for the MSR set.

Next, we applied the fine-tuning methods described in Section 3.3 to the transformer-based models for the MSR set. The last one hundred questions of the development set were assigned to the holdout set for tuning optimization parameters and model selection, while the others were used for gradient updates. Among various design variables for optimization, we presumed that the learning rate and the decision on which layers to either freeze or fine-tune were the most important ones throughout preliminary experiments. Thus, a grid search for them was conducted with BERT-large-uncased-wwm, which achieved the best performance in the above experiment. We constructed each batch to contain only a single question and ran five epochs of gradient updates while using the slanted triangular learning rate schedule of warm-up for the first epoch. Unless noted otherwise, the optimizer was kept the same as the implementation employed (https://github.com/huggingface/transformers) [42].

Figure 4 displays holdout accuracies for different learning rates and subsets of weights to freeze during fine-tuning. The results show that the maximum accuracy that can be obtained by an appropriate learning rate did not differ much across different choices of updatable layers unless only a few top layers were updated. It indicates that we can achieve a satisfactory accuracy by only tuning learning rates while spending less effort on determining which layers to freeze. This simplification is more practically convenient than gradual unfreezing [14], which increases the number of trainable weights as iterations progress. Consequently, we took the approach that updates all layers for the remaining experiments.

![Figure 4](image)

**Figure 4.** Holdout accuracies for different learning rates and subsets of weights to freeze during fine-tuning. Up to layer $l$ in the legend indicates the setting that fixes the weights of the first $l$ self-attention layers as well as the embeddings. Note that, since the experimented BERT contains 24 self-attention layers in total, the bottommost setting updates the regression layer only.

Next, we conducted an experiment to validate the proposed method for the construction of input sequences. Table 4 indicates that our simplified approach was comparable or superior to the others, and, accordingly, we used it for the following experiments.

Finally, we present the MSR test accuracies of the fine-tuned models in Table 5. To see the effect of fine-tuning, we also display side by side the corresponding results without fine-tuning, which were copied from Table 3. The learning rate was chosen from \( \{1 \times 10^{-4}, 5 \times 10^{-5}, 2 \times 10^{-5}, 1 \times 10^{-5}, 5 \times 10^{-6}, 2 \times 10^{-6}, 1 \times 10^{-6}\} \) for each pre-trained model by comparing the best holdout accuracy results of five individual runs of fine-tuning with different random seeds. As shown in Table 5, the accuracies increased after fine-tuning for most pre-trained models, among which BERT-large models without the whole-word-masking strategy gained the most substantial amount of increase.
Nonetheless, the whole-word-masking models were still at the top through improving the accuracy by a healthy margin compared to the models that learned from the official training data.

Table 4. Accuracy results (%) on the holdout set for different input construction methods. For each method, we performed five runs of fine-tuning with different random seeds, and the average accuracy is reported with the standard deviation in parentheses.

| Input Construction                          | Accuracy  |
|--------------------------------------------|-----------|
| Proposed method                            | 84.2 (2.3)|
| with [SEP]                                 | 84.2 (1.3)|
| with segment identifiers                   | 84.6 (2.9)|
| with [SEP] and segment identifiers         | 74.6 (1.8)|

Table 5. Accuracy results (%) of transformer-based models for the MSR test set with and without fine-tuning, which are presented in w/o FT and w/FT columns, respectively.

| Model                        | w/o FT | w/FT  |
|------------------------------|--------|-------|
| BERT-base-uncased            | 58.46  | 68.65 |
| BERT-base-cased              | 60.19  | 73.27 |
| BERT-base-multilingual-uncased | 20.58  | 21.92 |
| BERT-base-multilingual-cased | 41.92  | 47.88 |
| BERT-large-uncased           | 55.00  | 79.23 |
| BERT-large-cased             | 49.23  | 79.42 |
| BERT-large-uncased-wwm       | 77.12  | 86.15 |
| BERT-large-cased-wwm         | 77.12  | 85.77 |
| GPT2                         | 44.04  | 38.27 |
| GPT2-medium                  | 52.69  | 59.23 |

5. SAT Sentence Completion

To further verify our findings, we gathered 152 sentence completion questions from the eight SAT practice exams provided by College Board (https://www.collegeboard.org) between 2003 and 2014. The reading part of the collected practice set includes sentence completion questions, each of which has one or two blanks in a sentence and provides five candidate choices. The collected SAT set contains that of Tang [7] but is covered by that of Woods [8].

Since any standard training corpus is not specified for the SAT set, previous researchers used different training corpora such as GloWbe [43], English Gigaword (LDC2009T13), and Wikipedia dumps. We used 1B word benchmark [40] to train the word RNNs, whose \( d_{emb} \), \( d_{hid} \), and \( d_{out} \) were set to 500, 2000, and 500, respectively. The learning rate was initialized to 1.0 and then multiplied by 0.8 after every epoch. We applied importance sampling to the output embedding matrix with sample size of 8192 [12] to deal with the large vocabulary size, which was 409K even after lower-casing. Other training details were kept identical to those for the MSR set, described in Section 4.1.

Table 6 shows the SAT evaluation results, including the previous best results. We also evaluated the word RNNs that had been trained on the MSR official training corpus (denoted by 19C novels). With more training data and larger network sizes, the SAT accuracies of the word RNNs increased considerably, exceeding the previously published results while being comparable to that of LM1B. Note that, besides its large size, LM1B utilized character level CNN layers to acquire intermediate representations for 793K word types [12], requiring more computing resources than our RNNs. In addition, bidirectional modeling was beneficial, as observed in Table 2.

Similar to the experimentation in Section 4.2, we also evaluated the transformer-based models with and without fine-tuning (Table 7). Since the SAT set is not large enough for splitting, we treated the entire MSR set as the training set while assigning the last hundred questions to the held-out validation set. With the learning rate that had been selected for each pre-trained model in the previous
section, we obtained the best model by re-running the fine-tuning five times on the new training set with different random seeds.

### Table 6. Accuracy results (%) for the SAT set. The accuracies in italic were copied from their corresponding publications. We report the average accuracy with the standard deviation if a model was optimized repeatedly.

| Model (\(d_{emb}\)-\(d_{hid}\)-\(d_{out}\)) | Training Corpus | Accuracy |
|---------------------------------------------|-----------------|---------|
| NPMI + Co-occ. Freq. + LSA + CBOW + CSKIP [7] | GloWbe          | 59      |
| PMI using Unigrams + Bigrams + Trigrams [8] | English Gigaword | 58.95  |
| Unidirectional word RNN (200-600-400)     | 19C novels      | 29.6 (1.5) |
| Bidirectional word RNN (200-600-400)      |                 | 33.3 (2.0) |
| Unidirectional word RNN (500-2000-500)    | 1B word benchmark | 66.5    |
| Bidirectional word RNN (500-2000-500)     | 1B word benchmark | 69.1    |
| LM1B (1024-8196-1024)                     |                 | 71.0    |

### Table 7. Accuracy results (%) of transformer-based models for the SAT set with and without fine-tuning, which are presented in w/o FT and w/FT columns, respectively.

| Model                        | w/o FT | w/FT  |
|------------------------------|--------|-------|
| BERT-base-uncased            | 30.92  | 42.11 |
| BERT-base-cased              | 30.92  | 52.63 |
| BERT-base-multilingual-uncased | 17.11  | 22.37 |
| BERT-base-multilingual-cased | 23.03  | 36.84 |
| BERT-large-uncased           | 25.66  | 73.03 |
| BERT-large-cased             | 27.63  | 73.03 |
| BERT-large-uncased-wwm       | 63.82  | 80.26 |
| BERT-large-cased-wwm         | 59.87  | 80.92 |
| GPT2                         | 38.16  | 32.24 |
| GPT2-medium                  | 53.29  | 53.29 |

Among the networks without fine-tuning, the whole-word-masking models were outstanding, while being outperformed by the word RNNs. Fine-tuning with the MSR questions was quite beneficial to the BERT models for the SAT questions, which indicated that the model learned a common solving strategy that worked across questions from different source texts. In addition, the BERT-large models outperformed the BERT-base models after fine-tuning, possibly due to the fact that the sentences and choice texts of the SAT questions were written using a difficult vocabulary compared to the MSR questions. As a result, the fine-tuned whole-word-masking models achieved state-of-the-art performance on the SAT questions.

### 6. TOPIK Cloze Questions

In this section, we attempt to confirm how effective the aforementioned methods are for a dataset other than in English. To do this, we newly collected 1823 cloze-style questions from Test of Proficiency in Korean (TOPIK) ([https://www.topik.go.kr](https://www.topik.go.kr)), provided by the website ([https://studytopik.go.kr](https://studytopik.go.kr)). TOPIK aims to assess the linguistic ability and guide the learning of the Korean language for non-native Korean speakers, and its results may be a prerequisite for entrance into universities or employment in companies and public institutions.

Tables 8 and 9 show the statistics of the collected TOPIK dataset. According to the level of difficulties, TOPIK test types have been divided into Levels 1–6 (lower is easier) until 2004, novice/intermediate/advanced from 2004 to 2014, and Level I/II (I is easier than II) since 2014. Most of the multiple-choice cloze-style questions appear in the easier test types except that 78 questions are from Level II. TOPIK is composed of sections including vocabulary, reading, and writing. Each
section contains a different number of cloze-style questions, while the lengths of questions in the writing section are larger than those of the others. Each question was tagged with ‘long’ or ‘short’ for its passage length and ‘single’ or ‘multi’ for the number of speakers. We counted for each tag type the number of words after splitting with whitespace and punctuation. While the passage lengths of the collected questions are not restricted to a single sentence, more than 70% are questions tagged as ‘short’, whose passages contain fewer than 10 words on average. Some instances from the TOPIK dataset are shown in Table 10.

| Table 8. Statistics of the cloze-style questions collected from previous TOPIK tests. |
| --- |
| Section | Type | The Number of Questions |
| | | Level 1 | Level 2 | Novice | Level I | Level II | Total |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Vocabulary | Short-single | 30 | 28 | 175 | 233 |
| | Short-multi | 78 | 91 | 303 | 472 |
| | Long-single | 6 | 14 | 48 | 68 |
| | Long-multi | 32 | 12 | 102 | 146 |
| | Subtotal | 146 | 145 | 628 | 919 |
| Reading | Short-single | 26 | 19 | 100 | 36 | 12 | 193 |
| | Short-multi | 4 | 4 | 14 |
| | Long-single | 19 | 13 | 65 | 49 | 66 | 212 |
| | Subtotal | 49 | 32 | 165 | 85 | 78 | 409 |
| Writing | Short-single | 25 | 22 | 90 | 47 |
| | Short-multi | 109 | 94 | 129 | 332 |
| | Long-single | 32 | 21 | 65 | 53 |
| | Long-multi | 41 | 22 | 63 | 63 |
| | Subtotal | 175 | 170 | 150 | 495 |
| Total | | 370 | 347 | 943 | 85 | 78 | 1823 |

| Table 9. Lengths of the cloze-style questions collected from previous TOPIK tests. |
| --- |
| Section | Type | Passage Length | Choice Length |
| | | Words | Chars | Words | Chars |
| --- | --- | --- | --- | --- | --- |
| Vocabulary | Short-single | 5.9 | 14.7 | 1.0 | 2.0 |
| | Short-multi | 11.2 | 25.0 | 1.3 | 3.5 |
| | Long-single | 42.4 | 107.7 | 1.7 | 4.2 |
| | Long-multi | 30.5 | 69.7 | 1.4 | 3.9 |
| | Subtotal | 15.2 | 35.6 | 1.3 | 3.3 |
| Reading | Short-single | 9.7 | 25.3 | 1.0 | 3.3 |
| | Short-multi | 10.2 | 26.0 | 1.2 | 5.0 |
| | Long-single | 52.4 | 139.1 | 2.3 | 6.6 |
| | Subtotal | 31.8 | 84.3 | 1.7 | 5.0 |
| Writing | Short-single | 14.0 | 35.9 | 3.7 | 10.1 |
| | Short-multi | 10.9 | 24.5 | 3.4 | 8.7 |
| | Long-single | 45.3 | 117.0 | 4.0 | 11.1 |
| | Long-multi | 41.4 | 93.1 | 3.4 | 8.5 |
| | Subtotal | 18.8 | 44.2 | 3.5 | 9.1 |
| Total | | 19.9 | 48.9 | 5.0 | 5.2 |
Table 10. Samples from the TOPIK dataset.

| Type       | Question                      | Translation                        |
|------------|-------------------------------|------------------------------------|
| Short-single | 온을 찾으려고 갑니다. 1. 은행 2. 운동장 3. 정찰서 4. 백화점 | I’m going to the _______ to find the money. |
|            |                               | 1. bank 2. playground 3. police office 4. department store |
| Short-multi | 여침에 많이 식사하셨어요? 우리는 ______ 시간이 다 다르니까 같이 못 먹어요. 1. 쉬 2. 끝나는 3. 일어나는 4. 네는 | Do you eat together in the morning? No, we can’t eat together because the times we ______ are different. 1. rest 2. finish 3. wake up 4. pay |
| Long-single | 친절은 다른 사람을 위한 따뜻한 마음과 행동입니다. 친절한 사람은 다른 사람에게 행동을 하지 않습니다. 그리고 남이 어려울 때 적극적으로 도와 줍니다. 친절한 말과 행동은 이 세상을 더 아름답게 만듭니다. 1. 좋은 2. 기쁜 3. 나쁜 4. 착한 | Kindness is a warm heart and action for others. Kind people do not act ______ others. They actively help when others are in trouble. Kind words and actions make this world more beautiful. 1. well with 2. happy with 3. bad to 4. nicely to |
| Long-multi | 경찰관: 어디에서 잃어버리시어요? 1. 아주머니: 백시 안에 넣고 내렸어요. 경찰관: 가방이 아주머니: 약간 흐려진 가방이에요. 경찰관: 뭐가 들어 있었습니까? 아주머니: 작할이 암 잘 찾아야 하세요. 경찰관: 잊어버렸습니다. 아주머니: 이저씨, 꼭 좀 부탁드립니다. 경찰관: 너무 걱정하지 마십시오. 함께 가서 기다리세요. 1. 어떻게 생겼어요? 2. 어떤 가게에서 샀어요? 3. 어떻게 만들었어요? 4. 어디에서 샀어요? | Officer: Where did you lose it? Ma’am: I left it in the cab. Officer: Your bag, ______ Ma’am: It’s a big black bag. Officer: What do you have in it? Ma’am: My wallet, I must find it. Officer: I’ll see what I can do. Ma’am: Please, officer. Officer: Don’t worry too much, go home and wait. 1. what does it look like? 2. what store did you buy it from? 3. how did you make it? 4. where did you buy it? |

We evaluated our word RNNs and the transformer-based models against the TOPIK dataset, which was split in half for development and test. We trained the word RNNs with the Sejong corpus (downloaded through https://ithub.korean.go.kr/user/total/database/corpusManager.do), applying the wordpiece tokenizer that was used for the BERT-base-multilingual-cased model. The network size and the training details were identical to those for the MSR experiment. The pre-trained BERT multilingual models were evaluated with or without fine-tuned in the same manner as the previous sections.

As shown in Table 11, with or without fine-tuning, the BERT models were inferior to our word RNNs. Moreover, the uncased BERT without fine-tuning was on par with a random classifier, whose expected accuracy is 25% for the TOPIK questions, each of which contains four choices. BERT-base-multilingual-cased was also not satisfactory on the Korean dataset as well as on the English datasets (see Sections 4.2 and 5), which requires further inspection and analysis to utilize multilingual models properly. Meanwhile, bidirectional modeling was not effective, which is likely due to the inadequate size of the training corpus having fewer than 20M words.

For further analysis, we visualize the mean accuracies of the models on the TOPIK development set for different difficulty levels and question types for each section in Figure 5. The fine-tuned models received supervision directly from the development set, thus being excluded from the analysis. Since we trained the word RNNs repeatedly with different random seeds, their mean accuracies are presented.

As can be seen, the vocabulary part was relatively easy, while the writing section was difficult for the models, whose accuracy correlated with the length of the choice text. The word RNNs yielded low accuracy when there were multiple speakers in the question compared to the single-speaker case since the training corpus does not contain many articles in the form of conversations. Meanwhile, the accuracies dropped as the difficulty levels increased (from Level 1 to 2, and from Level I to II) for the word RNNs. Lastly, the accuracy patterns over the subsets of the two BERT models were similar, which implies that the uncased model did not operate just randomly although its average performance was on par with a random classifier.
Table 11. Accuracy results (%) for the TOPIK set. We report the average accuracy with the standard deviation if a model was optimized repeatedly. For the fine-tuned (FT) models, the held-out accuracies are presented in Dev. column.

| Model                        | Dev.       | Test     |
|------------------------------|------------|----------|
| Unidirectional word RNN      | 43.4 (0.5) | 42.3 (0.6) |
| Bidirectional word RNN       | 43.3 (0.5) | 40.8 (1.1) |
| BERT-base-multilingual-uncased | 25.8       | 24.9     |
| BERT-base-multilingual-cased | 29.9       | 32.6     |
| FT BERT-base-multilingual-uncased | 30.0       | 27.3     |
| FT BERT-base-multilingual-cased | 38.0       | 33.7     |

Figure 5. Performance analysis on the TOPIK set. The mean accuracies for each difficulty level and question type are shown in (a,b), respectively. The black dotted line signifies the accuracy of a random classifier, which equals 0.25.
One possible direction to improve pre-trained multilingual LMs for MRC questions in a target language is to apply unsupervised domain adaptation [44]. Hundreds of questions would not be enough for a multilingual LM to fully adapt to both the target language and the downstream task. Accordingly, additional training of a pre-trained LM with unlabeled text data in a target language will be useful. Part-of-speech tags and dependency edges and labels can be augmented as supervision [45] to provide more explicit syntactic information in the target language. Inserting adapter modules [46] to the multilingual LM can be an efficient solution to capture language-specific properties with the modules while retaining universal linguistic abstractions learned by the pre-trained model. In addition, starting from a multilingual model optimized with translation-based objectives [28,47] and adapting it with available cross-lingual supervision for the target language and the downstream task is another interesting research direction. Since research on multilingual transfer learning is still active, we are currently investigating how to effectively apply those newly proposed methods to the TOPIK dataset.

7. Conclusions

In this study, we explored various methods based on neural LMs for sentence completion. With well-tuned network sizes and optimization parameters, we were able to enhance the performance of word RNNs, which reached beyond the previous state-of-the-art results on the MSR and SAT datasets. Furthermore, by fine-tuning the pre-trained transformer-based LMs with a few sentence-completion examples, we improved the accuracies significantly for both datasets. In addition, this paper presents the experiments on the newly collected cloze-style questions in the Korean language. The experimental results reveal that applying the multilingual BERT models for the Korean dataset led to unsatisfactory results, which necessitates further investigation. We hope our reproducible benchmarks help subsequent research to develop and validate diverse neural approaches in language understanding.

Author Contributions: Conceptualization, data curation, methodology, software, and writing—original draft, H.P.; and project administration, writing—review and editing, funding acquisition, methodology, and supervision, J.P. All authors have read and agreed to the published version of the manuscript.

Funding: Kakao Corp., Kakao Brain Corp., and National Research Foundation of Korea

Acknowledgments: This work was supported by Kakao and Kakao Brain corporations, and in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2019R1F1A1053366). The authors thank the administrative support from the Institute for Industrial Systems Innovation of Seoul National University.

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

References
1. Hermann, K.M.; Kocisky, T.; Grefenstette, E.; Espeholt, L.; Kay, W.; Suleyman, M.; Blunsom, P. Teaching Machines to Read and Comprehend. In Proceedings of the NIPS, Vancouver, BC, Canada, 7–12 December 2015; pp. 1693–1701.
2. Chen, D.; Bolton, J.; Manning, C.D. A Thorough Examination of the CNN/Daily Mail Reading Comprehension Task; ACL (1); The Association for Computer Linguistics: Stroudsburg, PA, USA, 2016.
3. Hill, F.; Bordes, A.; Chopra, S.; Weston, J. The Goldilocks Principle: Reading Children’s Books with Explicit Memory Representations. arXiv 2016, arXiv:1511.02301.
4. Zweig, G.; Platt, J.C.; Meek, C.; Burges, C.J.C.; Yessenalina, A.; Liu, Q. Computational Approaches to Sentence Completion; ACL (1); The Association for Computer Linguistics: Stroudsburg, PA, USA, 2012; pp. 601–610.
5. Zweig, G.; Burges, C.J.C. A Challenge Set for Advancing Language Modeling; WLM@NAACL-HLT; Association for Computational Linguistics: Stroudsburg, PA, USA, 2012; pp. 29–36.
6. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. arXiv 2013, arXiv:1301.3781.
7. Tang, E. Assessing the Effectiveness of Corpus-Based Methods in Solving SAT Sentence Completion Questions. JCP 2016, 11, 266–279. [CrossRef]
8. Woods, A. *Exploiting Linguistic Features for Sentence Completion*; ACL (2); The Association for Computer Linguistics: Stroudsburg, PA, USA, 2016.
9. Melamud, O.; Goldberger, J.; Dagan, I. Context2vec: Learning Generic Context Embedding with Bidirectional LSTM. In Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, Berlin, Germany, 11–12 August 2016; pp. 51–61.
10. Tran, K.M.; Bisazza, A.; Monz, C. *Recurrent Memory Networks for Language Modeling*; HLT-NAACL; The Association for Computational Linguistics: Stroudsburg, PA, USA, 2016; pp. 321–331.
11. Park, H.; Cho, S.; Park, J. Word RNN as a Baseline for Sentence Completion. In Proceedings of the 2018 IEEE 5th International Congress on Information Science and Technology (CiST), Marrakech, Morocco, 21–27 October 2018; pp. 183–187.
12. Józefowicz, R.; Vinyals, O.; Schuster, M.; Shazeer, N.; Wu, Y. Exploring the Limits of Language Modeling. *arXiv* 2016, arXiv:1602.02410.
13. Melis, G.; Dyer, C.; Blunsom, P. On the State of the Art in Evaluation in Neural Language Models. *arXiv* 2018, arXiv:1707.05589.
14. Howard, J.; Ruder, S. *Universal Language Model Fine-tuning for Text Classification*; ACL (1); Association for Computational Linguistics: Stroudsburg, PA, USA, 2018; pp. 328–339.
15. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. *Improving Language Understanding by Generative Pre-Training*; Technical Report; OpenAI: San Francisco, CA, USA 2018.
16. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*; NAACL-HLT (1); Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 4171–4186.
17. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.G.; Salakhutdinov, R.; Le, Q.V. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *arXiv* 2019, arXiv:1906.08237.
18. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* 2019, 1, 9.
19. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. In Proceedings of the Advances in neural information processing systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
20. Hardalov, M.; Koychev, I.; Nakov, P. Beyond English-only Reading Comprehension: Experiments in Zero-Shot Multilingual Transfer for Bulgarian. *arXiv* 2019, arXiv:1908.01519.
21. Cui, Y.; Liu, T.; Chen, Z.; Wang, S.; Hu, G. Consensus Attention-based Neural Networks for Chinese Reading Comprehension. *arXiv* 2016, arXiv:1607.02250.
22. Liu, S.; Zhang, X.; Zhang, S.; Wang, H.; Zhang, W. Neural Machine Reading Comprehension: Methods and Trends. *Appl. Sci.* 2019, 9, 3698. [CrossRef]
23. Deutsch, D.; Roth, D. *Summary Cloze: A New Task for Content Selection in Topic-Focused Summarization*; EMNLP/IJCNLP (1); Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 3718–3727.
24. Schwartz, R.; Sap, M.; Konstas, I.; Zilles, L.; Choi, Y.; Smith, N.A. *Story Cloze Task: UW NLP System*; LSDSem@EACL; Association for Computational Linguistics: Stroudsburg, PA, USA, 2017; pp. 52–55.
25. Xie, Q.; Lai, G.; Dai, Z.; Hovy, E.H. Large-scale Cloze Test Dataset Designed by Teachers. *arXiv* 2017, arXiv:1711.03225.
26. Huh, M.; Agrawal, P.; Efros, A.A. What makes ImageNet good for transfer learning? *arXiv* 2016, arXiv:1608.08614.
27. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. *Deep Contextualized Word Representations*; NAACL-HLT; Association for Computational Linguistics: Stroudsburg, PA, USA, 2018; pp. 2227–2237.
28. Conneau, A.; Lample, G. Cross-lingual Language Model Pretraining. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 7057–7067.
29. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* 2019, arXiv:1907.11692.
30. Pires, T.; Schlinger, E.; Garrette, D. *How Multilingual is Multilingual BERT?* ACL (1); Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 4996–5001.
31. Wu, S.; Dredze, M. *Beto, Bentz, Becas: The Surprising Cross-Lingual Effectiveness of BERT*; EMNLP/IJCNLP (1); Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 833–844.
32. Artetxe, M.; Ruder, S.; Yogatama, D. On the Cross-lingual Transferability of Monolingual Representations. *arXiv* 2019, arXiv:1910.11856.
33. Mikolov, T.; Karafiát, M.; Burget, L.; Černocký, J.; Khudanpur, S. Recurrent neural network based language model. In Proceedings of the Eleventh Annual Conference of the International Speech Communication Association, Chiba, Japan, 26–30 September 2010; pp. 1045–1048.
34. Zaremba, W.; Sutskever, I.; Vinyals, O. Recurrent Neural Network Regularization. *arXiv* 2014, arXiv:1409.2329.
35. Berglund, M.; Raiko, T.; Honkala, M.; Kärkkäinen, L.; Vetek, A.; Karhunen, J. Bidirectional Recurrent Neural Networks as Generative Models. In Proceedings of the NIPS, Vancouver, BC, Canada, 7–12 December 2015; pp. 856–864.
36. Trinh, T.H.; Le, Q.V. A Simple Method for Commonsense Reasoning. *arXiv* 2018, arXiv:1806.02847.
37. Mirowski, P.; Vlachos, A. Dependency Recurrent Neural Language Models for Sentence Completion; ACL (2); The Association for Computer Linguistics: Stroudsburg, PA, USA, 2015; pp. 511–517.
38. Levesque, H.J.; Davis, E.; Morgenstern, L. The Winograd Schema Challenge. In Proceedings of the Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning, Rome, Italy, 10–14 June 2012.
39. Mikolov, T. Statistical language models based on neural networks. In Proceedings of the Google, Mountain View, CA, USA, 2 April 2012.
40. Chelba, C.; Mikolov, T.; Schuster, M.; Ge, Q.; Brants, T.; Koehn, P.; Robinson, T. One billion word benchmark for measuring progress in statistical language modeling. *arXiv* 2014, arXiv:1312.3005.
41. Wu, Y.; Schuster, M.; Chen, Z.; Le, Q.V.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; et al. Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *arXiv* 2016, arXiv:1609.08144.
42. Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. HuggingFace’s Transformers: State-of-the-art Natural Language Processing. *arXiv* 2019, arXiv:1910.03771.
43. Davies, M.; Fuchs, R. Expanding horizons in the study of World Englishes with the 1.9 billion word Global Web-based English Corpus (GloWbE). *English World-Wide* 2015, 36, 1–28. [CrossRef]
44. Han, X.; Eisenstein, J. Unsupervised Domain Adaptation of Contextualized Embeddings for Sequence Labeling; EMNLP/IJCNLP (1); Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 4237–4247.
45. Kondratyuk, D.; Straka, M. 75 Languages, 1 Model: Parsing Universal Dependencies Universally; EMNLP/IJCNLP (1); Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 2779–2795.
46. Houlsby, N.; Giurgiu, A.; Jastrzkebski, S.; Morrone, B.; de Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; Gelly, S. Parameter-Efficient Transfer Learning for NLP. *Proc. Mach. Learn. Res.* 2019, 97, 2790–2799.
47. Conneau, A.; Khandelwal, K.; Goyal, N.; Chaudhary, V.; Wenzek, G.; Guzmán, F.; Grave, E.; Ott, M.; Zettlemoyer, L.; Stoyanov, V. Unsupervised Cross-lingual Representation Learning at Scale. *arXiv* 2019, arXiv:1911.02116.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).