ANNA: Enhanced Language Representation for Question Answering

Anonymous ACL submission

Abstract

Pre-trained language models have brought significant improvements in performance in a variety of natural language processing tasks. Most existing models performing state-of-the-art results have shown their approaches in the separate perspectives of data processing, pre-training tasks, neural network modeling, or fine-tuning. In this paper, we demonstrate how the approaches affect performance individually, and that the language model performs the best results on a specific question answering task when those approaches are jointly considered in pre-training models. In particular, we propose an extended pre-training task, and a new neighbor-aware mechanism that attends neighboring tokens more to capture the richness of context for pre-training language modeling. Our best model achieves new state-of-the-art results of 95.7% F1 and 90.6% EM on SQuAD 1.1 and also outperforms existing pre-trained language models such as RoBERTa, ALBERT, ELECTRA, and XLNet on the SQuAD 2.0 benchmark.

PASSAGE Architecturally, the school has a Catholic character. Atop the Main Building's gold dome is a golden statue of the Virgin Mary. Immediately in front of the Main Building and facing it, is a copper statue of Christ with arms upraised with the legend "Venite Ad Me Omnes". Next to the Main Building is the Basilica of the Sacred Heart. Immediately behind the basilica is the Grotto, a Marian place of prayer and reflection. It is a replica of the grotto at Lourdes, France where the Virgin Mary reputedly appeared to Saint Bernadette Soubirous in 1858. At the end of the main drive (and in a direct line that connects through 3 statues and the Gold Dome), is a simple, modern stone statue of Mary."

QUESTION: What sits on top of the Main Building at Notre Dame?

ANSWER: a golden statue of the Virgin Mary

Figure 1: Example of a passage with a pair of question and answer sampled from the SQuAD 1.1 dataset.

1 Introduction

Question answering (QA) is the task of answering given questions, which demands a high level of language understanding and machine reading comprehension abilities. As pre-trained language models based on Transformer (Vaswani et al., 2017) have brought a huge improvement in performance on a broad range of natural language processing (NLP) tasks including QA tasks, methodologies for QA tasks are widely used to develop applications such as dialog systems (Bansal et al., 2021) and chatbots (Hemant et al., 2022; Duggirala et al., 2021) in a variety of domains.

Pre-trained language models like BERT (Devlin et al., 2018) are designed to represent individual words for contextualization. However, recent extractive QA tasks such as Stanford Question Answering Dataset (SQuAD) benchmarks (Rajpurkar et al., 2016, 2018) involve reasoning relationships between spans of texts that include a group of two or more words in the evidence document (Lee et al., 2016). In the example, as shown in Figure 1, "a golden statue of the Virgin Mar", the correct answer for the question "What sits on top of the Main Building at Notre Dame?", is a group of words consisting of nouns and other words and is called as a noun phrase, which performs as a noun in a sentence. Since predicting a span of answer texts including a start and end positions may be challenging for self-supervised training rather than predicting an individual word, we introduce a novel pre-training approach that extends a standard masking scheme to wider spans of texts such as a noun-phrase rather than an entity level and prove that this approach is more effective for an extractive QA task by outperforming existing models.

In this paper, we present a new pre-training approach, **ANNA** (**A**pproach of **N**oun-phrase based language representation with **N**eighbor-aware **A**ttention), which is designed to better understand syntactic and contextual information based on comprehensive experimental evaluation of data processing, pre-training tasks, attention mechanisms. First, we extend the conventional pre-training tasks. Our models are trained to predict not only individual tokens but also an entire span of noun phrases during the pre-training procedure. This noun-phrase span masking scheme lets models learn contextualized representations in the whole span level, which benefits predicting answer texts for the specific extractive QA tasks. Second, we enhance the self-attention approach by incorporating a novel neighbor-aware mechanism in Transformer architecture (Vaswani et al., 2017). We find that more consideration of relationships between neighboring tokens by masking diagonality in attention matrix is helpful for contextualized representations. Additionally, we use a larger volume of corpora for pre-training language models and find that using a lot of of additional datasets does not guarantee the best performance.

We evaluate our proposed models on the SQuAD datasets which is a major extractive QA benchmarks for pre-trained language models. For SQuAD 1.1 task, ANNA achieves new state-of-the-art results of 90.6% Exact Match (EM) and 95.7% F1-score (F1). When evaluated on the SQuAD 2.0 development dataset, the results show that our proposed approaches obtain competitive performance outperforming self-supervised pre-training models such as BERT, ALBERT, RoBERTa, and XLNet models.

We summarize our main contributions as follows:

- We propose a new pre-trained language model, ANNA that is designed to address extractive QA tasks. ANNA is trained to predict the masked group of words that is an entire noun phrase, in order to better learn syntactic and contextual information by taking advantage of span-level representations.

- We introduce a novel transformer encoding mechanism stacking new neighbor-aware self-attention on an original self-attention in the transformer encoder block. The proposed method takes into account neighbor tokens more importantly than identical tokens during the computation of attention scores.

- ANNA establishes new state-of-the-art results on the SQuAD 1.1 leaderboard and outperforms existing pre-trained language models for the SQuAD 2.0 dataset.

2 Related works

Pre-trained contextualized word representations There have been many recent efforts on pre-training language representation models aiming for capturing linguistic and contextual information, and the models have brought a significant improvement of performance in a variety of NLP tasks. ELMo (Peters et al., 2018) is a deep contextualized word representation to learn complex characteristics of word use across linguistic contexts, and pre-trained models with these representations have shown noticeable improvements in many NLP challenges. BERT (Devlin et al., 2018) is a pre-trained language model with a deep bidirectional long short-term memory, which learns context in text using the masked language modeling (MLM) and the next sentence prediction (NSP) objectives for self-supervised pre-training. The latest language models (Liu et al., 2019; Lan et al., 2019; Yang et al., 2019b; Radford et al., 2018; Raffel et al., 2019a; Lewis et al., 2019) influenced by BERT mainly employ the transformer architecture (Vaswani et al., 2017) for pre-training but are trained with similar or extended to the pre-training objectives used in BERT implementation for enhancement of performance. There also exist many attempts to improve the capabilities of the standard transformer mechanism in contextualized word representations.

Extension of MLM Many recent studies have attempted to use different pre-training objectives by extending the MLM task in language modeling including BART (Lewis et al., 2019) and T5 (Raffel et al., 2019b). ELECTRA (Clark et al., 2020) introduces a new pre-training method of replaced token detection that replaces input tokens with alternative samples and detects whether the tokens are replaced or not. MASS (Song et al., 2019) is pre-trained on the sequence to sequence framework where fragments of input sentences are masked, and the masked fragment is predicted in its decoder part. XLNet (Yang et al., 2019b) adopts a span-based masking approach that predicts a masked subsequent span of tokens in a context of tokens autoregressively. SpanBERT (Joshi et al., 2020) and REALM (Guu et al., 2020) employ a span masking scheme that masks spans of tokens rather than random individual tokens, and the model is designed to learn span representations during pre-training. Similarly, LUKE (Yamada et al., 2020), ERNIE (Zhang et al., 2019), and KnowBERT (Peters et al., 2019) learn joint representations of words and entities by incorporating knowledge of entity embeddings.

Improvement of Attention Mechanism Since the standard transformer architecture has flexibility, many studies have shown the implementation of Transformer-based variants for improving further performance on language modeling and NLP tasks such as machine translation. Shaw et al. extends self-attention mechanism by incorporating embeddings of relative positions or distances between sequence elements, which is beneficial for performance improvement in machine translation tasks. Yang et al. introduces a context-aware self-attention approach that improves the self-attention with additional contextual information. Sukhbaatar et al. presents a novel attention method extending the self-attention layer with persistent vectors storing information which plays a similar role as the feed-forward layer. Fan et al. proposes a mask attention network that is a sequential layered structure incorporated a new dynamic mask attention layer with the self-attention and feed-forward networks.

3 Methodology

We introduce a novel transformer encoder architecture integrating a new neighbor-aware mechanism for pre-training a language model. Figure 2 demonstrates the architecture of ANNA model. ANNA extends the original transformer encoder blocks by including a neighbor-aware self-attention layer stacked on a multi-head self-attention layer.

3.1 Neighbor-aware Self-Attention

In this study, we propose a neighbor-aware attention mechanism. We assume that a single self-attention in Transformer encoder may be insufficient to learn context and the pre-trained models based on the transformer are hard to predict correct answers in downstream tasks due to linguistic noises brought in unrelated areas to a potential answer in the transformer encoder blocks. Here, we integrate a new neighbor-aware self-attention layer that is designed to remove influences of identical tokens by ignoring the diagonality in attention matrix when attention scores are computed. Instead, other tokens are more attended, so that the neighbor-aware mechanism enhances better understanding for relationships between tokens in inputs.

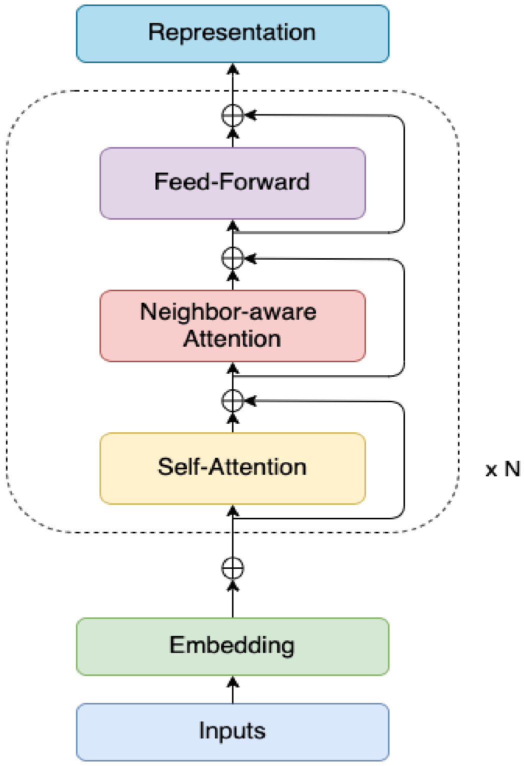

Figure 2: Architecture of ANNA.

As the Self-Attention layer shown in Figure 2 is adopted from the standard transformer architecture (Vaswani et al., 2017), we denote the self-attention as A_S that is calculated using query (Q), key (K) and value (V) projections as follows:

$$A_S(Q,K,V) = S_S(Q,K)V$$

$$S_S(Q,K) = \left[\frac{exp(Q_i K_j^T / \sqrt{d_k})}{\sum_k exp(Q_i K_k^T / \sqrt{d_k})} \right]$$

where Q, K and V represent HW_q, HW_k and HW_v, respectively. $H \in R^{L \times d}$ denoted as the input hidden vectors, L is the length of the input sequence, and d is the hidden size. $W_q, W_k, W_v \in R^{d \times d}$ are the projection matrices, and d_k is the query/key dimension. $A_S, A_N \in R^{L \times L}$ represents the attention matrices.

We define the Neighbor-aware Attention layer presented with A_N as follows:

$$A_N(Q,K,V) = S_N(Q,K)V$$

$$S_N(Q,K) = \frac{M(i,j) exp(Q_i K_j^T / \sqrt{d_k})}{\sum_k M(i,j) exp(Q_i K_k^T / \sqrt{d_k})}$$

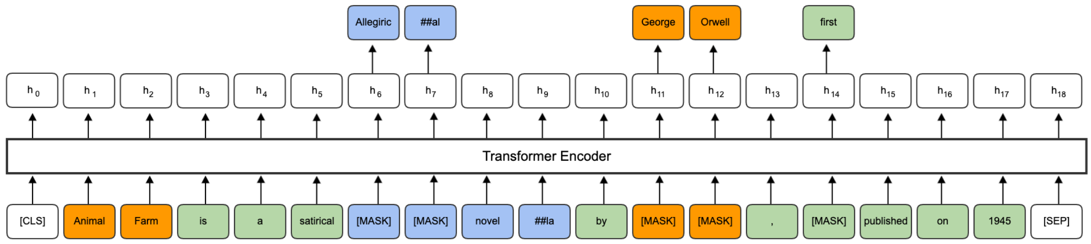

Figure 3: Example of the input sequence "*Animal Farm is a satirical allegorical novella by George Orwell, first published on 1945*" for pre-training ANNA. Different types of masking schemes are illustrated with such colors: masking a noun or noun phrase span (Orange), a whole word masking (Blue), and a wordpiece token masking (Green).

$$M(i,j) = \begin{cases} 0, & \text{if i = j} \\ 1, & \text{others} \end{cases}$$

where M denotes a mask that functions to omit capturing interactions of identical tokens. The interactions between each pair of input tokens x_i and x_j at positions *i* and *j* for $0 \leq i, j \leq L$ are calculated except for *i* = *j*.

3.2 Pre-training Task

We present a new pre-training task for training ANNA model. We follow the conventional MLM pre-training objective similar to BERT (Devlin et al., 2018). BERT is more sensible and effective to deeply represent context fusing the left and the right text with the MLM objective rather than unidirectional language models (Radford et al., 2018, 2019; Brown et al., 2020) or shallow Bi-LSTM models (Clark et al., 2018; Huang et al., 2015). In addition, a new masking scheme is applied for focusing on noun phrases in order to train our language model for better understanding syntactic and lexical information considering the specific downstream tasks. Here, we define three different masking schemes as illustrated in Figure 3. First, we use a span masking scheme that masks a group of texts in a span-level adopted by SpanBERT (Joshi et al., 2020). In this study, nouns or noun phrases identified by spaCy's parser (Honnibal and Montani, 2017) are randomly masked for span masking selection. Then we apply a whole word masking approach that masks all of the sub-tokens correspondings to a word at once, while we randomly mask tokens not included in the above two cases.

Following BERT, we randomly select 15% of the tokens in input sequences, and 80% of the selected tokens are replaced with the special token [MASK]. We keep 10% of the tokens in the rest of them unchanged, and the other 10% are replaced with randomly selected tokens. Our language model is also designed to train for the prediction of each token in the masked span by computing the cross-entropy loss function. However, the next sentence prediction (NSP) objective used in the BERT implementation is not used in this study, as RoBERTa (Liu et al., 2019) removes the NSP task due to performance decreases on downstream tasks.

3.3 Vocabulary and Tokenizer

In this study, we build a new vocabulary of 127,490 wordpieces that are extracted from the English Common Crawl corpus (Raffel et al., 2019a) and English Wikipedia dump datasets. The vocabulary consists of sub-words (30%) tokenized by the WordPiece algorithm (Wu et al., 2016), and 70% of the rest include noun-phrase words in their original form. We aim to prevent words from being out of vocabulary words and also keep noun phrases as the original forms so that our model is able to take many words in order to better learn human linguistic understanding during training.

In addition, we propose a new approach of word tokenization to suit our vocabulary used to pre-train ANNA model. This approach avoids separating words by special symbols since our vocabulary contains words including special characters by tokenizing noun-phrase words with white space only. Many studies use a subword-based word representation method for efficiency in vocabulary. A word is represented with several subword units tokenized by BERT tokenizer as exampled in Table 1. However, we do not follow this conventional tok-

Words	BERT tokens	ANNA tokens
Sant’Egidio	Sant , ’ , E , ##gi , ##dio	Sant’Egidio
COVID-19	CO , ##VI , ##D , - , ’19’	COVID-19
U.S.	U , . , S , .	U.S.
Ph.D.	Ph , . , D , .	Ph.D.
l’amour	l , ’ , am , ##our	l’amour
non-profit	non , - , profit	non-profit
X-Files	X , - , Files	X-Files
UTF-16	U , ##TF , - , 16	UTF-16
C++	C , + , +	C++

Table 1: Comparison of tokenization results between BERT and ANNA.

enization method (Wu et al., 2016), since we use a span masking scheme that masks an entire noun phrase randomly selected during a pre-training procedure. It is not suitable to train models as the length of masking tokens gets longer if subword units are used for the span masking scheme. We also aim to represent a whole-word token rather than subword units when attention scores are calculated. We implement an ANNA tokenizer in order to enhance a better understanding of contexts by not separating words as much as possible. Table 1 compares word tokenization results between BERT and ANNA tokenizers.

3.4 Pre-training Datasets

We use an English Wikipedia dataset like BERT (Devlin et al., 2018), and add publicly available English-language corpora such as a Colossal-Cleaned version of Common Crawl (C4) corpus (Raffel et al., 2019a), Books3 (Gao et al., 2020), and OpenWebText2 (OWT2) extended from WebText (Radford et al., 2019) and OpenWebTextCorpus (Gokaslan and Cohen) for pre-training our models. As shown in Table 2, the total size of data is about 900GB for the four corpora.

For pre-training language models with a large volume of corpora, it is crucial to generate high-quality data for inputs. We use heuristic pre-processing techniques to improve the data quality for the generation of input sequences as follows:

- Each document is split into sentences, and we filter the sentences including less than 10 words out due to their incompleteness. Also, documents with less than 100 words are ignored for input sequences.

- Text noises such as paragraph separators, special characters, URL addresses, and directory paths are heuristically filtered by regular expressions.

- For Books3 data, non-English documents are deleted by a language-detection module (Shuyo, 2010) which is utilized for the deletion of documents written in non-English words in the Common Crawl dataset.

- Since the maximum sequence length is 512 tokens, we split the pre-processed documents into multiple sentence chunks that do not exceed the predefined maximum length for the input of pre-training.

After the extensive data pre-processing procedure, we gain the size of 12GB, 580GB, 51GB, and 22GB for Wikipedia, C4, Books3, and OWT2, respectively. The pre-processed texts are tokenized into 410B word-piece tokens in total for pre-training our models.

In this study, we conduct an experiment in order to investigate whether the use of different sources of data for pre-training language models affects model performance on downstream tasks. We compare the performance of models pre-trained with different datasets in Table 3. We observe that C4 improves performance on the SQuAD 1.1 task when it is added to the Wikipedia dataset, but that models pre-trained over Books3 and OWT2 datasets are not beneficial for performance increases. We also find that the use of the larger volume of data including all of these four corpora is not helpful to improve performance. Thus we use both the C4 data and the Wikipedia corpus for pre-training ANNA models. Pre-training details for ANNA models can be found in Appendix A.

	Wikipedia	C4	*Books3*	*OWT2*
Size of text	16GB	730GB	100GB	62GB
Token counts for text	3.3B	160B	22B	13B
Size of pre-processed text	12GB	580GB	51GB	22GB
Token counts for pre-processed text	2.6B	126B	12B	5B

Table 2: Statistics of four corpora for pre-training including before and after the pre-processing procedure.

Corpora	EM	F1
Wikipedia	85.51	90.99
Wikipedia + C4	**85.90**	**91.02**
Wikipedia + Books3	85.40	90.79
Wikipedia + OWT2	84.79	90.27
ALL	85.14	90.22

Table 3: Comparison of model performance pre-trained with the different data sources. Models pre-trained with different pre-training corpora are evaluated on the SQuAD1.1 dataset. ALL includes the four datasets of Wikipedia, C4, Books3, and OWT2. Due to the limitation of computing resources, $\text{ANNA}_{\text{Base}}$ model is used for this experiment.

4 Experiments

In this section, we present the fine-tuning results of ANNA transferred to specific extractive question answering tasks.

We evaluate ANNA on SQuAD 1.1 and 2.0 tasks that are well-known machine reading comprehension benchmarks in the NLP area. The dataset of SQuAD 1.1 consists of around 100k pairs of a question and an answer along with Wikipedia passages where the answers are included. This task is to predict a correct span of an answer text for a given question from the corresponding Wikipedia passage (Rajpurkar et al., 2016). For SQuAD 2.0, the dataset is extended to the SQuAD 1.1 dataset by combining over 50,000 unanswerable questions, so that systems are required to predict answers to both answerable and unanswerable questions (Rajpurkar et al., 2018). We follow the fine-tuning procedure of BERT (Devlin et al., 2018), but the provided SQuAD training dataset only is used for fine-tuning, while BERT augments its training dataset with other QA datasets available in public.

Table 4 indicates the results of our best performing system compared with top results on the SQuAD 1.1 leaderboard. We also compare ours with BERT baselines. ANNA establishes a new state-of-the-art result on this task outperforming LUKE (Yamada et al., 2020) by EM 0.4 points and F1 0.3 points on the test dataset. LUKE is the latest best performing system in the leaderboard, and it is designed for contextualized representations of words and entities. As for a comparison with SpanBERT (Joshi et al., 2020) that masks contiguous sequences of token for span representations, ANNA also achieves better performance by both EM 0.8 points and F1 0.9 points.

ANNA is evaluated on SQuAD 2.0 development dataset, and the results are compared with the published pre-trained language models (Devlin et al., 2018; Liu et al., 2019; Lan et al., 2019; Yang et al., 2019b; Clark et al., 2020) in Table 5, which demonstrates that ANNA outperforms all of those language models and in particular, produces performance increases than ELECTRA by 0.4 points of EM and 0.2 points of F1.

5 Model Analysis

We conduct four additional experiments in terms of perspectives such as data processing, pre-training task, and attention mechanisms. We report a detailed analysis of how those approaches affect the performance of ANNA on a specific downstream task individually. In this study, $\text{ANNA}_{\text{Base}}$ model is used for these additional experiments due to the limitation of computing resources.

5.1 Effect of Data Processing

We describe several data pre-processing techniques we conduct to build a high-quality dataset for pre-training ANNA in Section 3.4. Here we demonstrate how the use of the data processing techniques affects the performance on the extractive question answering task. There exist documents with a variety of ranges of word length in the pre-training corpora. For a generation of an input sequence, documents containing less than 100 words are filtered out, while the others are split into multiple sentence chunks. Due to the maximum sequence length of 512, we limit the size of the chunks to not exceeding approximately 300 words. We observe that the data processing procedure making a suitable word

System	Dev EM	Dev F1	Test EM	Test F1
$\text{BERT}_{\text{Large}}$ (Devlin et al., 2018)	84.2	91.1	85.1	91.8
$\text{BERT}_{\text{Large}}$ (ensemble)	-	-	87.4	93.1
SpanBERT (Joshi et al., 2020)	-	-	88.8	94.6
$\text{XLNet}_{\text{Large}}$ (Yang et al., 2019b)	89.0	94.5	89.9	95.1
LUKE (Yamada et al., 2020)	89.8	95.0	90.2	95.4
$\text{ANNA}_{\text{Base}}$	87.0	92.8	-	-
$\text{ANNA}_{\text{Large}}$	**90.0**	**95.4**	**90.6**	**95.7**

Table 4: Performance of systems evaluated on the SQuAD 1.1 datasets.

System	SQuAD 2.0 Dev EM	SQuAD 2.0 Dev F1
$\text{BERT}_{\text{Large}}$ (Devlin et al., 2018)	79.0	81.8
$\text{ALBERT}_{\text{Large}}$ (Lan et al., 2019)	85.1	88.1
RoBERTa (Liu et al., 2019)	86.5	89.4
$\text{XLNet}_{\text{Large}}$ (Yang et al., 2019b)	87.9	90.6
$\text{ELECTRA}_{\text{Large}}$ (Clark et al., 2020)	88.0	90.6
$\text{ANNA}_{\text{Large}}$	**88.4**	**90.8**

Table 5: Performance of systems evaluated on the SQuAD 2.0 development dataset.

length for the max sequence length is helpful to improve performance slightly as shown in Table 6. However, the input sequences overlapped with 128 tokens at the back and front between successive sentence chunks rather hurt system performance.

5.2 Effect of Pre-training Mechanism

We investigate how different MLM objectives affect the performance of models on a specific downstream task. During a pre-training procedure, a model is trained with a deep bidirectional representation of input sequences. First, we concatenate part-of-speech (POS) tags to each word, then we apply a whole word masking approach to explore whether a masking method employing syntactic information is helpful to understand the context. We also mask tokens identified as named entities and noun phrases instead of masking single tokens randomly. In all of the experiments, we use the same percentage of 15% for the masking tasks. Table 7 compares results on the SQuAD 1.1 task for models using those MLM schemes. Comparing with the standard MLM approach that simply masks 15% of tokens, the pre-trained models using Entity and Noun-phrase MLM schemes improve performance, but the approach masking words including POS tags decreases performance than the standard MLM. Thus we use the Noun-phrase MLM approach to pre-train ANNA models for final results.

5.3 Effect of Neighbor-aware Self-Attention

We attempt to implement a new transformer encoder focusing on relatives, entities, or neighbors in input tokens in order to enhance capturing syntactic and contextual information. First, we extend the original self-attention based on the transformer in order to consider the pair-wise relationships between input tokens. The relation matrix of input tokens is simply added when attention scores are computed. For an entity-self-attention that focuses on named entities, we identify named entities in text and then compute additional attention scores to those entities for learning effective representations. We describe the mechanism of a neighbor-aware self-attention in detail in Section 3.1. We report that the neighbor-aware self-attention approach performs better than the original self-attention and other transformer modifications on the extractive question-answering task in Table 8. We consider that the neighbor-aware query mechanism is effective to capture relation information of neighboring tokens in an input sequence.

5.4 Effect of Layer-stacking Approach

We examine how approaches to stack sub-layers in a transformer encoder architecture impact performance. We compose a transformer encoder block by collaborating three sub-layers such as a self-

Data Processing	SQuAD1.1 Dev EM	SQuAD1.1 Dev F1
Wiki+C4 (Without sentence chunking)	85.9	91.0
Wiki+C4 (Sentence chunking with 128 token-overlap)	85.0	90.5
Wiki+C4 (Sentence chunking)	**86.3**	**91.2**

Table 6: Comparison of model performance pre-trained with the use of different data processing techniques.

Model	SQuAD1.1 Dev EM	SQuAD1.1 Dev F1
Standard MLM	83.7	89.1
w/POS	80.7	87.1
Entity	85.3	90.8
Noun phrase	**86.3**	**91.2**

Table 7: Results of different masking schemes during the pre-training task.

Model	SQuAD1.1 Dev EM	SQuAD1.1 Dev F1
Self-Att.	86.1	91.3
Relative-QK-Att.	86.0	91.1
Relative-QV-Att.	85.2	90.7
Entity-Self-Att.	85.7	90.9
Neighbor-Aware-Att.	**86.4**	**91.4**

Table 8: Comparison of model performance pre-trained with different transformer variants. Att is an abbreviation for Attention.

attention, a neighbor-aware self-attention, and a feed-forward network in different combinations. We evaluate the models using different combination methods of stacking layers and report the results on the SQuAD 1.1 dataset in Table 9.

We observe that a self-attention substituted with a neighbor-aware attention in an original transformer architecture decreases performance by F1 0.3 points. When a neighbor-aware attention is stacked between a self-attention and a feed-forward network, the model slightly performs better than the original transformer. The sequential layered structure of a self-attention, a neighbor-aware attention, and a feed-forward network achieve the best performance on the exact matching criteria, which demonstrates that our proposed approach has an effect on the extractive question answering task. We consider that attention scores computed in a self-attention layer are re-weighted to actually related tokens by ignoring identical tokens during the computation of attention scores in the neighbor-aware attention so that the neighbor-aware mechanism is helpful to capture relationships between input tokens.

Model	SQuAD1.1 Dev EM	SQuAD1.1 Dev F1
SA $\rightarrow$ FFN	85.6	90.9
NAA $\rightarrow$ FFN	85.5	90.6
SA $\rightarrow$ SA $\rightarrow$ FFN	85.5	91.0
NAA $\rightarrow$ NAA $\rightarrow$ FFN	86.1	91.5
NAA $\rightarrow$ SA $\rightarrow$ FFN	86.1	91.4
SA $\rightarrow$ NAA $\rightarrow$ FFN	**86.4**	**91.4**

Table 9: Performance of different stacking approaches of Self-attention (SA), Neighbor-aware-attention (NAA) and Feed-forward-network (FNN) layers in transformer encoder blocks.

6 Conclusion

In this paper, we present a novel pre-trained language representation model, ANNA which improves the original transformer encoder architecture by collaborating a neighbor-aware mechanism, and is pre-trained for contextualized representations of words and noun phrases in a span level. The experimental results show that ANNA achieves a new state-of-the-art on the specific extractive question answering task by outperforming published language model systems including BERT baselines, as well as the latest top system on the corresponding leaderboard. There are two main directions for future research: (1) validating the competitiveness of ANNA to a variety of NLP tasks; and (2) enhancing the robustness of ANNA in order to apply for real-world question answering tasks in business.

References

Aakash Bansal, Zachary Eberhart, Lingfei Wu, and Collin McMillan. 2021. A neural question answering system for basic questions about subroutines. In *2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pages 60–71. IEEE.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Kevin Clark, Minh-Thang Luong, Christopher D Manning, and Quoc V Le. 2018. Semi-supervised sequence modeling with cross-view training. *arXiv preprint arXiv:1809.08370*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Vishnu Dutt Duggirala, Rhys Sean Butler, and Farnoush Banaei Kashani. 2021. ita: A digital teaching assistant. In *CSEDU (2)*, pages 274–281.

Zhihao Fan, Yeyun Gong, Dayiheng Liu, Zhongyu Wei, Siyuan Wang, Jian Jiao, Nan Duan, Ruofei Zhang, and Xuanjing Huang. 2021. Mask attention networks: Rethinking and strengthen transformer. *arXiv preprint arXiv:2103.13597*.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Aaron Gokaslan and Vanya Cohen. Openwebtext corpus.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*.

P Hemant, Pramod Kumar, and CR Nirmala. 2022. Effect of loss functions on language models in question answering-based generative chat-bots. In *Machine Learning, Advances in Computing, Renewable Energy and Communication*, pages 271–279. Springer.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Kenton Lee, Shimi Salant, Tom Kwiatkowski, Ankur Parikh, Dipanjan Das, and Jonathan Berant. 2016. Learning recurrent span representations for extractive question answering. *arXiv preprint arXiv:1611.01436*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Matthew E Peters, Mark Neumann, Robert L Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. 2019. Knowledge enhanced contextual word representations. *arXiv preprint arXiv:1909.04164*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019a. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019b. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*.

Nakatani Shuyo. 2010. Language detection library for java. *Retrieved Jul*, 7:2016.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. *arXiv preprint arXiv:1905.02450*.

Sainbayar Sukhbaatar, Edouard Grave, Guillaume Lample, Herve Jegou, and Armand Joulin. 2019. Augmenting self-attention with persistent memory. *arXiv preprint arXiv:1907.01470*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. Luke: deep contextualized entity representations with entity-aware self-attention. *arXiv preprint arXiv:2010.01057*.

Baosong Yang, Jian Li, Derek F Wong, Lidia S Chao, Xing Wang, and Zhaopeng Tu. 2019a. Context-aware self-attention networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 387–394.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019b. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. Ernie: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129*.

Appendix

A Pre-training Details

Table 10 summarizes hyperparameters that we use for pre-training our two models: $\text{ANNA}_{\text{Base}}$ (L=12, H=768, A=12, Total Parameters=160M) and $\text{ANNA}_{\text{Large}}$ (L=24, H=1024, A=16, Total Parameters=550M). We use the maximum sequence length of 512, the Adam optimization (Kingma and Ba, 2014) with learning rates of 2e-4 and 1e-4 is used for the large and base models, respectively. Our large model $\text{ANNA}_{\text{Large}}$ is trained on 256 TPU v3 for 1M steps with the batch size of 2048, and it takes about 10 days.

Hyper-parameter	$\text{ANNA}_{\text{Large}}$	$\text{ANNA}_{\text{Base}}$
Number of layers	24	12
Hidden size	1024	768
FFN inner hidden size	4096	3072
Attention heads	16	12
Attention head size	64	64
Dropout	0.1	0.1
Warmup steps	10k	10k
Learning rates	2e-4	1e-4
Batch size	2048	1024
Weight decay	0.01	0.01
Max steps	1M	1M
Learning rate decay	Linear	Linear
Adam ε	1e-6	1e-6
Adam β_1	0.9	0.9
Adam β_2	0.999	0.999
Number of TPU	266	64
Training time	10 days	5 days

Table 10: Hyperparameters for pre-training ANNA models.