Impact of Grammar on Language Model Comprehension

Kimia Ameri, Michael Hempel, *Member, IEEE,* Hamid Sharif, *Fellow, IEEE*
Department of Electrical and Computer Engineering
University of Nebraska-Lincoln, USA
{kameri2}@huskers.unl.edu, {mhempel, hsharif }@unl.edu
Juan Lopez Jr. *Senior Member, IEEE*, Kalyan Perumalla
Oak Ridge National Laboratory, Oak Ridge, TN
{lopezj, perumallaks }@ornl.gov

***Abstract*—In this paper, we introduce a new language model based on transformers with the addition of syntactical information into the embedding process. We show that our proposed Structurally Enriched Transformer (SET) language model outperforms baseline datasets on a number of downstream tasks from the GLUE benchmark. Our model improved CoLA classification by 17 points over the BERT-Base model. Machine Learning (ML) and Natural Language Processing (NLP) are playing an increasingly vital role in many different areas, including cybersecurity in IT and OT networking, with many associated research challenges. The performance of attention-based models has been demonstrated to be significantly better than that of traditional algorithms in several NLP tasks. Transformers are comprised of multi attention heads stacked on top of each others. A Transformer is capable of generating abstract representations of tokens input to an encoder based on their relationship to all tokens in a sequence. Despite the fact that such models can learn syntactic features based on examples alone, researchers have found that explicitly feeding this information to deep learning models can significantly boost their performance. A complex model like transformers may benefit from leveraging syntactic information such as part of speech (POS).**

***Index Terms*—Natural Language Processing, Transfer Learning, Transformers, BERT, Part of Speed, Grammar Enriched**

I. Introduction

Cybersecurity threats to both IT and OT deployments have rapidly risen in recent years. This threat to networked devices and their communications can have far-reaching consequences, including in critical infrastructure industries such as the energy sector. To aid in these cybersecurity efforts, our team has been developing a semi-supervised vetting system (CYVET) [1]. CYVET is focused on cybersecurity for networking and for networked devices. It leverages many different NLP techniques, including document classification, sequence extraction, claim identification and sentiment analysis [2]. NLP is an increasingly important tool in today's cybersecurity arsenal.

Within NLP, grammatical induction is a method of learning grammar for syntactic parsing [3]. In natural language processing, grammar induction has been used in many aspects, including semantic parsing [4] and natural language understanding [5]. There are many challenges associated with grammar induction, such as sparsity and ambiguity of the data [6]. A neural network and transformer-based model with over-parameterization and continuous representation learning has been shown to be a powerful tools for solving unsupervised problems such as syntactic analysis [7]–[9].

Several pre-trained language representation models, including BERT [10] and GPT [11], have been developed in recent years for various natural language processing (NLP) applications. These language models are trained on unlabeled text to learn general knowledge from large corpora such as BookCorpus and the English Wikipedia. While existing pre-trained language models can learn to recognize useful linguistic information from unlabeled text [12], world facts and factual knowledge are generally not captured very well [5], [13]. More importantly, in a natural language, grammar regulates sentence structures and can help to understand the language. Recent studies showed that phrase structure grammar could be understood by pretrained language models [14]–[16]. It also showed that adding parse and latent trees on top of a pretrained language model can be useful for understating structure and grammar of a sentence. These models were trained on a parser tree to be able to understand the latent tree.

In this paper we focus on the impact of Part of Speech (POS) tags into language understanding. We propose a model that feeds the syntactic feature of a word, represented as a POS tag, along with its token into the transformer architecture. For each word, its token and assigned POS tags are embedded together and passed to the transformers. Words and POS tags are combined to form the token representation. We evaluate our proposed model on a number of General Language Understanding Evaluation (GLUE) benchmarks. Most of cybersecurity networking problems are binary classification problems, and therefore in this paper we similarly only focus on datasets with binary class labels. We showed that adding POS features into embedding improves the classifier performance in several downstream tasks.

This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The publisher acknowledges the US government license to provide public access under the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

II. Related Works

Pre-trained language models (PLMs) refer to Natural Language Processing (NLP) models trained on a large generic domain corpus of text. Embeddings from Language Models (ELMo) [17], Universal Language Model with Fine-Tuning (ULMFiT) [18], the Generative Pre-Training (GPT) model [11], and Bidirectional Encoder Representations from Transformers (BERT) [10] are well-established PLMs. In general, PLMs can be divided into two main categories based on their main application [19]:

1) Word-embedding language models such as ELMo [17].
2) Multi-purpose language models such as ULMFiT [18], Google BERT [10] and OpenAI GPT [11]

The introduction of Transformer-based PLMs, especially BERT, resulted in significant improvements for many NLP downstream tasks [11], [20]. However, recent studies showed that these models acquire only a weak understanding of syntactic structure through their training [5], [13], [21], [22]. Researchers found that adding syntax information can improve performance measures of NLP models [12], [23], [24].

One type of syntax information used to improve the performance of PLMs is POS tags. Certain POS types tend to become the keyword more frequently in certain domains or implementations [25], [26]. Huang *et. al.* [27] showed that using POS tags can help to facilitate the development of sentiment-favorable representations.

In this paper, we investigate the effect of adding POS tags into the embedding layer to train a new language model we call SET, the Structurally Enriched Transformers language model.

The remainder of this paper is organized as follows. In section 2, we briefly review related works, including background information related to Transformers and BERT. In section 3, we describe the overall system framework and our proposed architecture, including our training dataset, feature engineering tokenizer and our Structurally Enriched Transformers (SET) language model details. We present and discuss our findings and results by comparing SET's results with other language models in section 4, and our analysis and discussions in section 5. Finally, in section 6 we present our conclusions and future work.

A. Transformers

Transformer-based models were originally developed for machine translation [28] and have since then found their way into numerous other NLP applications. The GPT and BERT language model architectures contain multiple transformer blocks. These transformers are stacked on top of each other, which helps the model to extract more informative features from inputs using an attention mechanism. In transformer-based models, all input words first need to be converted to a token t. Each token is then converted to an embedding vector Ex_t, where E is a trainable matrix with M rows, where M is the total number of tokens, containing both words and sub-words. These embedding vectors are fed into encoder blocks with self attention mechanism. Mathematically, the self-attention matrix formula is (1):

$$\text{Self-Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \quad (1)$$

where Q is the Query matrix, K is the Key matrix, and V is the Value matrix. The Key matrix's dimension is provided by parameter d_k. As a result of the self-attention mechanism, each matrix expresses a different representation of the same initial embedding [28]. The softmax score in the attention equation determines how much each word will be expressed at this position. Multiplying the softmax score with the Value matrix (V) produces the attention value [28].

B. BERT

The Bidirectional Encoder Representations from Transformers (BERT) language model is a well-known PLM that was trained on a large general corpus. BERT has two main steps: Pre-training and Fine-tuning. For pre-training, BERT uses two main techniques: Masked Language Modeling (MLM) and Next-Sentence Prediction (NSP) [10]. BERT is available in two versions:

1) BERT-Base: 12 encoder layers, 768 dimensions, 12 multi-head attentions, 110M parameters;
2) BERT-Large: 24 encoder layers, 1024 dimensions, 16 multi-head attentions, 340M parameters.

Both models were trained on a corpus comprised of the BookCorpus and the English Wikipedia, which together contain more than 3.5 billion words [10].

For MLM, BERT trained a deep bidirectional representation of a sentence by randomly masking an input token and then predicting it. NSP is utilized to characterize and learn sentence relationships, on the other hand. These two unsupervised tasks helped BERT in producing context-sensitive embeddings. These embeddings and the self attention mechanism used in transformers help BERT to better understand the language. The transformers output from the MLM and NSP pre-training of BERT can be modified to be adapted to all downstream tasks [29]. For fine-tuning, a softmax layer is added on top of the stacked encoders as a classifier, which allows the model to perform various downstream tasks such as classification and sentiment analysis [10]. Equation (2) denotes the classifying loss calculated in fine-tuning:

$$\log(\text{softmax}(CW^T)) \quad (2)$$

where C denotes the encoders aggregation of the first token and W is the final weight.

For this research, we follow the pre-training and fine-tuning strategies defined by BERT to build our language model.

III. Proposed Architecture

In this section, we introduce our SET language model and its implementation details. In this research, we add the POS structural feature to the BERT model, and pre-trained the model for masked language and next sentence prediction. The proposed model architecture consists of six encoder layers,

with six heads for multi-headed attention. We used the same parameters as described in BERT, the dropout rate of 0.1 and a batch size of 2048. We utilize the AdamW optimizer to train the model with $\beta 1 = 0.9$ and $\beta 2 = 0.998$, and employed a label-smoothing factor of 0.1. Our model has 6M total trainable parameters and was trained for 113,300 steps.

The model proposed in this study was trained on a machine with four NVIDIA RTX A6000 GPUs, with 48 GB of RAM per GPU. The whole training process took 16h:28m. We used Pytorch [30] and Tensorflow [31] packages to implement this model. Figure 1 shows an overview of our proposed model architecture. In the following subsections we explain its components in detail.

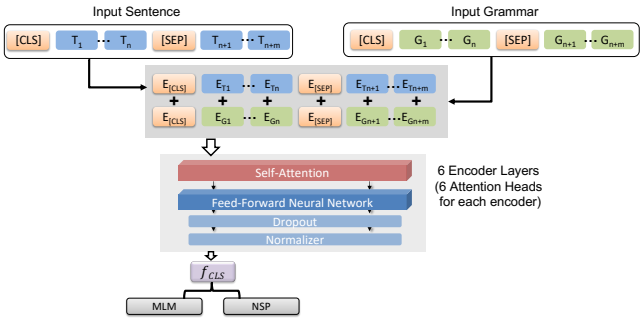

Fig. 1: Model architecture for the proposed SET Model

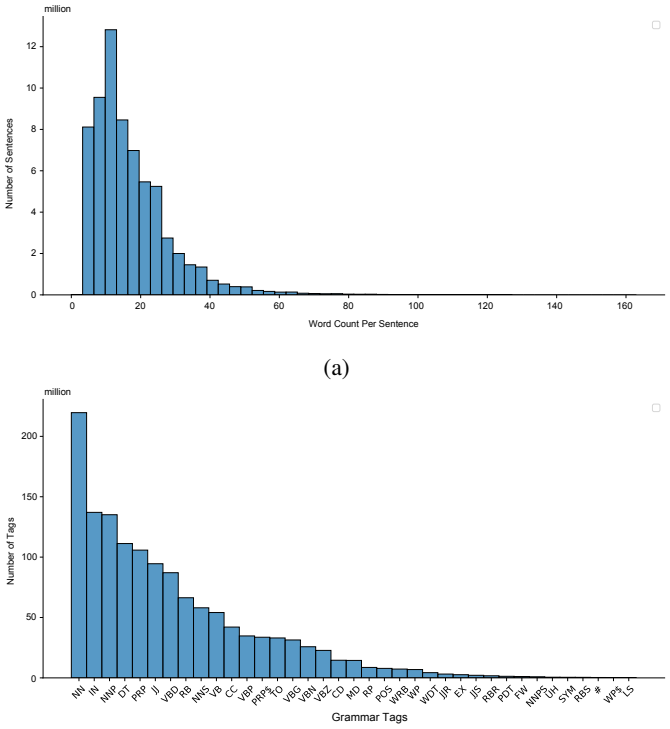

Fig. 2: Histogram of sentence word count for our selected corpus from BookCorpus and Wikipedia sentences (top) and the POS tag histogram for all sentences (bottom).

A. Training Dataset

The proposed model was trained on a subset of the BookCorpus and Wikipedia datasets. As part of our cleaning process, we removed duplicates and very long sentences that were more than 128 words in length. As the result, we trained our model on a corpus of 21,895 documents with 67,360,191 sentences and 1.1 billion words. Figure 2 demonstrates a histogram of the word count for each sentence in our selected corpus (on the left). We also demonstrate the frequency of each POS tag in our corpus in Figure 2 on the right. The mean sentence length in our corpus is 17.3 words, which shows that most of our training database's sentences are shorter than 128 words.

B. Feature Engineering Tokenizer

In this paper, we focused on evaluating the impact of grammar tags on language model understanding and performance. For this purpose, we added the POS tags from the NLTK python library [32] into the BERT tokenizer. A pre-trained NLTK 3.7 model was used to tag words in the source sequence with their POS tags, in order to aid the transformer in acquiring and utilizing syntactic information.

BERT has a vocabulary file for all its tokens, comprised of 30,522 tokens. At the beginning of its vocabulary file are 1,000 rows marked as $[unused]$. This allows researchers to introduce new tokens into those spaces. New POS tokens were assigned to rows 50-86 in the vocabulary file, each mapped to a specific POS tag. This will let the embedding matrix have the same size as the BERT model. Therefore, we can use the pre-trained embedding weights from BERT and benefit from transfer learning. Using these POS tags, along with the BERT tokenizer, helps us to build a feature engineered embedding encoder to train our new language model SET.

The first step in adding structural features, such as POS tags, to a BERT-Base model is mapping each token from the WordPiece tokenizer to the appropriate POS tag (Figure 4). The input sequence is first preprocessed by assigning POS tags to each source sequence. Following that, each sentence will be tokenized using the WordPiece model [33]. The next step will be to assign the proper POS to each token (sub-word) from WordPiece. For example, if the word $embedding$ were used as a noun in a sentence, the tokenizer would break it into sub-words em, bed, and ing, and each sub-word would be assigned the POS NN. If, on the other hand, the word $embedding$ were used as a verb in a different sentence, the tokenizer would arrange it into the same sub-words em, bed, and ing, but each sub-word would be assigned the POS VBG. Thus, it shows that the POS tag can convey additional, clarifying, information about these tokens. In order to avoid any confusion between POS tags such as TO and token to, we created a unique tokens for each POS tag. (see Figure 3). The embedding's grammar and sentence are then extracted from a trainable embedding matrix using a look-up table (Figure 1). As a result, syntactic and semantic elements are combined to form a compound representation.

POS Tag	Description	Token	POS Tag	Description	Token
CC	Coordinating conjunction	FENCC	RB	Adverb	FENRB
CD	Cardinal number	FENCD	RBR	Adverb, comparative	FENRBR
DT	Determiner	FENDT	RBS	Adverb, superlative	FENRBS
EX	Existential *there*	FENEX	SYM	Symbol	FENSYM
FW	Foreign word	FENFW	TO	*to*	FENTO
IN	Preposition or subordinating conjunction	FENIN	UH	Interjection	FENUH
JJ	Adjective	FENJJ	VB	Verb, base form	FENVB
JJR	Adjective, comparative	FENJJR	VBD	Verb, past tense	FENVBD
JJS	Adjective, superlative	FENJJS	VBG	Verb, gerund or present participle	FENVBG
NN	Noun, singular or mass	FENNN	VBN	Verb, past participle	FENVBN
NNS	Noun, plural	FENNNS	VBP	Verb, non-3rd person singular present	FENVBP
NNP	Proper noun, singular	FENNNP	VBZ	Verb, 3rd person singular present	FENVBZ
NNPS	Proper noun, plural	FENNNPS	WDT	Wh-determiner	FENWDT
PDT	Predeterminer	FENPDT	WP	Wh-pronoun	FENWP
POS	Possessive ending	FENPOS	WP$	Possessive wh-pronoun	FENWPS
PRP	Personal pronoun	FENPRP	WRB	Wh-adverb	FENWRB
PRP$	Possessive pronoun	FENPRPS	WRB$	Possessive Wh-adverb	FENWRBS
LS	List item marker	FENLS	MD	Model	FENMD

Fig. 3: POS tag definitions from NLTK and our specific assigned tokens

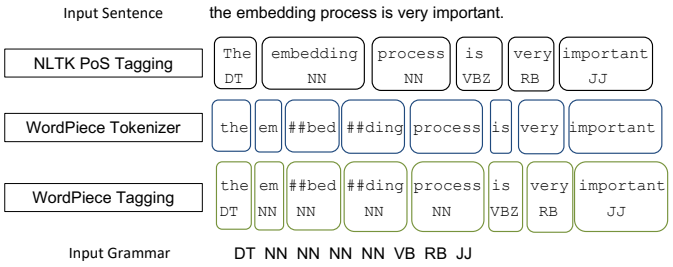

Fig. 4: Prepossessing steps to map NLTK tags to WordPiece tokens

C. The Structurally Enriched Transformers Language Model

In our proposed Structurally Enriched Transformers language model (SET), each token's input representation is calculated by summation of its token, grammar, segment, and position embeddings.

We use the same format as BERT's input, with a [CLS] token added in the beginning of each sequence and [SEP] token added to separate two sentences. These [CLS] tokens are fed into an output layer for classification and [SEP] is a separator token useful for downstream tasks such as question answering. These input vectors for each sentence are then fed into the encoders. Our model consists of six encoders stacked on top of each other. Similar to BERT, each encoder consists of multi-head, self-attention, and feed-forward neural network (FFN) sub-layers 1. The self-attention layer of the first encoder is initialized with the embedding matrix from the combined vectors of each word token and its POS tag in that sentence. Next, the Query, Key, and Value matrices are calculated for this embedding by the attention mechanism (Formula 1). Downstream tasks use the aggregate sequence representation corresponding to the [CLS] token from the final encoder, similar to BERT.

IV. Results

In this section, we present our language model fine-tuning results on benchmark datasets. The fine-tuning process uses the final weights from the pre-training stage and connects classifier layers with a dropout on top of the final encoder. We follow BERT's hyperparameter settings, including a dropout rate of 0.1, 768 neurons for the final classifier layers, a batch size of 32, and 3 training epochs. For each dataset, we choose the learning rate among the values of 4e-5, 3e-5 and 2e-5, which is within the hyperparameters defined by BERT [10]. The benchmark datasets utilized in this paper are part of the General Language Understanding Evaluation (GLUE) benchmark [34]. GLUE contains a variety of text classification tasks designed to assess general language comprehension abilities. In general, GLUE has nine different datasets, which are categorized into single sentence tasks, tasks related to similarity and paraphrases, as well as tasks related to natural language inference [34]. In this paper we only evaluate our proposed PLM on binary classification, and similarity and paraphrases datasets. Specifically, the list of datasets we used for this paper are:

- **QQP:** Question Pairs is a binary classification task with the goal of determining whether two questions on Quora are semantically equivalent [35]
- **SST-2:** Stanford Sentiment Treebank uses human annotations of movie reviews to classify binary single-sentence pairs [36].
- **CoLA:** is a binary single-sentence classification task designed to predict linguistically acceptable or unacceptable English sentences [37]
- **MRPC:** The Microsoft Research Paraphrase Corpus is a collection of sentence pairs automatically extracted from online news sources and human annotations on whether they are semantically equivalent [38].

The results obtained for our SET Model on the GLUE benchmark tests are shown in Table II. As can be seen from Table I, our proposed model outperforms all other models in CoLA and QQP.

Table II compares the detailed parameters for each model of the test set. We evaluated and reported the best value for each dataset among the 4e-5, 3e-5 and 2e-5 learning rates.

TABLE I: GLUE test results scored. The number below each task denotes the number of training examples.

	QQP (363k)	SST-2 (67k)	CoLA[a] (8.5k)	MRPC (5.7k)
OpenAI GPT [39]	70.3	91.3	45.4	80
BiLSTM+ELMo+Attn [34]	64.8	90.4	36	73.3
BERT Base [10]	71.2	90.5	52.1	85.8
BERT Large [10]	72.1	94.9	60.5	89.3
Syntax-infused BERT [23]	71.4	93.9	52.9	88.8
Proposed Model	**79.8**	81	**69.1**	71.1

[a]The evaluation metric for CoLA is Matthews Correlation

TABLE II: Hyper Parameters and GLUE Classification Results on Validation Set

	Learning Rate	Training Loss	Training[a] Accuracy	Validation Loss	Validation[a] Accuracy	Number of Epochs	Training Time
QQP	4e-5	0.38	0.82	0.42	0.798	3	0:28:03
SST-2	2e-5	0.198	0.936	0.558	0.813	3	0:21:26
CoLA	2e-5	0.609	0.704	0.616	0.691	3	0:02:13
MRPC	3e-5	0.528	0.743	0.594	0.711	3	0:01:13

[a]The evaluation metric for CoLA is Matthews Correlation

V. Analysis and Discussion

This paper presents our SET language model, which aims to capture the impact of grammar roles on language model understanding and performance. Our SET model outperforms all other models on two out of the four tasks we tested from the GLUE benchmark dataset. Our model achieved a 17 points improvement over BERT on the CoLA dataset. As we defined before, the CoLA task assesses a sentence's linguistic structure, which is clearly benefiting from the POS embeddings included in our model. The examples in Table III are from the CoLA database, showing how BERT mislabeled the data and how our model was able to correctly classify.

Furthermore, our SET model outperforms BERT on the QQP task, which is concerned with evaluating semantic relatedness. In Table IV, some examples of predictions made on the QQP dataset are presented.

VI. Conclusions and Future Work

In this paper, we introduced SET - a Structurally Enriched Transformers language model. This model is the result of added POS tags to the embedding matrix. We trained our language model on a dataset of 67.3 million sentences, utilizing a smaller architecture of only six encoders with 6M parameters, compared to other language models such as BERT and GPT. We find that our SET model can nevertheless outperform BERT Baseline and GPT on a number of GLUE downstream tasks. For example, we could note a significant improvement of 17 points compared to the BERT-Base model on the CoLA dataset, which evaluates the linguistic structure of a sentence. This work marks only a first step in our systematic evaluation of our hypothesis that a variety of NLP tasks could be improved by leveraging explicit prior syntactic information such as the POS. For our future work we will continue our evaluation of different grammar embedding techniques, different layer types for SET's encoders, as well as optimum training parameter set selection. We also plan on evaluating SET on other benchmark datasets related to semantic comprehension in NLP.

References

[1] Kimia Ameri, Michael Hempel, Hamid Sharif, Juan Lopez Jr, and Kalyan Perumalla. Smart semi-supervised accumulation of large repositories for industrial control systems device information. In *ICCWS 2021 16th International Conference on Cyber Warfare and Security*, pages 1–11. Academic Conferences Limited, 2021.

[2] Kimia Ameri, Michael Hempel, Hamid Sharif, Juan Lopez Jr, and Kalyan Perumalla. Design of a novel information system for semi-automated management of cybersecurity in industrial control systems. *ACM Transactions on Management Information Systems (TMIS)*, 2022.

[3] Bowen Li, Lili Mou, and Frank Keller. An imitation learning approach to unsupervised parsing. *arXiv preprint arXiv:1906.02276*, 2019.

[4] Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. Lexical generalization in ccg grammar induction for semantic parsing. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1512–1523, 2011.

[5] Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. Kepler: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194, 2021.

[6] Bowen Li, Jianpeng Cheng, Yang Liu, and Frank Keller. Dependency grammar induction with a neural variational transition-based parser. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6658–6665, 2019.

[7] Yo Joong Choe, Jiyeon Ham, Kyubyong Park, and Yeoil Yoon. A neural grammatical error correction system built on better pre-training and sequential transfer learning. *arXiv preprint arXiv:1907.01256*, 2019.

[8] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341*, 2019.

[9] Jian Li, Zhaopeng Tu, Baosong Yang, Michael R Lyu, and Tong Zhang. Multi-head attention with disagreement regularization. *arXiv preprint arXiv:1810.10183*, 2018.

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

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

[12] Nelson F Liu, Matt Gardner, Yonatan Belinkov, Matthew E Peters, and Noah A Smith. Linguistic knowledge and transferability of contextual representations. *arXiv preprint arXiv:1903.08855*, 2019.

[13] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866, 2020.

TABLE III: Examples of randomly selected sentences from the CoLA dataset that were mislabeled by BERT and correctly classified by our SET model.

Sentence	Label	BERT Class	Our Model Class
Mary intended John to go abroad.	0	1	0
The problem perceives easily.	0	1	0
Mary saw the boy walking toward the railroad station.	1	0	1

TABLE IV: Comparison of the assigned class label for some random sentences from the QQP dataset by BERT and our SET model.

Sentence 1	Sentence 2	Label	BERT Class	Our Model Class
Why did the US invade Iraq in 2003?	What led to the US invading Iraq in 2003?	1	0	1
Is global warming really increasing?	What are the causes of climate change?	0	1	0
How does cannabis affect cancer?	Can cannabis oil cure cancer?	1	0	1

[14] Taeuk Kim, Jihun Choi, Daniel Edmiston, and Sang-goo Lee. Are pre-trained language models aware of phrases? simple but strong baselines for grammar induction. *arXiv preprint arXiv:2002.00737*, 2020.

[15] Phu Mon Htut, Kyunghyun Cho, and Samuel R Bowman. Grammar induction with neural language models: An unusual replication. *arXiv preprint arXiv:1808.10000*, 2018.

[16] Bowen Li, Taeuk Kim, Reinald Kim Amplayo, and Frank Keller. Heads-up! unsupervised constituency parsing via self-attention heads. *arXiv preprint arXiv:2010.09517*, 2020.

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

[18] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

[19] Kimia Ameri, Michael Hempel, Hamid Sharif, Juan Lopez Jr, and Kalyan Perumalla. Cybert: Cybersecurity claim classification by fine-tuning the bert language model. *Journal of Cybersecurity and Privacy*, 1(4):615–637, 2021.

[20] Tao Shen, Xiubo Geng, Tao Qin, Daya Guo, Duyu Tang, Nan Duan, Guodong Long, and Daxin Jiang. Multi-task learning for conversational question answering over a large-scale knowledge base. *arXiv preprint arXiv:1910.05069*, 2019.

[21] Adhiguna Kuncoro, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom. Lstms can learn syntax-sensitive dependencies well, but modeling structure makes them better. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436, 2018.

[22] Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. What does bert learn about the structure of language? In *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[23] Dhanasekar Sundararaman, Vivek Subramanian, Guoyin Wang, Shijing Si, Dinghan Shen, Dong Wang, and Lawrence Carin. Syntax-infused transformer and bert models for machine translation and natural language understanding. *arXiv preprint arXiv:1911.06156*, 2019.

[24] Kelly W Zhang and Samuel R Bowman. Language modeling teaches you more syntax than translation does: Lessons learned through auxiliary task analysis. *arXiv preprint arXiv:1809.10040*, 2018.

[25] NM Dhanya, RK Balaji, and S Akash. Aixam-ai assisted online mcq generation platform using google t5 and sense2vec. In *2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS)*, pages 38–44. IEEE, 2022.

[26] Manish Agarwal and Prashanth Mannem. Automatic gap-fill question generation from text books. In *Proceedings of the sixth workshop on innovative use of NLP for building educational applications*, pages 56–64, 2011.

[27] Minlie Huang, Qiao Qian, and Xiaoyan Zhu. Encoding syntactic knowledge in neural networks for sentiment classification. *ACM Transactions on Information Systems (TOIS)*, 35(3):1–27, 2017.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[29] Kimia Ameri, Michael Hempel, Hamid Sharif, Juan Lopez Jr, and Kalyan Perumalla. An accuracy-maximization approach for claims classifiers in document content analytics for cybersecurity. *Journal of Cybersecurity and Privacy*, 2(2):418–443, 2022.

[30] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[31] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016.

[32] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

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

[34] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[35] Wei Wang, Ming Yan, and Chen Wu. Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering. *arXiv preprint arXiv:1811.11934*, 2018.

[36] Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. Learned in translation: Contextualized word vectors. *Advances in neural information processing systems*, 30, 2017.

[37] Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019.

[38] Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. The fifth pascal recognizing textual entailment challenge. In *TAC*, 2009.

[39] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding with unsupervised learning. 2018.