Expert Systems With Applications 165 (2021) 113898

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Hierarchical and lateral multiple timescales gated recurrent units with pre-trained encoder for long text classification

Dennis Singh Moirangthem, Minho Lee *

School of Electronics Engineering, IT-1, Kyungpook National University, 80 Daehakro, Bukgu, Daegu 41566, South Korea

ARTICLE INFO

Keywords:
Text classification
Multiple timescale
Temporal hierarchy
BERT
Pre-trained encoder

ABSTRACT

Text classification, using deep learning techniques, has become a research challenge in natural language processing. Most of the existing deep learning models for text classification face difficulties when the length of the input text increases. Most models work well on shorter text inputs, however, their performance degrades with the increase in the input length. In this work, we introduce a model for text classification that can alleviate this problem. We present the hierarchical and lateral multiple timescales gated recurrent units (HL-MTGRU), in combination with pre-trained encoders to address the long text classification problem. HL-MTGRU can represent multiple temporal scale dependencies for the discrimination task. By combining the slow and fast units of the HL-MTGRU, our model effectively classifies long multi-sentence texts into the desired classes. We also show that the HL-MTGRU structure helps the model to prevent degradation of performance on longer text inputs. We demonstrate that the proposed network with the help of the latest pre-trained encoders for feature extraction outperforms the conventional models on various long text classification benchmark datasets.

1. Introduction

Text classification is a category of Natural Language Processing (NLP) with real-world applications. The goal of this task is to assign labels to texts. It has a number of applications including topic labeling (Wang & Manning, 2012), sentiment classification (Maas et al., 2011; Pang et al., 2008), chat discrimination (Moirangthem et al., 2017), and spam detection (Sahami et al., 1998). Conventional methods (Wang & Manning, 2012) involve representing documents with sparse lexical features like *n*-grams, and then linear models or kernel methods are used on the representation for the task. An important intermediate step is text representation. More recent approaches used deep learning, such as convolutional neural networks (CNN) (Conneau et al., 2016; Johnson & Zhang, 2017; Kalchbrenner et al., 2014; Kim, 2014; Shen et al., 2018; Zhang et al., 2015), recurrent neural networks (RNN) based on long short-term memory (LSTM) (Hochreiter & Schmidhuber, 1997) in Liu et al. (2016), Seo et al. (2017) and Yogatama et al. (2017), and attention mechanisms (Lin et al., 2017; Yang et al., 2016) to learn text representations. The recently proposed Transformer (Vaswani et al., 2017), uses only self-attention mechanism for sequence tasks such as machine translation and achieved superior performance compared to its recurrent counterparts.

Language model pre-training for learning universal language representations with the help of huge amounts of unlabeled data has shown to be very effective. A recent survey by Gao et al. (2019) has demonstrated the effectiveness. Some of the most outstanding examples are embeddings from language models (Peters et al., 2018), generative pre-training (Radford et al., 2018) and bidirectional encoder representations from Transformers (BERT) (Devlin et al., 2018). There are also a number of variants introduced recently. These models are neural language models trained on a large amount of text data using unsupervised objectives. For example, BERT is a Transformer encoder, which is trained on unlabeled text for masked word prediction and next sentence prediction tasks. Unlike other models, BERT produce bidirectional encoder features from text sequences. In order to use such pre-trained models in natural language understanding (NLU) tasks, fine tuning of the models is required with additional layers using task-specific training data for each task. For example, BERT can be fine-tuned to address a range of NLU tasks (Devlin et al., 2018). However, the number of parameters in such models are large and the resources required to run increases. Recently, Lan et al. (2020) introduced a new pre-trained encoder model with parameter reduction techniques in order to alleviate scaling issues in such pre-trained models. The model is a lite version of BERT (ALBERT) and significantly reduce the number of parameters but retaining the performance and thereby improving parameter-efficiency. ALBERT has 18x fewer parameters and is about 1.7x faster than BERT. In our work, we adopt ALBERT for feature extraction due to its reduced number of parameters and enhanced performance.

* Corresponding author.
E-mail addresses: mdennissingh@gmail.com (D.S. Moirangthem), mholee@gmail.com (M. Lee).

https://doi.org/10.1016/j.eswa.2020.113898
Received 26 February 2020; Received in revised form 28 July 2020; Accepted 16 August 2020
Available online 21 August 2020

The recently introduced language model pre-trained Transformers have been widely used for sentence level text classification (Devlin et al., 2018; Lan et al., 2020; Radford et al., 2018) and have been very successful. However, recent studies reveal that the lack of recurrence in the Transformer models hinders its further improvement due to its limitations in handling longer sequences (Chen et al., 2018; Dehghani et al., 2019). Introducing recurrence in pre-trained Transformers can be one way to take advantage of the existing models in order to further enhance the performance. In this work, we address the long text classification problem, which is a multiple sentence text classification problem. This task requires handling of longer sequences efficiently. Moreover, the classification model should also be robust in handling diverse sequence lengths. For example, in the Yahoo Answers topic classification task, the input text length may vary from 130 to 4018 characters. These varying lengths of text should be properly handled by the model to classify the inputs to 10 different topics. Therefore, there is a need for the development of a model that can handle both long and short sequences efficiently in order to address long multi-sentence text classification problems.

In this work, we develop a deep network for long text classification with the help of a multiple timescales gated recurrent unit (MTGRU) (Kim et al., 2016; Moirangthem & Lee, 2017, 2020; Moirangthem et al., 2017) and a pre-trained Transformer encoder. We take advantage of the aforementioned recurrency and self-attention mechanisms while also recycling existing pre-trained models in a modularized manner, saving time and computational power while enhancing its performance. Although pre-trained Transformer based approaches to NLP tasks have shown enhanced performance, we argue that better representations can be obtained by incorporating a temporal hierarchy in the model architecture of the large pre-trained language model Transformers. The intuition underlying our model is that different parts of the model will focus on different lengths of the text with the goal to address better classification of a diverse length of input texts. The temporal hierarchy concept with MTGRU has also been proven to perform well in language modeling (Moirangthem & Lee, 2017; Moirangthem et al., 2017) and summarization (Kim et al., 2016) tasks. The MTGRU is known to handle long term dependency better with the help of varying timescales to represent multiple compositionalities of language. The temporal hierarchy approach has also been shown to eliminate the need for complex structures and normalization techniques (Chung et al., 2017; Cooijmans et al., 2017; Ha et al., 2017; Krueger & Memisevic, 2016), and thereby increasing the computational efficiency of the model. Moreover, pre-trained encoders have been proven to be great feature extractors, which are suitable for several NLU tasks including text classification problems. However, the available pre-trained encoders such as BERT and ALBERT are trained on a masked language model (MLM). In the MLM training objective, the output vectors are grounded to tokens instead of sentences, while in long text classification, we must encode and represent multi-sentence inputs. We solve this representation issue by introducing the proposed MTGRU layers on top of the pre-trained encoders.

We improve the MTGRU model by introducing a hierarchical and lateral multiple timescales structure. The conventional hierarchical MTGRU is most effective for handling long term dependencies in very long text inputs for applications such as summarization but performs comparable to vanilla GRU with shorter text inputs. Our proposed hierarchical and lateral multiple timescales gated recurrent unit (HL-MTGRU) is significantly different from the conventional MTGRU structure. HL-MTGRU follows a lateral (branch or root) architecture and a hierarchical structure where the slow and fast units are directly connected to the inputs and the final outputs of the units are combined to form the final representation. The fast units are also connected to the slow units in a hierarchical fashion. The hierarchical and lateral connections in the HL-MTGRU will enable encoding of rich features that have different temporal dependencies from the input sentences in order to help classify the information correctly. This structure enables all the layers with different timescales to capture relevant features directly from the inputs keeping the advantages of hierarchical multi-layer structures. This feature enables efficient handling of both short and long sequence data. Since the data consist of inputs of different lengths, HL-MTGRU proves to be more suitable for this task.

Our major contributions are as follows:

- We introduce the HL-MTGRU network, with the help of pre-trained encoders, to build rich features from input texts to classify texts of diverse lengths.
- The HL-MTGRU architecture enables our model to perform well on longer text sequences with the help of the slow layer as well as maintain comparable performance on shorter sequences.
- In order to demonstrate that the proposed model outperforms the existing models, we report the performance on various long text classification benchmark datasets.
- The results of our experiments demonstrate that the proposed model achieves state-of-the-art performance on the benchmark datasets.

2. Related work

Deep neural network models have demonstrated huge success in many NLP tasks, including learning distributed word, sentence, and document representations (Le & Mikolov, 2014; Mikolov et al., 2013), neural machine translation (Cho et al., 2014), sentiment classification (Kim, 2014), etc. Neural network models require little external domain knowledge in learning distributed sentence representations and such model can produce satisfactory results in related tasks like document classification, text categorization, and so on.

Neural network models are constructed upon either the input word sequences or the transformed syntactic parse tree in sentence representation learning tasks. CNN and RNN have been the popular ones for a long period of time. The capability of CNN to capture local correlations as well as extracting higher-level correlations through pooling empowers it to model sentences naturally from consecutive context windows. Kim (2014) proposed a CNN architecture with multiple filters and multiple channels for text classification. However, recently, language model pre-training has emerged to be superior in representation learning.

Language Model Pre-training Pre-trained word embeddings can offer significant improvements over embeddings learned from scratch (Mikolov et al., 2013; Pennington et al., 2014). It has become an important component of modern NLP systems. Sentence embeddings (Kiros et al., 2015; Logeswaran & Lee, 2018) and paragraph embeddings (Le & Mikolov, 2014) are also currently used as features in downstream tasks. In order to achieve the state-of-the-art performance on several major NLP benchmarks, Peters et al. (2018) concatenated embeddings derived from language model as additional features. Transfer learning with a large amount of supervised data can also achieve good performance in addition to pre-training with unsupervised data. It has been used in natural language inference (Conneau et al., 2017) and machine translation (McCann et al., 2017) tasks.

More recently, pre-trained language models (Devlin et al., 2018; Gao et al., 2019; Radford et al., 2018) have emerged as a key technology for achieving impressive gains in a wide variety of natural language tasks. Such methods pre-train language models on a large network with a large amount of unlabeled data and are fine-tuned on downstream tasks. These models extend the idea of word embeddings by learning contextual representations from large-scale corpora using a language modeling objective. Models such as OpenAI GPT (Radford et al., 2018), BERT (Devlin et al., 2018), ALBERT (Lan et al., 2020), have achieved significant performance enhancements with the pre-training strategy. BERT is a new language representation model which is trained with a masked language model (MLM) objective to predict words which are randomly masked or replaced and a "next sentence prediction" task

on a corpus of 3300M words. It is a bi-directional encoder, unlike previous bidirectional language models limited to a combination of two unidirectional language models (i.e., left-to-right and right-to-left). BERT was the first fine-tuning based representation model that achieves state-of-the-art results for a range of NLP tasks, demonstrating the enormous potential of the fine-tuning method. However, the recently introduced enhancement known as ALBERT reduced the number of parameters while retaining the performance of BERT. The reduced parameter solves the scaling problem to some extent and it opens up new opportunities for us to experiment with added deep layers to enhance the performance on specific tasks such as long text classification, text summarization, and other NLU tasks.

Recurrence in Transformer Recent studies have shown the importance of recurrency in complementing self-attention networks, namely Transformer (Chen et al., 2018; Hao et al., 2019). While Transformers are able to abstract high-level information from inputs beyond their sequential nature and use parallel computation to their advantage, recurrent approaches enable the capture of important input features such as hierarchical structure representation (Tran et al., 2018). Several models have been proposed to either introduce recurrence inside Transformers (Hao et al., 2019) or introduce self-attention mechanism in RNNs (Wang et al., 2019). Our approach, on the other hand, aims to take advantage of both recurrency and self-attention mechanisms while also reusing existing pre-training models in a modularized manner (Dathathri et al., 2020), saving time and computational power while enhancing its performance.

3. The proposed model

In this section, we describe in detail the proposed long text classifier model. We develop a hybrid of hierarchical and lateral MTGRU (HL-MTGRU) in a unified architecture for semantic sequence modeling. We extract the text features with the help of pre-trained Transformer encoders and feed the features directly to the HL-MTGRU. This architecture enables the network to learn multiple temporal scale dependencies from higher-order features. We hypothesize that the combination of slow and fast features will be beneficial for the text classification task with different lengths. The model also introduces recurrence into a pre-trained Transformer with the help of the HL-MTGRU network to enhance the performance on long text classification. Our model is designed to take advantage of both recurrency and the benefits of Transformers while also reusing existing pre-training models in a modularized manner, saving time and computational power while enhancing its performance.

3.1. The multiple timescales gated recurrent unit

The MTGRU shown in Fig. 1(a) is inspired by the concept of multiple timescales and temporal hierarchy found in the human brain (Botvinick, 2007; Ding et al., 2016; Meunier et al., 2010). RNNs are known to have the ability to deal with variable-length input sequences and discover long-term dependencies. A variety of RNNs have been proposed to enhance this ability with better memory storage and access. The most popular variants are LSTM (Hochreiter & Schmidhuber, 1997) and gated recurrent units (GRU) (Cho et al., 2014). In a recent study, Ding et al. (2016) demonstrated strong evidence for a neural tracking of hierarchical linguistic structures in the human brain. The study conducted experiments to determine whether neural representation of language (speech) tracks hierarchical linguistic structures, rather than prosodic and statistical transitional probability cues. The study found that the brain tracks and represents linguistic structures hierarchically. Earlier studies have also shown similar findings (Botvinick, 2007; Meunier et al., 2010), but Ding et al. (2016) was the first to confirm the same in the domain of language (speech). Ding et al. (2016) hypothesized that concurrent neural tracking of hierarchical linguistic structures provides a mechanism for temporally integrating smaller linguistic units into larger structures. For example, they found that the human brain is sensitive to the compositional structure of language. The activity of listeners' brains was recorded using magnetoencephalography (MEG) and when the human brain is presented with individual words, phrases, or a whole sentence, it was found that the brain activates or "spikes" at each level of composition. Therefore, our knowledge of the hierarchical nature of linguistic structures, and the theory of linguistic compositionality have been proved to be biologically plausible. Previous works have applied this hierarchical structure to RNNs in movement tracking (Paine & Tani, 2004), sensorimotor control systems (Yamashita & Tani, 2008) and speech recognition (Heinrich et al., 2012). The hierarchical MTGRU demonstrates the ability to capture multiple compositionalities similar to the findings of Ding et al. (2016) as shown by Moirangthem and Lee (2020). This better representation learning capability enhances the ability of the network to model longer sequences of text.

The multiple timescales in an MTGRU network is implemented with the help of a timescale variable implemented inside a conventional GRU, thereby adding another gating unit. The timescale gating unit essentially modulates the mixture of the past and current hidden states. Each step in an MTGRU takes as input $\mathbf{x}_t, \mathbf{h}_{t-1}$ and produces the hidden $\mathbf{h_t}$. The timescale τ added to the activation $\mathbf{h}_t$ of the MTGRU is shown in Eq. (1). τ is used to control the timescale of each GRU cell. Larger τ results in slower cell outputs but it makes the cell focus on the slow features, for example features from longer texts. On the other hand, smaller τ results in faster cells. One τ controls the slow cells and another τ controls the fast cells in each layer. The τ is made as a trainable variable like any other weight of the network and is optimized during the training based on the final loss (Moirangthem & Lee, 2020).

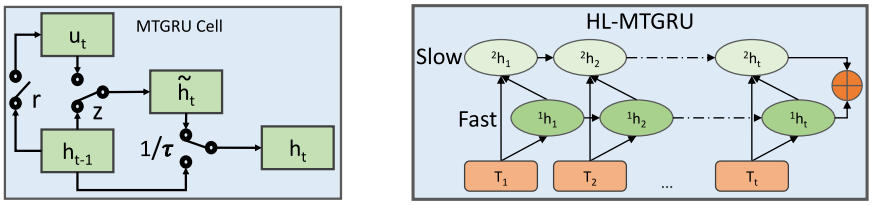
$$
\begin{aligned}
r_t &= \sigma(W_{xr}x_t + W_{hr}h_{t-1}) \\
z_t &= \sigma(W_{xz}x_t + W_{hz}h_{t-1}) \\
u_t &= \tanh(W_{xu}x_t + W_{hu}(r_t \odot h_{t-1})) \\
\tilde{h}_t &= z_t h_{t-1} + (1 - z_t)u_t \\
h_t &= \tilde{h}_t \frac{1}{\tau} + (1 - \frac{1}{\tau})h_{t-1}
\end{aligned}
\tag{1}
$$

where $\sigma(\cdot)$ and $\tanh(\cdot)$ are the sigmoid and tangent hyperbolic activation functions, $\odot$ denotes the element-wise multiplication operator, and $\mathbf{r}_t$, $\mathbf{z}_t$ are referred to as *reset*, *update* gates, respectively. $\mathbf{u}_t$ and $\tilde{\mathbf{h}}_t$ are the candidate activation and candidate hidden state of the MTGRU.

3.2. The hierarchical and lateral MTGRU layer

For the long text classification task, we develop a hierarchical and lateral multiple timescales architecture where half of the MTGRU units are fast and the remaining half are slow as shown in Fig. 1(b). The fast and slow units can capture different temporal dependencies from the input sequences. The fast timescales layer can capture fast changing features (e.g. character or word) whereas slower timescales can represent phrase or sentence level features (Moirangthem & Lee, 2020). The proposed HL-MTGRU structure follows a lateral (branch or root) architecture where the slow and fast units are directly connected to the inputs. This hierarchical architecture is introduced in the network by connecting the fast units to the slow units thereby making the HL-MTGRU to follow a multilayer structure. The HL-MTGRU structure is implemented using multiple single layer MTGRU networks whose timescales are different and the input to each layer comes directly from the input features. And the final output representation features of each layer are combined to form the penultimate representation of the input sequence that includes both fast and slow features.

The fast and slow layers of the HL-MTGRU network work differently. The fast layer is a basic MTGRU layer as shown in Eq. (1). The slow layer, however, needs to take the input features as well as the



(a) (b)

Fig. 1. (a) A Multiple Timescales Gated Recurrent Unit. The τ parameter is optimized for each layer and it controls the timescale of the layer. (b) The Hierarchical Lateral MTGRU with fast and slow layers.

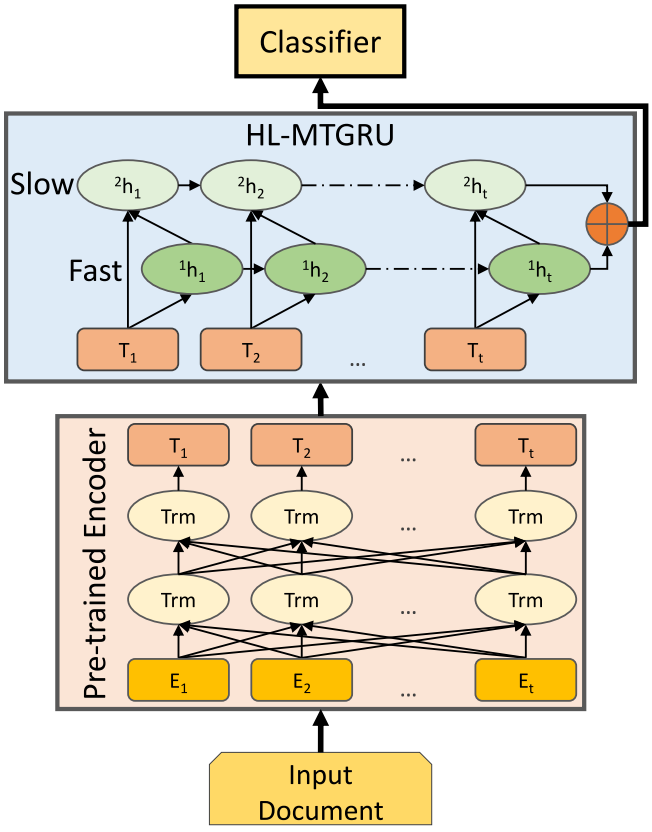


Fig. 2. The proposed HL-MTGRU classifier. The pre-trained encoder layer is used to extract word level features from the input text. The HL-MTGRU layer handles the dynamic features with multiple timescales hierarchy and produces a final representation of the input for classification.

fast layer features. Hence the slow layer is changed to accommodate additional connections as shown in Eq. (2).

$$\begin{aligned} r_t &= \sigma(W_{xr}x_t + W_{vr}v_t + W_{hr}h_{t-1}) \\ z_t &= \sigma(W_{xz}x_t + W_{vz}v_t + W_{hz}h_{t-1}) \\ u_t &= \tanh(W_{xu}x_t + W_{vu}v_t + W_{hu}(r_t \odot h_{t-1})) \\ \tilde{h}_t &= z_t h_{t-1} + (1 - z_t)u_t \\ h_t &= \tilde{h}_t \frac{1}{\tau} + (1 - \frac{1}{\tau})h_{t-1} \end{aligned} \tag{2}$$

where the W_{vr}, W_{vz}, and W_{vu} are introduced to accommodate the hierarchical connections of the fast and slow layers of the MTGRU network.

3.3. Pre-trained encoder layer

Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018) follows a new pre-training objective: the MLM and the "next sentence prediction" task. The MLM randomly masks some of the tokens from the input, and the objective is to predict the original word of the masked word based only on its context. The MLM objective allows the representation to combine the left and the right contexts and provides the way to train a deep bidirectional Transformer. This is different with the conventional left-to-right language model pre-training. In addition to the MLM objective, a "next sentence prediction" task is also used that jointly trains text representations from large unlabeled data.

A Lite BERT (ALBERT) (Lan et al., 2020) architecture has reduced the number of parameters from the traditional BERT model. ALBERT incorporates two parameter reduction techniques that lift the major obstacles in scaling pre-trained models. The first one is a factorized embedding parameterization and the second technique is cross-layer parameter sharing. This technique prevents the parameter from growing with the depth of the network. Both methods significantly reduce the number of parameters without sacrificing performance, thus proving to be a better alternative to BERT.

BERT and ALBERT model architectures are a multi-layer bidirectional Transformer encoder based on the original implementation described in Vaswani et al. (2017). Since we are using the BERT and ALBERT models from Devlin et al. (2018) and Lan et al. (2020), which are effectively identical to the original, we will omit an exhaustive background description of the model architecture.

Fine Tuning BERT takes an input of a sequence and outputs the representation of the sequence. The sequence has one or two segments that the first token of the sequence is always [CLS], which contains the special classification embedding and another special token [SEP] is used for separating segments. For text classification tasks, BERT takes the final hidden state h of the first token [CLS] as the representation of the whole sequence. A simple Softmax classifier is added to the top of BERT to predict the probability of label c:

$$p(c|h) = Softmax(Wh) \tag{3}$$

where W is the task-specific parameter matrix. We fine-tune all the parameters from BERT as well as W jointly by maximizing the log-probability of the correct label. The fine tuning method is also identical for ALBERT.

The proposed HL-MTGRU with the pre-trained encoder hybrid network consists of an ALBERT layer followed by the fast and slow HL-MTGRU layers as shown in Fig. 2. The fast units as well as the slow units are directly connected to the features of the pre-trained encoder. The fast unit features are also connected to the slow layer to incorporate a hierarchical connection. Finally, the combined last hidden representation of the HL-MTGRU is passed to a fully connected Softmax layer whose output is the probability distribution over the labels. The entire network is optimized on the final classification task.

4. The text classification datasets

In this section, we describe several benchmark datasets for text classification with diverse lengths of input text for gauging the performance

Dataset	Label	Sample
Yelp R.	+1	"Been going to Dr. Goldberg for over 10 years. I think I was one of his 1st patients when he started at MHMG. He's been great over the years and is really all about the big picture. [...]"
Amazon R.	3(/5)	"I love this show, however, there are 14 episodes in the first season and this DVD only shows the first eight. [...]. I hope the BBC will release another DVD that contains all the episodes, but for now this one is still somewhat enjoyable."
Yahoo A.	"Computer, Internet"	"What should I look for when buying a laptop? What is the best brand and what's reliable?,Weight and dimensions are important if you're planning to travel with the laptop. Get something with at least 512 mb of RAM. [..] is a good brand, and has an easy to use site where you can build a custom laptop."

Fig. 3. Examples of text samples and their labels.

Table 1
The details of the text classification datasets.

Dataset name	Classes	Training set	Testing set
Yelp P	2	560,000	38,000
Yelp F	5	650,000	50,000
Yahoo	10	1,400,000	60,000
Amazon F	5	3,000,000	650,000
Amazon P	2	3,6000,000	400,000
DBPedia	14	560,000	70,000

of our text classifier. Table 1 shows the statistics of the datasets that we have used. Some of the data samples are shown in Fig. 3.

DBpedia Ontology Classification The DBpedia ontology classification dataset (Zhang et al., 2015) is constructed by picking 14 non-overlapping classes from DBpedia 2014.

Amazon Review The Amazon review dataset (Zhang et al., 2015) is obtained from the Stanford Network Analysis Project (SNAP). This dataset contains review texts of extremely variate character lengths from 3 to 32,788 and hence, it is very suitable for evaluating our work.

Yahoo! Answers Topic Classification (Zhang et al., 2015) collected this data from the "Yahoo! Answers" corpus as of October 25th, 2007. It includes all the questions and their corresponding answers of various lengths, up to 4000 characters.

Yelp Reviews (Zhang et al., 2015) processed the Yelp Dataset Challenge 2015 dataset to perform two classification tasks. The first task predicts stars assigned by the user and the other task predicts the polarity label by considering stars, 1 and 2 as negative, and 3 and 4 as positive.

5. Experiments and results

We evaluate the performance of the proposed method and compare it to the conventional models using the text classification datasets described in Section 4.

5.1. Experiment settings

We train the proposed HL-MTGRU with ALBERT model in an end-to-end fashion. We use 256 units of MTGRU where half of the units are fast, and the remaining are slow units to construct the HL-MTGRU structure. We follow Moirangthem and Lee (2020) to initialize the timescale parameter τ for the fast units and the slow units to the default value of 1.0 and the τ parameter is also trained using the final loss. After training, the final τ values are 1.06 and 1.22 for the fast and the slow layers, respectively. The learning rate to update the τ, which is different from the global learning rate, is set to 0.00001 in order to avoid large changes in the timescales.

We follow Lan et al. (2020) and use the AdamOptimizer (Kingma & Ba, 2014) with a "weight decay rate" of 0.01, β_1 of 0.9, β_2 of 0.999, and ϵ of $1e$-6. We also use gradient clipping with "clip by global norm" of 1.0. The global learning rate is set to $2e$-5. For regularization, we employ a dropout of 0.5 on the HL-MTGRU layers to avoid overfitting. We utilize the validation performance for early stopping of the training for better generalization.

Training Process We expect a significant imbalance in the training process since the ALBERT is already pre-trained while the HL-MTGRU is initialized randomly and needs to be trained from scratch. In order to solve this issue, we implement a simple two-step training process to train the proposed model. We first train only the HL-MTGRU network using the features from ALBERT until we get a stable performance. Then, we train the entire network thereby, fine-tuning ALBERT. We utilized 2 Nvidia Quadro V100 GPUs for training our proposed model. The ALBERT model used in our architecture is the large version with 235M parameters.

5.2. Baseline models

We compare our model to both traditional and deep learning models in order to offer fair comparisons to existing models.

The traditional methods are those that use a hand-crafted feature extractor and a linear classifier. These include "Bag-of-words and its TFIDF", "Bag-of-ngrams and its TFIDF", and "Bag-of-means on word embedding". These methods use term-frequency inverse-document-frequency (TFIDF) (Salton & Buckley, 1988) in combination of bag-of-words or bag-of-ngrams model. We compare to the results of these models from the work of Zhang et al. (2015).

We also compare to several deep learning models that do not use any hand crafted features.

Table 2
Classification error (%) of our HL-MTGRU with ALBERT model against other methods on various multi-sentence classification benchmark datasets.

Model	Yelp P	Yelp F	Yahoo	Amazon F	Amazon P	DBP
BoW TFIDF	6.34	40.14	28.96	44.74	9.00	2.63
ngrams	4.36	43.74	37.53	45.73	7.98	1.37
ngrams TFIDF	4.56	45.20	31.49	47.56	8.46	1.31
LSTM	5.26	41.83	29.16	40.57	6.10	1.45
CNN	4.60	37.95	28.80	40.43	4.93	1.37
GoogleNet (2018)	9.55	43.55	24.90	40.35	3.01	–
DPCNN (2017)	2.64	30.58	23.90	34.81	3.32	–
D-LSTM (2017)	7.40	40.40	26.30	–	–	1.30
ULMFit (2018)	2.16	29.98	–	–	–	0.80
BERT (2019)	1.89	29.32	24.02	34.17	2.63	0.64
XLNet (2019)	1.37	27.05	21.20	31.67	2.11	0.60
GRU with ALBERT	1.42	27.31	22.31	31.69	2.17	0.62
ALBERT	1.39	27.11	22.37	31.67	2.13	0.61
Our Model	**1.31**	**26.89**	**21.14**	**30.82**	**2.01**	**0.60**

Baseline models We compare to the CNN and LSTM results from Zhang et al. (2015), Deep pyramid convolutional neural networks (DPCNN) (Johnson & Zhang, 2017), GoogleNet (Nawaz et al., 2018), D-LSTM (Yogatama et al., 2017), and ULMFit (Howard & Ruder, 2018).

Pre-trained Encoders We report the performance of BERT from Xie et al. (2019). The model configuration of the reported performance is BERT_{Large}, which has 334M parameters. We also compare to the performance of the current state-of-the-art XLNet model from Yang et al. (2019).

ALBERT We fine tune the ALBERT_{Large} model with the objective given in Eq. (3). The ALBERT structure is augmented with a classification layer.

GRU with ALBERT We also train a GRU model with ALBERT. This is almost identical to our proposed model, but instead of the HL-MTGRU, a GRU layer with 256 units is added and optimized.

5.3. Results

Table 2 shows the result of the comparison of our model with various other models using publicly available sentence classification datasets. These results illustrate that our proposed model either performed comparable to or outperformed existing models. The training curve of the proposed model compared to the BERT and ALBERT models shown in Fig. 4 demonstrates that our proposed model converges faster.

In order to differentiate the performance of the proposed HL-MTGRU with ALBERT model, the GRU with ALBERT model, and ALBERT, we divide the test data of the Yahoo Answers dataset according to the length of the texts and is evaluated using the trained models. Fig. 5 shows the comparison of the performance accuracy on different lengths of test data. As shown in the figure, the HL-MTGRU structure enables the model to outperform GRU and ALBERT on longer text inputs. It can also be seen that there is no significant performance degradation with the increase in input length whereas, the performance of GRU and ALBERT drop significantly with longer text inputs.

Statistical Significance Test We conduct the Wilcoxon signed-rank test to compare the results of our model to the results reported in the most recent baselines. This statistical significance test shows that the classification performance of our proposed model has a significant improvement over the baselines with $p < 0.05$. Table 3 shows the results of the statistical analysis.

Effect of Multiple Timescales We show further experiment results in Fig. 6 to demonstrate the effect of the fast and slow layers in our proposed classifier. We split the Yahoo data test set into two groups based on the length. The first group consists of sequences with less than 250 words and the other group is the rest of the data. We then take our trained HL-MTGRU network and separate the fast and slow layers to test the difference of performance on those two test data. As shown in Fig. 6, the network with only fast layer performs better than the one with only a slow layer for the shorter sequence data, and vice versa. We also observe that the combined hierarchical and lateral network performs significantly better in both the data groups. Hence, this result illustrates that fast and slow layers can capture rich dynamic features from diverse sequence length inputs, thereby increasing the overall classification performance.

Ablation Study In this study, we analyze the performance of the classifier with and without our proposed model features. The results

Table 3
Wilcoxon signed-rank test results comparing our model with the most recent baselines. The classification performance of our proposed model is shown to have a significant improvement in the comparison ($*\boldsymbol{p} < 0.05$).

Model 1	Model 2	Z-value	$*p$-value
Our Model	BERT (2019)	−2.201	0.028
Our Model	XLNet (2019)	−2.023	0.043
Our Model	ALBERT (2020)	−2.201	0.028
Our Model	GRU with ALBERT	−2.201	0.028

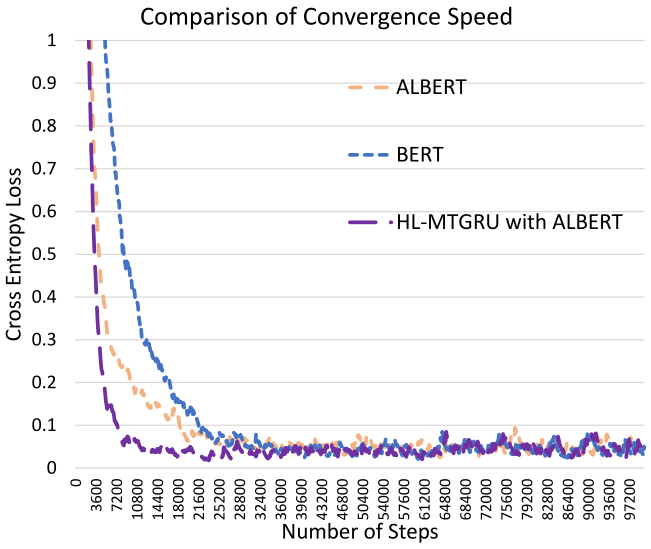


Fig. 4. Training performance comparison between the proposed model and conventional model.

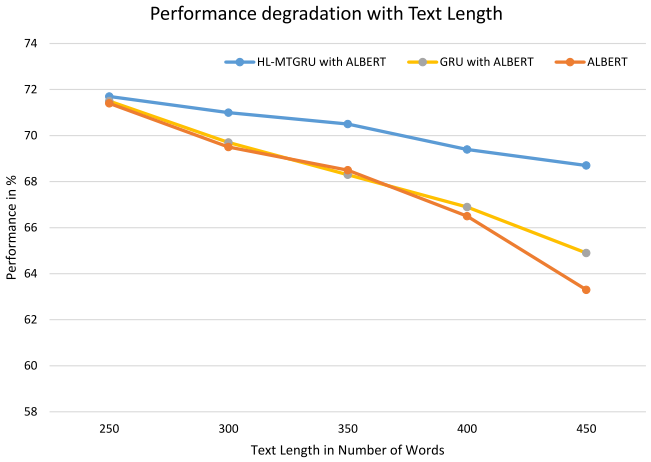


Fig. 5. Classification performance comparison based on the length of input.

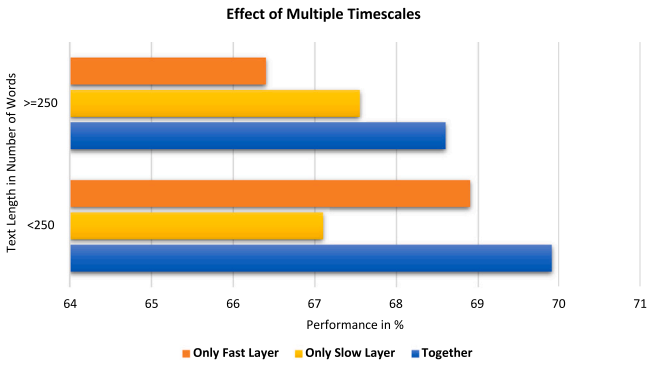


Fig. 6. The effect of multiple timescales in text classification and ablation study.

presented in Table 2, Figs. 5 and 6 can be interpreted as an ablation study of the proposed model. Table 2 shows that the HL-MTGRU with ALBERT significantly outperforms GRU with ALBERT, where GRU is the model without our proposed multiple timescales temporal hierarchy. Fig. 5 illustrates the advantage of our proposed multiple timescales approach in handling longer sequence of data while the GRU model without our proposed features fails to maintain the performance with increasing sequence length. Furthermore, we evaluate the performance of the proposed model with different temporal hierarchy components. In the first case, we train the model consisting only the fast layers. Secondly, we train the model with only slow layers and finally, the proposed model with one fast and one slow layer is trained. All of the three models are evaluated on short and long test data. The results are shown in Fig. 6 and it demonstrates the significance of fast and slow layers according to different length of input data. These results also show the importance of the HL-MTGRU model that combines both fast and slow features for the long text classification task.

6. Discussion

We have investigated in detail the difference in performance between the pre-trained Transformer models, a Transformer and GRU hybrid model, and our proposed model. GRU is the base model of our HL-MTGRU, and both models have been integrated with the large pre-trained ALBERT Transformer model in this work. The results shown in Table 2 illustrate that our HL-MTGRU model significantly outperforms the GRU model. The statistical significance test results given in Table 3 also show a significant increase of performance of our model compared to the recent state-of-the-art models. When we look at the results illustrated in Fig. 5, the performance of the proposed HL-MTGRU with ALBERT increased significantly compared to that of GRU with ALBERT. As shown in Eq. (1), we know that if τ is close to 1, which is the case of a fast HL-MTGRU layer, the model becomes a vanilla GRU. Therefore, a vanilla GRU is considered as a fast layer and hence, a GRU with ALBERT network can be considered as a network with only fast units. The difference in performance when we have all the RNN units as fast, i.e. GRU with ALBERT, and when we have a combination of slow and fast units, i.e. HL-MTGRU with ALBERT, shows the effectiveness of the multiple timescales approach. The results in Figs. 5 and 6 show the significance of the features from slow and fast layers, where the fast features help maintain the performance with shorter text inputs and the slow features enable the model to perform significantly better with longer text inputs. This confirms our hypothesis that the proposed HL-MTGRU with the help of both slow and fast units can help encode different dynamic features of a varying sequence length, in order to help improve the classification accuracy.

The results also illustrate that introducing recurrence into the pre-trained Transformer model can help improve the performance of these models. The recurrence modeling with the help of the HL-MTGRU outperformed other large pre-trained Transformer models and achieved the state-of-the-art on various benchmarks. The enhanced performance of the proposed model in such diverse length text datasets also confirms that the rich features of the slow and fast layers help in the discrimination task even with diverse sequence lengths.

7. Conclusion and future work

This paper addressed the issue of text classification for multiple lengths of text. We developed a hybrid model consisting of a HL-MTRGU network with a pre-trained encoder to classify different sets of text inputs. The proposed HL-MTGRU was able to effectively classify the texts inputs despite the variance in their length. We evaluated the performance of the proposed hybrid model on various benchmark text classification datasets with differing lengths in order to compare to several existing models. The results of our experiments illustrated that the proposed end-to-end learning hybrid network with multiple timescales not only performed significantly better in case of longer texts inputs but also maintained good performance in case of shorter texts.

In the future, we plan to utilize our classification model for chat discrimination to develop a hybrid system in order to improve such chatbot agents. This will also allow us to evaluate the effectiveness of our model for this application.

CRediT authorship contribution statement

Dennis Singh Moirangthem: Conceptualization, Methodology, Data curation, Writing - original draft, Software. **Minho Lee:** Supervision, Investigation, Validation, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partly supported by Institute of Information & Communications Technology Planning & Evaluation (IITP), South Korea grant funded by the Korea government (MSIT) (2016-0-00564, Development of Intelligent Interaction Technology Based on Context Awareness and Human Intention Understanding) (50%) and Technology Innovation Program: Industrial Strategic Technology Development Program (No: 10073162) funded By the Ministry of Trade, Industry & Energy (MOTIE, Korea) (50%).

References

Botvinick, M. M. (2007). Multilevel structure in behaviour and in the brain: a model of Fuster's hierarchy. *Philosophical Transactions of the Royal Society, Series B (Biological Sciences)*, *362*(1485), 1615–1626.

Chen, M. X., Firat, O., Bapna, A., Johnson, M., Macherey, W., Foster, G., Jones, L., Parmar, N., Schuster, M., & Chen, Z. (2018). The best of both worlds: Combining recent advances in neural machine translation. arXiv preprint arXiv:1804.09849.

Cho, K., van Merrienboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. CoRR, abs/1406.1078.

Chung, J., Ahn, S., & Bengio, Y. (2017). Hierarchical multiscale recurrent neural networks. In *Proceeding of the international conference on learning representations*.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., & Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. arXiv preprint arXiv:1705.02364.

Conneau, A., Schwenk, H., Barrault, L., & Lecun, Y. (2016). Very deep convolutional networks for text classification. arXiv preprint arXiv:1606.01781.

Cooijmans, T., Ballas, N., Laurent, C., Gülçehre, Ç., & Courville, A. (2017). Recurrent batch normalization. In *Proceeding of the international conference on learning representations*.

Dathathri, S., Madotto, A., Lan, J., Hung, J., Frank, E., Molino, P., Yosinski, J., & Liu, R. (2020). Plug and play language models: A simple approach to controlled text generation. In *International conference on learning representations*. URL: https://openreview.net/forum?id=H1edEyBKDS.

Dehghani, M., Gouws, S., Vinyals, O., Uszkoreit, J., & Kaiser, L. (2019). Universal transformers. In *International conference on learning representations*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Ding, N., Melloni, L., Zhang, H., Tian, X., & Poeppel, D. (2016). Cortical tracking of hierarchical linguistic structures in connected speech. *Nature Neuroscience*, *19*(1), 158–164.

Gao, J., Galley, M., & Li, L. (2019). Neural approaches to conversational AI. *Foundations and Trends® in Information Retrieval*, *13*(2–3), 127–298.

Ha, D., Dai, A., & Le, Q. V. (2017). HyperNetworks. In *Proceeding of the international conference on learning representations*.

Hao, J., Wang, X., Yang, B., Wang, L., Zhang, J., & Tu, Z. (2019). Modeling recurrence for transformer. arXiv preprint arXiv:1904.03092.

Heinrich, S., Weber, C., & Wermter, S. (2012). Adaptive learning of linguistic hierarchy in a multiple timescale recurrent neural network. In *International conference on artificial neural networks* (pp. 555–562). Springer.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. In *Proceedings of the 56th annual meeting of the association for computational linguistics* (Vol. 1: Long Papers). (pp. 328–339).

Johnson, R., & Zhang, T. (2017). Deep pyramid convolutional neural networks for text categorization. In *Proceedings of the 55th annual meeting of the association for computational linguistics* (Vol. 1: Long Papers). (pp. 562–570).

Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A convolutional neural network for modelling sentences. In *Proceedings of the 52nd annual meeting of the association for computational linguistics* (Vol. 1: Long Papers) (Vol. 1). (pp. 655–665).

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *EMNLP* (pp. 1746–1751). Association for Computational Linguistics.

Kim, M., Dennis Singh, M., & Lee, M. (2016). Towards abstraction from extraction: Multiple timescale gated recurrent unit for summarization. In *1st Rep4NLP* (pp. 70–77). Association for Computational Linguistics.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Skip-thought vectors. In *Advances in neural information processing systems* (pp. 3294–3302).

Krueger, D., & Memisevic, R. (2016). Regularizing RNNs by stabilizing activations. In *Proceeding of the international conference on learning representations*.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). ALBERT: A lite BERT for self-supervised learning of language representations. In *International conference on learning representations*.

Le, Q. V., & Mikolov, T. (2014). Distributed representations of sentences and documents. In *ICML* (Vol. 14). (pp. 1188–1196).

Lin, Z., Feng, M., Santos, C. N. d., Yu, M., Xiang, B., Zhou, B., & Bengio, Y. (2017). A structured self-attentive sentence embedding. arXiv preprint arXiv:1703.03130.

Liu, P., Qiu, X., & Huang, X. (2016). Recurrent neural network for text classification with multi-task learning. arXiv preprint arXiv:1605.05101.

Logeswaran, L., & Lee, H. (2018). An efficient framework for learning sentence representations. arXiv preprint arXiv:1803.02893.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies (Vol. 1)* (pp. 142–150). Association for Computational Linguistics.

McCann, B., Bradbury, J., Xiong, C., & Socher, R. (2017). Learned in translation: Contextualized word vectors. In *Advances in neural information processing systems* (pp. 6294–6305).

Meunier, D., Lambiotte, R., Fornito, A., Ersche, K. D., & Bullmore, E. T. (2010). Hierarchical modularity in human brain functional networks. arXiv preprint arXiv:1004.3153.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems (Vol. 26)* (pp. 3111–3119). Curran Associates, Inc.

Moirangthem, D. S., & Lee, M. (2017). Temporal hierarchies in multilayer gated recurrent neural networks for language models. In *Neural networks (IJCNN), 2017 international joint conference on* (pp. 2152–2157). IEEE.

Moirangthem, D. S., & Lee, M. (2020). Abstractive summarization of long texts by representing multiple compositionalities with temporal hierarchical pointer generator network. *Neural Networks*, *124*, 1–11.

Moirangthem, D. S., Son, J., & Lee, M. (2017). Representing compositionality based on multiple timescales gated recurrent neural networks with adaptive temporal hierarchy for character-level language models. In *Proceedings of the 2nd workshop on representation learning for NLP* (pp. 131–138). Association for Computational Linguistics.

Nawaz, S., Calefati, A., Janjua, M. K., & Gallo, I. (2018). Seeing colors: Learning semantic text encoding for classification. arXiv preprint arXiv:1808.10822.

Paine, R. W., & Tani, J. (2004). Motor primitive and sequence self-organization in a hierarchical recurrent neural network. *Neural Networks*, *17*(8–9), 1291–1309, New Developments in Self-Organizing Systems.

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, *2*(1–2), 1–135.

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing*. (pp. 1532–1543).

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: Human language technologies* (Vol. 1) (Long Papers). (pp. 2227–2237).

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. URL https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/languageunsupervised/languageunderstandingpaper.pdf.

Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998). A Bayesian approach to filtering junk e-mail. In *Learning for text categorization: Papers from the 1998 workshop* (Vol. 62). (pp. 98–105). Madison, Wisconsin.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, *24*(5), 513–523.

Seo, M., Min, S., Farhadi, A., & Hajishirzi, H. (2017). Neural speed reading via skim-rnn. arXiv preprint arXiv:1711.02085.

Shen, D., Zhang, Y., Henao, R., Su, Q., & Carin, L. (2018). Deconvolutional latent-variable model for text sequence matching. In *Thirty-second AAAI conference on artificial intelligence*.

Tran, K., Bisazza, A., & Monz, C. (2018). The importance of being recurrent for modeling hierarchical structure. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 4731–4736). Brussels, Belgium: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/D18-1503, URL: https://www.aclweb.org/anthology/D18-1503.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).

Wang, Z., Ma, Y., Liu, Z., & Tang, J. (2019). R-transformer: Recurrent neural network enhanced transformer. arXiv preprint arXiv:1907.05572.

Wang, S., & Manning, C. D. (2012). Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th annual meeting of the association for computational linguistics: Short papers (Vol. 2)* (pp. 90–94). Association for Computational Linguistics.

Xie, Q., Dai, Z., Hovy, E., Luong, M. T., & Le, Q. V. (2019). Unsupervised data augmentation for consistency training. arXiv preprint arXiv:1904.12848.

Yamashita, Y., & Tani, J. (2008). Emergence of functional hierarchy in a multiple timescale neural network model: A humanoid robot experiment. *PLoS Computational Biology*, *4*(11), 1–18.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems* (pp. 5754–5764).

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016). Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Human language technologies*. (pp. 1480–1489).

Yogatama, D., Dyer, C., Ling, W., & Blunsom, P. (2017). Generative and discriminative text classification with recurrent neural networks. arXiv preprint arXiv:1703.01898.

Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Advances in neural information processing systems* (pp. 649–657).