A Multi-layer Bidirectional Transformer Encoder for Pre-trained Word Embedding: A Survey of BERT

Rohit Kumar Kaliyar
Dept. Of Computer Science
Engineering
Bennett University
Greater Noida, India
rk5370@bennett.edu.in

***Abstract*—Language modeling is the task of assigning a probability distribution over sequences of words that matches the distribution of a language. A language model is required to represent the text to a form understandable from the machine point of view. A language model is capable to predict the probability of a word occurring in the context-related text. Although it sounds formidable, in the existing research,most of the language models are based on unidirectional training. In this paper, we have investigated a bi-directional training model-BERT (Bidirectional Encoder Representations from Transformers). BERT builds on top of the bidirectional idea as compared to other word embedding models (like Elmo). It practices the comparatively new transformer encoder-based architecture to compute word embedding. In this paper, it has been described that how this model is to be producing or achieving state-of-the-art results on various NLP tasks. BERT has the capability to train the model in bi-directional over a large corpus. All the existing methods are based on unidirectional training (either the left or the right). This bi-directionality of the language model helps to obtain better results in the context-related classification tasks in which the word(s) was used as input vectors. Additionally, BERT is outlined to do multi-task learning using context-related datasets. It can perform different NLP tasks simultaneously. This survey focuses on the detailed representation of the BERT- based technique for word embedding, its architecture, and the importance of this model for pre-training purposes using a large corpus.**

***Index Terms*—Language modeling, Bidirectional Encoder, BERT, NLP**

I. INTRODUCTION

Over the previous years, distributed linguistics representations [1] have incontestable to be effective and flexible supervisors of previous learning to be incorporated into downstream context-related real-world applications [1,4,9,20]. We managed to drive from the theoretic framework behind word vector space models [4, 7, 9] and have one in each of their real-world limitations [4]. The importance of conflation deficiency [1, 9, 22], that arises from expressing to a word with all its possible meanings as one single vector. At that time, we conduce to explain yet this deficiency may be self-addressed through a transformation from the word level to a lot of fine-grained level of words for representing unambiguous lexical significance [5, 7, 9]. We tend to begin a comprehensive survey to define the wide scope of strategies [4,9,10,13,15] within the two elementary components of word illustration, i.e., unsupervised and learning-based [1,8,19].

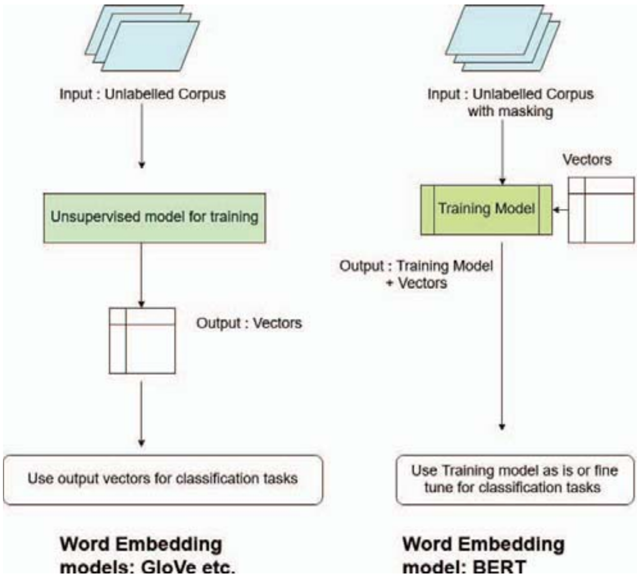
Input : Unlabelled Corpus
Unsupervised model for training
Output : Vectors
Use output vectors for classification tasks
Word Embedding models: GloVe etc.
Input : Unlabelled Corpus with masking
Vectors
Training Model
Output : Training Model + Vectors
Use Training model as is or fine tune for classification tasks
Word Embedding model: BERT

Fig. 1. Unsupervised Training Approaches (word-embeddings)

Traditional word-based vectors [4, 7, 18] are similar to shallow representations [3, 9, 12] (which consist of a single layer of weights, known as an embedding). These vectors use previous knowledge in the first layer of the model for feature-based training. We need to train the rest of the network from scratch for a new target task. They normally fail to capture context-level information that might be even more useful for training. Word embeddings are useful in only capturing semantic meanings of words, but we also need to understand higher-level concepts like long-term dependencies, negation, etc.

Word embedding is a pre-trained language representation model where the objective function [4, 9,15] consists typically

978-1-7281-2791-0/20/$31.00 ©2020 IEEE

in predicting the next word for a given context. The task of pre-training the language representation model is typically performed off-line with frameworks like Genism or word2vec [3,5,8] using large corpus like Google News corpus[16] (3 billion running words, 300 dimension English word vectors). Static Word Embeddings [4, 7, 8] failed to capture the context-related features [1]. These types of embeddings are capable to generate the same word in different contexts. The main objective of contextualized words embeddings [4,7,9,14] is to capturing words semantics in different contexts to address the issue of polysemous [9]. Static Word Embeddings [4,12,17] could only leverage off the vector outputs from unsupervised models for downstream tasks not the unsupervised models themselves. They were mostly shallow models [1,9,13] to begin with and were often discarded after training (e.g. word2vec, Glove) The output of Contextualized (Dynamic) Word Embedding [4,7,9] training is the trained model and vectors—not just vectors. We can observe the same under-standing from figure1. BERT-based modeling is also explained compared to other pre-trained word embedding models.

BERT (Bidirectional Encoder Representations from Transformers) was released in late 2018 [25] for the first time for pre-training. In this paper, we have investigated about BERT representations for a deeper understanding of encoder architecture and practical guidance [25] for using transfer learning models in NLP-based context-related tasks. In this research, we have also investigated the architecture of BERT, which makes it a powerful feature extractor. Technological implementations have also been discussed in this research. BERT is a strategy for pre-training language representations that were utilized to make efficient models that NLP experts would be able to download and use pre-training. We utilize these models to extract high-quality language features from your content information, or we can fine-tune these models on a particular task (classification, substance recognition [2, 25], etc.) with your information to deliver best in class predictions.

BERT is unsupervised in nature. It is a bi-directional pre-trained word embedding model used for various NLP-based tasks. It is built in such a way that it can handle a large plain text corpus forpre-training.Itgained huge attention at the time of its releaseandshownstate-of-the-art results on 11 different NLP-based tasks. BERT is a baseline model that considers the next sentence from the left sides and therefore the right sides of every word for training the model. The two key factors of BERT for better results are (1) Masking a par to fin formation's-tokens to avoid cycles and (2) Pre-training a sentence using a relationship model. Finally, we can observe that BERT is a major and important model trained on a huge word-based corpus for efficient classifications [25].

II. FUNCTIONALITY OFBERT

From figure 2, we can observe the practicality of the BERT [25] that are used to extract features, particularly word and sentence embedding vectors [25], from the whole extracted text information. Initially, these embeddings are helpful for keywordexpansion,linguisticssearch,andknowledgeretrieval for pre-training [1,25,30] the model. As a general example, if you wish to meet client queries or explorations against previously clarified questions [25], these representations can assist you more accurately for retrieving better results in terms of matching the customer's intention although there's no keyword or phrase overlap.

Embedding vectors are very helpful to utilize top quality features as inputs to train our models. Deep learning models [2,19,25], like LSTMs or CNN's, require inputs in the form of numerical vectors. These vectors are needed for training and translating the required features. These features arein the form of vocabulary-based features. Secondly, grammatical features are essential for better representations in numerical form. in this approach, words have been represented in, either as indexed values [25] or as neural word embedding, where vocabulary words are fixed-length feature embeddings that outcome from models like Word2Vec orFast-Text.

The core functionality of BERT is that it offers a margin of space over word embedding models like word2Vec because each word has a fixed representation. In word2Vec model, the functionality is notwithstanding the context inside which the word appears. BERT produces a word representation that is progressively implicated by the words around them with the help of masking.

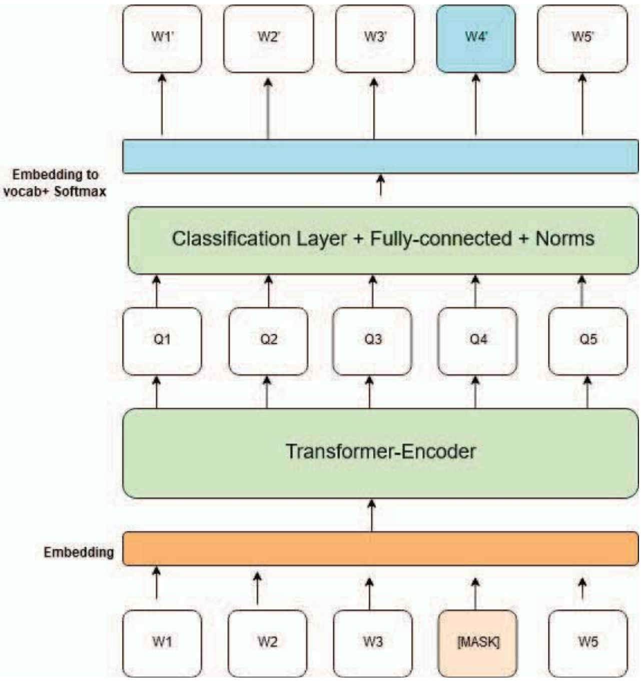
W1'
W2'
W3'
W4'
W5'
Embedding to vocab+ Softmax
Classification Layer + Fully-connected + Norms
Q1
Q2
Q3
Q4
Q5
Transformer-Encoder
Embedding
W1
W2
W3
[MASK]
W5

Fig. 2. Classification using BERT

A. Comparison to others pre-trained word embedding models

Glove and Word2vec are word-based models [1, 17, 25] – such type of models which take words as input and the output will be word embedding vectors. Elmo, in contrast, is a character-based model [26] using character convolutions and can handle out of vocabulary words for this context. The learned representations are words, however (shown in the table below). BERT represents input as sub words and learns embeddings for sub words. So, it has a vocabulary that is about 30,000 for a model trained a corpus with a large number of unique words (millions) - which is much smaller in contrast to a Glove,Word2vec,or Elmo model trained on the same corpus. Representing input as sub words as opposed to words has become the latest trend because it strikes a balance between character-based and word-based representations [25,26] - the most important benefit being the avoidance of OOV (out of vocabulary) cases which the other two models (Glove, Word2vec) mentioned in the question suffer from. There has been recent work that character-based language models do not perform as well as word-based models for large corpus, which is perhaps an advantage of word-based models have over character-based input models like Elmo.

III. BERT TRANSFORMER ARCHITECTURE

Transformer architecture is shown in figure3. In this model architecture, Encoder, a Decoder and the Encoder'soutput is an input to the next Decoder. At its initial phase, it was trained by solving a different translating problem (translating English to German). We can observe from this figure that the transformer is capable of performing self-attention between all the Encoder input tokens since none of these are part of the prediction (the prediction was the translated German sequence, and this was masked for better prediction). BERT's word vector output encodes a rich linguistic structure. BERT approximately encodes syntax trees in the word embeddings it outputs for a sentence. It is possible to recover these trees by a linear transformation of the word embeddings. BERT appears to encode syntactic and semantic features inword vectors in complementary subspaces. Different meaning of a word has in different representations (determined by the sentence-related context) that are spatially separated in a fine- grained manner. From Table I and II, we can observethe representations and parameters for all recent pre-trained word embedding techniques.

IV. RELATEDWORK

Word embedding, otherwise called word representation, speaks of a word as a vector capturing both syntactic and semantic information, so the words with comparable meanings ought to have comparative vectors [4]. Although, classic embedding models, for example, Word2Vec [1,7,13,16], GloVe [2,5,8], fast Text [6], have been appeared to help improve the exhibition of existing models in an assortment of assignments like parsing [3,7], topic modeling [4,11,18], anddocument classification [1,6,13]. Each word is related to a single vector prompting a test on utilizing the vector in differing across linguistic contexts [19, 21]. To conquer that issue, a modern and advanced pre-trained word embedding is relevant embeddings (e.g., ELMO of [19], BERT of Devlin et al. [14,17,22] have been proposed and enables existing models to accomplish new best in class results on numerous NLP undertakings. Not quite the same as non-contextual embeddings, ELMO and BERT can capture latent syntactic-semantic information of a similar word dependent on its contextual uses.

,

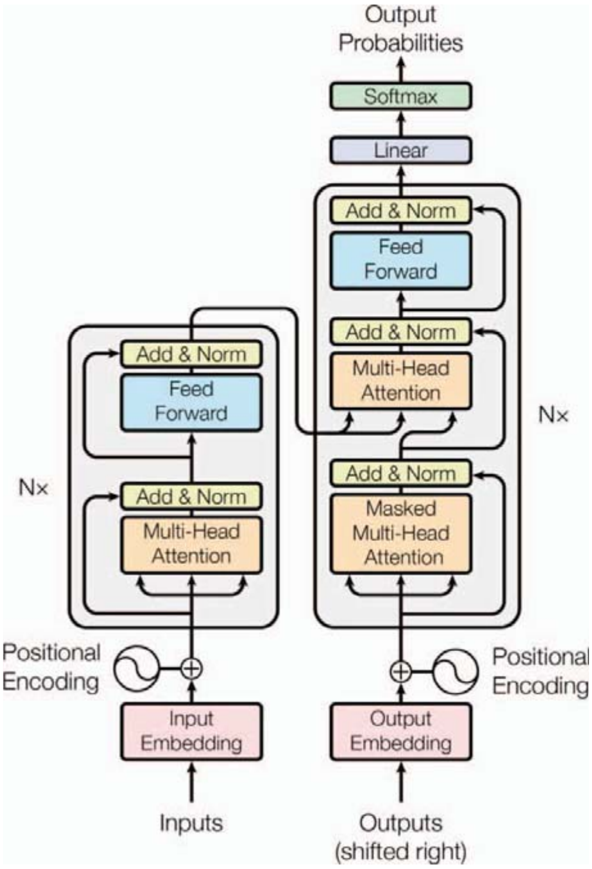

Fig. 3. BERT Transformer Architecture [25]

Accordingly, this paper joins both old-style embeddings (i.e., Word2Vec, fast Text) and new and efficient embeddings (i.e., ELMO, BERT) to assess their functionalities. Pre-trainedword representations [6,8] are a key part of numerous neural language models. In any case, adapting high-quality representations can be challenging. They have described a perfect model for both complex attributes of word used as input features (for example punctuations and semantic attributes) andvarious linguistic context-related features. In this paper, we presented a new kind of deeply contextualized word representation that legitimately addresses the two challenges, can be effectively incorporated into existing models, and fundamentally improves the state of the art in each considered case over a scope of challenging language-understanding issues. Representation of earlier pre-trained models shown in figure 4. A bi-directional model was needed for pre-trained on large corpus efficiently.

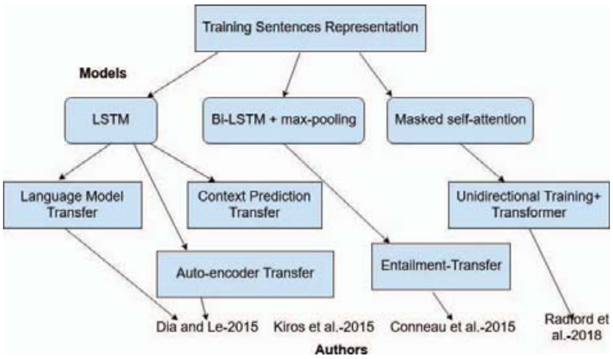

Fig. 4. Representations of earlier pre-trained models

Adapting generally appropriate representations of words has been a functioning territory of research for decades, including non-neural [2,4,9] and neural methods [22,23]. Pre-trained word embeddings are a basic piece of present-day NLP frameworks, offering critical enhancements over embeddings learned without any feature representation. To pre-trained word embeddings vectors, left-to-right language demonstrating destinations have been utilized [4,7], just as goals to the discriminate right from off base words in the left and right context [3]. These methodologies have been generalized up to coarser granularities, for example, sentence embeddings [24] or sentence embeddings [22].

A. Real-Time Natural Language Understanding with BERT Using TensorRT

BERT provided a leap in accuracy for NLP tasks that brought high-quality language-based services [24,25] within the reach of companies across many industries. To use the model in production, we need to consider factors such as latency, in addition to accuracy, which influences end-user satisfaction with a service. BERT requires significantcompute during inference due to its 12/24-layer stacked multi-head attention network. This has posed a challenge for companies to deploy BERT as part of real-time applications until now.

V. LIMITATIONS

In the field of advanced word embedding, numerous models (like Elmo and BERT) have been investigated. At the initial stage, pre-training in NLP was limited around the word embedding models like GloVe and word2Vec. Pre-trained word embedding models are generally used to train any model on the large-sized corpus. We use these trained models on labelled data for downstream context-related tasks. A better example is sentiment analysis. This process allows the pre-trained models to consist of semantic information that is learned from large-sized datasets. Word embeddings are very essential across a wide range of context-related tasks, but the rear numerous other limitations exists.

One important limitation in word embedding models is that these models is a very powerful in all context-related tasks. For example, word2Vec model is trained for shallow language modelling tasks. There exist many other limitations related to other word embedding models. We need dense architectures like LSTMs with these types of word embedding models to capture the combination of different combinations of input vectors based on different words and negations.

A major limitation with the word embedding models that it does not fit into every context-related task. For example, in a given sentence, a single word consists of different meanings. In the given sentence (I stole money from the bank), the word "bank" has a different meaning as compared in another given sentence (The bank of the river overflowed with water). Another major difference between traditional word embedding models and other word embedding models is that in the traditional word embedding a single vector for each word is allocated, which is used to represent the different meanings. These limitations have motivated the use of deep language models (language models that use dense architectures like LSTMs) for transfer learning. These models are very useful to train a complex deep neural network with the mapping a vector of each word based on the entire context-related sentence.Few examples of these type of models are Elmo, ULMFiT, and BERT.

TABLE I
REPRESENTATIONS OF PRE-TRAINED WORD EMBEDDINGS

Model	contentsensitiveembedding	Learnt Representation
Word2Vec	NO	words
GLoVe	NO	words
ELMo	Yes	words
BERT	Yes	sub words

TABLE II
PARAMETERS FOR BERT-BASE TECHNIQUES

Model	layers	hiddensize	attentionheads	Total parameters
BERT-Base	12	768	12	110M
BERT-Large	24	1024	16	340M

VI. CONCLUSION

Even though BERT is a powerful feature extractor compared to BiLSTM model on transfer encoder, over a large corpus, BERT holds significantly longer training andinference time. It also consists of large memory requirements. These practical concerns could be mitigated by designing fine-tuned BERT model for future research. With small-sized dataset, the performance of BERT increases become more sensational. It indicates that using pre-trained networks like BERT might be essential for achieving performance in suchcontext-related tasks. Smaller sizes are more common for datasets with more complex questions, and we believe pre-trained networks like BERT can have a bigger impact when training with all available data there. In the future, we plan to provide a better analysis of the fine-tuned BERT's attention and try to provide some insight into its learned decision process as well as its limitations. We also plan to investigate the robustness of existing methods as well as BERT-based models to typing mistakes and variation in expressing questions (e.g. Using synonyms). In the future, we would like to investigate various transfer structures on the top of pre-trained BERT, especially for the sake of enhancing the stability of the fine-tuning process. We observe in our investigation that the performance of fine-tune models based on BERT strongly depends on the initial random state, thus, further research on building more robust models is indispensable.

REFERENCES

[1] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).

[2] Bollacker, Kurt, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. "Freebase: a collaboratively created graph database for structuring human knowledge." In Proceedings of the 2008 ACM SIGMOD international conference on Management of data, pp. 1247-1250. AcM, 2008.

[3] Bordes, A., Usunier, N., Chopra, S., Weston, J.: Large-scale simple ques- tion answering with memory networks. arXiv preprintarXiv:1506.02075 (2015)

[4] Dai, Zihang, Lei Li, and Wei Xu. "Cfo: Conditional focused neural question answering with large-scale knowledge bases." arXiv preprint arXiv:1606.01994 (2016).

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

[6] Golub, David, and Xiaodong He. "Character-level question answering with attention." arXiv preprint arXiv:1604.00727 (2016).

[7] Sak, Haşim, Andrew Senior, and Françoise Beaufays. "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition." arXiv preprint arXiv:1402.1128 (2014).

[8] Howard, Jeremy, and Sebastian Ruder. "Universal language model fine-tuning for text classification." arXiv preprint arXiv:1801.06146 (2018).

[9] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).

[10] Liu, Xiaodong, Pengcheng He, Weizhu Chen, and Jianfeng Gao. "Multi-task deep neural networks for natural language understanding." arXiv preprint arXiv:1901.11504 (2019).

[11] Lukovnikov, Denis, Asja Fischer, Jens Lehmann, and Sören Auer. "Neural network-based question answering over knowledge graphs on word and character level." In Proceedings of the 26th international conference on World Wide Web, pp. 1211-1220. International World Wide Web Conferences Steering Committee, 2017.

[12] Maheshwari, Gaurav, Priyansh Trivedi, Denis Lukovnikov, Nilesh Chakraborty, Asja Fischer, and Jens Lehmann. "Learning to rank query graphs for complex question answering over knowledge graphs." In International Semantic Web Conference, pp. 487-504. Springer, Cham, 2019.

[13] Mohammed, Salman, Peng Shi, and Jimmy Lin. "Strong baselines for simple question answering over knowledge graphs with and without neural networks." arXiv preprint arXiv:1712.01969 (2017).

[14] Pennington, Jeffrey, Richard Socher, and Christopher Manning. "Glove: Global vectors for word representation." In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532-1543. 2014

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

[16] Petrochuk, Michael, and Luke Zettlemoyer. "Simplequestions nearly solved: A new upperbound and baseline approach." arXiv preprint arXiv:1804.08798 (2018).

[17] Radford, Alec, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. "Improving language understanding by generative pre-training." URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf (2018).

[18] Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. "Language models are unsupervised multitask learners." OpenAI Blog 1, no. 8 (2019).

[19] Ture, Ferhan, and Oliver Jojic. "No Need to Pay Attention: Simple Recurrent Neural Networks Work!(for Answering" Simple" Questions)." arXiv preprint arXiv:1606.05029 (2016).

[20] Caliskan, Aylin, Joanna J. Bryson, and Arvind Narayanan. "Semantics derived automatically from language corpora contain human-like biases." Science 356, no. 6334 (2017): 183-186.

[21] Clark, Kevin, and Christopher D. Manning. "Improving coreference resolution by learning entity-level distributed representations." arXiv preprint arXiv:1606.01323 (2016).

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

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

[24] Clevert, Djork-Arné, Thomas Unterthiner, and Sepp Hochreiter. "Fast and accurate deep network learning by exponential linear units (elus)." arXiv preprint arXiv:1511.07289 (2015).

[25] Finkel, Jenny Rose, Trond Grenager, and Christopher Manning. "Incorporating non-local information into information extraction systems by gibbs sampling." In Proceedings of the 43rd annual meeting on association for computational linguistics, pp. 363-370. Association for Computational Linguistics, 2005.

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

[27] Ling, Wang, Isabel Trancoso, Chris Dyer, and Alan W. Black. "Character-based neural machine translation." arXiv preprint arXiv:1511.04586 (2015).