Proceedings of the Federated Conference on Computer Science and Information Systems pp. 179–183
DOI: 10.15439/2020F20
ISSN 2300-5963 ACSIS, Vol. 21

Overview of the Transformer-based Models for NLP Tasks

Anthony Gillioz
University of Neuchâtel
Neuchâtel, Switzerland
Email: anthony.gillioz@unine.ch

Jacky Casas, Elena Mugellini, Omar Abou Khaled
University of Applied Sciences and Arts Western Switzerland
Fribourg, Switzerland
Email: {firstname.lastname}@hes-so.ch

***Abstract*—In 2017, Vaswani et al. proposed a new neural network architecture named Transformer. That modern architecture quickly revolutionized the natural language processing world. Models like GPT and BERT relying on this Transformer architecture have fully outperformed the previous state-of-the-art networks. It surpassed the earlier approaches by such a wide margin that all the recent cutting edge models seem to rely on these Transformer-based architectures.**

In this paper, we provide an overview and explanations of the latest models. We cover the auto-regressive models such as GPT, GPT-2 and XLNET, as well as the auto-encoder architecture such as BERT and a lot of post-BERT models like RoBERTa, ALBERT, ERNIE 1.0/2.0.

I. Introduction

THE understanding and the treatment of the ubiquitous textual data is a major research challenge. The tremendous amount of data produced by our society through social media and companies has exploded over the past years. All those information are most of the time stored under textual format. The human brain can extract the meaning out of text effortlessly, but this is not the case for a computer. It is then required to have performing and reliable techniques to treat this data.

The Natural Language Processing (NLP) domain aims to provide a set of techniques able to explain a wide variety of Natural Language tasks such as Automatic Translation [1], Text Summarization [2], Text Generation [3]. All those tasks have in common the meaning extraction process to be successful. Undoubtedly, if a technique were able to understand the underlying semantic of texts, this would help to resolve the majority of the modern NLP problems.

A big concern that restricts a general NLP resolver is the single-task training scheme. Gathering data and crafting a specific model to solve a precise problem works successfully. However, it forces us to come up with a solution not only each time a new issue arises but also to apply the model on another domain. A general multi-task solver may be preferable to avoid this time-consuming point.

Recurrent Neural Networks (RNN) were massively used to solve NLP problems. They have been popular for a few years in supervised NLP models for classification and regression. The success of RNNs is due to the Long Short Term Memory (LSTM) [4] and Gated Recurrent Unit (GRU) [5] architectures. Those two units prevent the vanishing gradient issue by providing a more direct way to the backpropagation of the gradient. It helps the computation when the sentences are long.

The high versatility of those networks can solve a wide variety of problems [6]. Unfortunately, those models are not perfect; the inherent recurrent structure made them hard to parallelize on multiple processes, and the treatment of very long clauses is also problematic due to the vanishing gradient.

To counter those two limiting constraints, [7] introduced a new model architecture: the Transformer. The proposed technique get rid of the recurrent architecture to rely on attention mechanism solely. Furthermore, it does not suffer from the gradient vanishing nor the hard parallelization issue. That facilitates and accelerates the training of broader networks.

This work aims to provide a survey and an explanation of the latest Transformer-based models.

II. Background

In this section, we introduce a general NLP background. It gives a broad insight into the unsupervised pre-training and the NLP state-of-the-art pre-Transformers.

A. Unsupervised Pre-training

The unsupervised pre-training is a particular case of semi-supervised learning. That is massively used to train the Transformer models. That principle works in two steps; the first one is the pre-training phase. It computes a general representation from raw data in an unsupervised fashion. Second, once it is computed, it can be adapted to a downstream task via fine-tuning techniques.

The principal challenge is to find an unsupervised objective function that generates a good representation. There is no consensus on which task provides the most efficient textual description. [8] propose a language modelling task, [9] introduce a masked language modeling objective, [10] use a multi-tasks language modeling.

B. Context-free representation

The recent significant increase in the performance of NLP models is due to the use of word embeddings. It consists of representing a word as a unique vector. The terms with the same meaning are located in a close area of each other. Word2Vec [11] and Glove [12] are the most frequently used word embedding methods. They treat a large corpus of text and

produce a unique word representation in a high dimensional space.

Byte Pair Encoding (BPE) [13] is another word embedding technique using subwords units out of character-level and word-level representation. [14] changed the implementation of BPE to be based on bytes instead of Unicode characters. Thus, he could reduce the vocabulary size from 100K+ to approximately 50K tokens. That has the advantage not to introduce [UKN] (unknown) symbols. Besides that, it does not involve a heuristic preprocessing of the input vocabulary. It is used when the amount of corpus to treat is too large and a more efficient technique than Word2Vec or Glove is required.

C. Attention Layer

Primarily proposed by [5], the attention mechanism aims to catch the long-term dependencies of sentences. The relationships between entities in phrases are hard to spot. Furthermore, it is necessary to get a strong understanding of the underlying structure of sentences. Indeed, if we can have a method that can tell us how the units of a sentence are correlated in a phrase, the language understanding tasks would be more straightforward.

The attention mechanism computes a relation mask between the words of a sentence and uses this mask in an encoder-decoder architecture to detect which words are related within each other. Using this process, the NLP tasks such as automatic translation are more flexible because they can have access to the dependencies of the sentence. In a translation context, it is a genuine advantage. Another notable benefit of the attention mechanism is the straightforward human-visualization of the model's outcome.

III. Dataset

The dominant strategy in the creation of deep learning systems is to gather a corpus corresponding to a given problem. The next step is to label this data and build a network that is supposedly able to explain them. This method is not suitable if we want to create a more comprehensive system (i.e. a system that can solve multiple problems without a significant architecture change).

That is then essential to learn on heterogeneous data to create general NLP models. If we want systems that can resolve several tasks at the same time, it is necessary to train this model on a wide variety of subjects. Hopefully, in our ubiquitous data world, a large number of raw texts are available online (e.g. Wikipedia, Web blogs, Reddit).

Table I shows the most commonly used datasets with their size and the number of tokens they contain. The tokenization is done with SentencePiece [15]. In a few cases, for example, in [16], the authors only used a subset of those datasets (e.g. Stories [17] is a subset of CommonCrawl dataset).

TABLE I
Datasets commonly used with Transformer-based models. (†: Tokenization done with SentencePiece, ‡: Uncompressed data)

Dataset	Size	Number of tokens †
BookCorpus [18]		
plus English Wikipedia	13GB	3.87B
Giga5 [19]	16GB	4.75B
ClueWeb09 [20]	19GB	4.3B
OpenWebText [21]	38GB	-
Real-News [22]	120GB ‡	-

IV. Benchmarks

During an extended period, the deep learning models have been trained to resolve one problem at a time. Further, when those models were used in another domain, they struggle to generalize correctly. That is the idea that promotes the creation of GLUE, SQuAD V1.1/V2.0 and RACE to have benchmarks able to check the reliability of models on various tasks.

GLUE: The General Language Understanding Evaluation (GLUE) [23] is a collection of nine tasks created to test the generalization of modern NLP models. It reviews a wide range of NLP problems like Sentiment Analysis, Question Answering and inference tasks. Because of the rapid improvement of the state-of-the-art on GLUE, SuperGLUE [24] is a new proposed benchmark to check general language systems but with more complicated more laborious tasks.

SQuAD: Stanford Question Answering Dataset (SQuAD) V1.1 [25] is a benchmark designed to resolve Reading Comprehension (RC) challenges. There are more than 100,000+ questions in the data set. There is no proposed answer like in the other RD datasets. The task contains a document, and the model has to find the answer directly in the text passage. SQuAD v2.0 [26] is based on the same principle than the V1.1, but this time the answer is not necessarily in the questions.

RACE: Reading Comprehension From Examinations (RACE) [27] is a collection of English questions set aside to Chinese students from middle school up to high school. Each item is divided into two parts, a passage that the student must read and a set of 4 potential answers. Considering that the questions are intended to teenagers, it requires keen reasoning skills to answer correctly to most of the problems. The reasoning subjects present in RACE cover almost all human knowledge.

V. Transformers

The RNNs (LSTM, GRU) have a recurrent underlying structure and are, by definition recurrent. It is then hard to parallelize the learning process because of this fundamental property. To overcome this issue, [7] proposed a new architecture solely based on the attention layers; the Transformer. It has the advantage to catch the long-range dependencies of a sentence and to be parallelizable.

A. Transformer architecture

The Transformer is based on an encoder-decoder structure, where it takes a sequence $X = (x_1, ..., x_N)$ and produce a latent representation $Z = (z_1, ..., z_N)$. Due to the auto-regressive property of this model, the output sequence $Y_M = (y_1, ..., y_M)$ is produced one element at a time. i.e. the

word Y_M used the latent representation Z and the previously created sequence $Y_{M-1} = (y_1, ..., y_{M-1})$ to be generated. The Encoder and the Decoder are using the same Multi-Head Attention layer. A single Attention layer maps a query Q and keys K to a weighted sum of the values V. For technical reason there is a scaling factor $\frac{1}{\sqrt{d_k}}$.

$$Attention(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d_k}})V$$

B. Auto-Regressive Models

The auto-regressive models take the previous outputs to produce the next outcome. It has the particularity to be a unidirectional network; it can only reach the left context of the evaluated token. However, despite this flaw, it can learn accurate sentence representations. It relies on the regular Language Modeling (LM) task as an unsupervised pre-training objective:

$$L(X) = \sum_i \log P(x_i | x_{i-k}, ..., x_{i-1}; \Theta)$$

This LM function maximizes the likelihood of the conditional probability P. Where X is the input sequence, k is the context window, and Θ are the parameters of the Neural Network.

Various models are using this property coupled with the Transformer architecture to produce accurate Language Model languages (i.e. it determines the statistical distribution of the learned texts). The first auto-regressive model using the Transformer architecture is GPT [8]. It has a pre-training Language Modeling phase where it learns on raw texts. In the second learning phase, it uses supervised fine-tuning to adjust the network to the downstream tasks.

GPT-2 [14] uses the same pre-training principles than GPT. Though, this time it tries to achieve the same results in a zero-shot fashion (i.e. without fine-tuning the network to the downstream tasks). To accomplish that goal, it must capture the full complexity of textual data. To do so, it needs a wider system with more parameters. The results of this model are competitive to some other supervised tasks on a few subjects (e.g. reading comprehension) but are far from being usable on other jobs such as summarization.

Another auto-regressive network is XLNet [28]. It aims to use the strength of the language modeling of the auto-regressive model and at the same time, use the bidirectionality of BERT [9]. To do so, it relies on transformer-XL [29], the state-of-the-art model for the auto-regressive network.

C. BERT

GPT and GPT-2 use a unidirectional language model; they can only reach the left context of the evaluated token. That property can harm the overall performance of those models in reasoning or question answering tasks. Because, in those topics, both sides of the sentence are crucial to getting an optimal sentence-level understanding.

To counter this unidirectional constraint, [9] introduced the Bidirectional Encoder Representations from Transformers (BERT). This model can fuse the left and the right context of a sentence, providing a bidirectional representation and allow a better context extractor for reasoning tasks. The architecture of BERT is based on the Multi-Head Attention layers encoder like proposed in [7]. Originally [9] proposed two versions of BERT, the base version with 110M of parameters and the large version with 340M parameters.

Like GPT and GPT-2, BERT has an unsupervised pre-training phase where it learns its language representation. Nevertheless, due to its inherent bidirectional architecture, it cannot be trained using the standard Language Model objective. Indeed, the bidirectionality of BERT allows each word to see itself, and therefore it can trivially predict the next token. To overcome this issue and pre-train their model, [9] use two unsupervised objective tasks: the Masked Language Model (MLM) and the Next Sentence Prediction (NSP).

Once the pre-training phase is over, it remains to fine-tune the model to the downstream tasks. Thanks to BERT's Transformer architecture, the downstream can be straightforwardly done because the same structure is used for the pre-training and the fine-tuning. It merely needs to change the final layer to match the requirements of the downstream task.

VI. Post-BERT

Due to the high performance of BERT on 11 NLP tasks, a lot of researchers inspired by BERT's architecture applied it and tweaked it to their needs [30], [31].

A. BERT improvement

Further, studies have been done to improve the pre-training phase of BERT. The post-BERT model RoBERTa [16] proposes three simple modifications of the training procedure. **(I)** Based on their empirical results, [16] shows that BERT is undertrained. To alleviate this problem, they propose to increase the length of the pre-training phase. By learning longer, the outcomes are more accurate. **(II)** As the results of [32] and [14] demonstrate, the accuracy of the end-task performance relies on the wide variety of trained data. Therefore, BERT must be trained on larger datasets. **(III)** In order to improve the optimization of the model, they propose to increase the batch size. There are two advantages to have a bigger batch size; First, the large batch size is easier to parallelize, and second, it increases the perplexity of the MLM objective.

B. Model reduction

Since the Transformer's revolution, state-of-the-art networks have become bigger and bigger. Accordingly, to have a better language representation and better end-task results, the models must grow to catch the high complexity of texts. This expansion of the network's size has a high computational cost. More powerful GPUs and TPUs are required to train those large models. If we take, for example, the Nvidia's GPT-8B [1] with 8 billion parameters, it became infeasible for small tech companies or small labs to train a network as huge as that.

[1] https://nv-adlr.github.io/MegatronLM

It is then necessary to find smaller systems that maintain the high performances of the bigger ones.

Working with smaller models has multiple advantages. If the model size is shrunk, it trains faster, and the inference time will also be reduced. If it is small enough, it can be run on smartphones or IoT devices in real-time.

One technique introduced to reduce the size of those big networks is the knowledge distillation. It is a compression method that consists of a small network (student) trained to reproduce the behaviour of a bigger version of itself (teacher). The teacher is primarily trained as a regular network, and after that, it is distilled to reduce its size. DistilBERT [33] is a distilled version of BERT that reduces the number of layers by a factor of 2. It retains 97% of BERT on the GLUE benchmark while being 40% smaller and 60% faster at the inference time.

Another way to reduce the size of BERT is by changing the architecture itself. AlBERT [34] proposes two ideas to decrease the number of parameters. The first approach factorizes the embedding of the parameters. It separates the large vocabulary embedding matrix into two smaller matrices. The size of the hidden layer is separated from the size of the vocabulary representation. The second method is a cross-layer parameter sharing. This technique prevents the parameters from growing with the depth of the network. With those two tricks, it allows reducing the size of the large BERT version by 18% without a loss of performance. Since this architecture is smaller, the training time is also faster.

C. Multitask Learning

BERT learns several tasks sequentially and increases the overall performance of the downstream end-tasks. The main issue with the continual pre-training method is that it must learn efficiently and quickly newly introduced sub-tasks, and it must remember what has been learned previously. The Multi-task Learning (MTL) principle is based on human consideration. If you learn how to do a first task, then a second related task is going to be more accessible to master. There are two main trends in MTL.

The first one uses an MTL scheme during the fine-tuning phase. MT-DNN [35] based on the backbone of BERT is using the same pre-training procedure, but during the fine-tuning step, it uses four multi-tasks. Training on all the GLUE tasks at the same time makes it gain an efficient generalization ability.

On the opposite [10] proposes an MTL process directly during the pre-training step; ERNIE 2.0 introduces a continual pre-training framework. More specifically, it uses a Sequential Multi-task Learning where it begins to learn a first task. When this first task is mastered, a new task is introduced in the continual learning process. The previously optimized parameters are used to initiate the model, the new task and the previous tasks are trained concurrently. There are three groups of pre-training tasks, and each of them aims to capture a different level of semantic:

Word-Aware Tasks: It captures the lexical information of the text: the Knowledge Masking Task (i.e. it masks phrases and entities), the Capitalization prediction (i.e. it predicts if a word has a capitalized first letter), and the Token-Document Relation Prediction Task (i.e. it predicts if a token of a sentence belongs to a document where the sentence initially appears).

Structure-Aware Tasks: It learns the relationship between sentences: sentence reordering task (i.e. split and shuffle a sentence and must find the correct order), sentence distance task (i.e. it must find if two sentences are adjacent, belong to the same document or if they are entirely unrelated).

Semantic-Aware Tasks: It learns a higher order of knowledge: discourse relation task (i.e. it predicts the semantic or rhetorical relation of sentences), IR relevance task (i.e. find the relevance of information retrieval in texts).

D. Specific language models

In order to tackle specific languages problems, different monolingual versions of BERT were trained in different languages. For example BERTje [36] is a Dutch version, AlBERTo [37] is an Italian version, and CamemBERT [38] and FlauBERT [39] are two different models for French. These models outperform vanilla BERT in different NLP tasks specific to these languages.

E. Cross-language model

XLM [40] aims to build a universal cross-language sentence embedding. The goal is to align sentence representations to improve the translation between languages. To do so, a Transformer architecture with two unsupervised tasks and one supervised is used. The effectiveness of cross-language pre-training in order to improve the multilingual machine translation is shown.

VII. Going Further

Despite the excellent performances of the Transformer architecture, new layers aiming to improve the performance and the complexity have been released.

The Transformer uses a gradient-based optimization procedure. Thus, it needs to save the activation value of all the neurons to be used during the back-propagation. Because of the massive size of the Transformer models, the GPU/TPU's memory is rapidly saturated. The Reformer [41] counter the memory problem of the Transformer by recomputing the input of each layer during the back-propagation instead of storing the information. The Reformer can also reduce the number of operations during the forward pass by computing a hash function that pairs similar inputs together. Like that, it does not compute all pairs of vectors to find the related ones. Therefore, it increases the size of the text it can treat at once.

Another way to improve the architecture of a network is by using an evolving algorithm as proposed by [42]. To create a new architecture designed automatically, they evolve a population of Transformers based on their accuracy. Using the Progressive Dynamic Hurdles (PDH), they could reduce the search space and the training time. With this technique and an extensive amount of computational power (around 200 TPUs), they could find a new architecture that outperforms the previous one.

VIII. Conclusion

The Transformer-based networks have pushed the reasoning-skills to human-level abilities. It can even excel the human capabilities on a few tasks of GLUE. Transformer-based networks have changed the face of NLP tasks. They can go far beyond the results obtained with RNNs, and they can do it faster. They have helped solve many problems at the same time by providing a direct and efficient way to combine several downstream tasks. Nevertheless, much work remains before having a system with a human-level comprehension of the underlying meaning of texts, that is also sufficiently small to run on devices with low computational power.

References

[1] F. J. Och and H. Ney, "The Alignment Template Approach to Statistical Machine Translation," *Computational Linguistics*, vol. 30, pp. 417–449, Dec. 2004.

[2] A. M. Rush, S. Chopra, and J. Weston, "A Neural Attention Model for Abstractive Sentence Summarization," *arXiv:1509.00685 [cs]*, Sept. 2015. arXiv: 1509.00685.

[3] L. Yu, W. Zhang, J. Wang, and Y. Yu, "SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient," *arXiv:1609.05473 [cs]*, Aug. 2017. arXiv: 1609.05473.

[4] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, pp. 1735–1780, Nov. 1997.

[5] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," *arXiv:1406.1078 [cs, stat]*, Sept. 2014. arXiv: 1406.1078.

[6] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A Search Space Odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, pp. 2222–2232, Oct. 2017. arXiv: 1503.04069.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," *arXiv:1706.03762 [cs]*, Dec. 2017. arXiv: 1706.03762.

[8] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving Language Understanding by Generative Pre-Training," p. 12, Nov. 2018.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv:1810.04805 [cs]*, May 2019. arXiv: 1810.04805.

[10] Y. Sun, S. Wang, Y. Li, S. Feng, H. Tian, H. Wu, and H. Wang, "ERNIE 2.0: A Continual Pre-training Framework for Language Understanding," *arXiv:1907.12412 [cs]*, Nov. 2019. arXiv: 1907.12412.

[11] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *arXiv:1301.3781 [cs]*, Sept. 2013. arXiv: 1301.3781.

[12] J. Pennington, R. Socher, and C. Manning, "Glove: Global Vectors for Word Representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Doha, Qatar), pp. 1532–1543, Association for Computational Linguistics, 2014.

[13] R. Sennrich, B. Haddow, and A. Birch, "Neural Machine Translation of Rare Words with Subword Units," *arXiv:1508.07909 [cs]*, June 2016. arXiv: 1508.07909.

[14] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language Models are Unsupervised Multitask Learners," p. 24, Nov. 2019.

[15] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing," *arXiv:1808.06226 [cs]*, Aug. 2018. arXiv: 1808.06226.

[16] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," *arXiv:1907.11692 [cs]*, July 2019. arXiv: 1907.11692 version: 1.

[17] T. H. Trinh and Q. V. Le, "A Simple Method for Commonsense Reasoning," *arXiv:1806.02847 [cs]*, Sept. 2019. arXiv: 1806.02847.

[18] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, "Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books," *arXiv:1506.06724 [cs]*, June 2015. arXiv: 1506.06724.

[19] R. Parker, D. Graff, and J. Kong, "English gigaword," *Linguistic Data Consortium*, Jan. 2011.

[20] J. Callan, M. Hoy, C. Yoo, and L. Zhao, "The ClueWeb09 Dataset - Dataset Information and Sample Files," Jan. 2009.

[21] A. Gokaslan and V. Cohen, *OpenWebText Corpus*. Jan. 2019.

[22] R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, and Y. Choi, "Defending Against Neural Fake News," *arXiv:1905.12616 [cs]*, Oct. 2019. arXiv: 1905.12616.

[23] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding," *arXiv:1804.07461 [cs]*, Feb. 2019. arXiv: 1804.07461.

[24] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems," *arXiv:1905.00537 [cs]*, July 2019. arXiv: 1905.00537.

[25] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ Questions for Machine Comprehension of Text," *arXiv:1606.05250 [cs]*, Oct. 2016. arXiv: 1606.05250.

[26] P. Rajpurkar, R. Jia, and P. Liang, "Know What You Don't Know: Unanswerable Questions for SQuAD," *arXiv:1806.03822 [cs]*, June 2018. arXiv: 1806.03822.

[27] G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy, "RACE: Large-scale ReAding Comprehension Dataset From Examinations," *arXiv:1704.04683 [cs]*, Dec. 2017. arXiv: 1704.04683.

[28] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized Autoregressive Pretraining for Language Understanding," *arXiv:1906.08237 [cs]*, June 2019. arXiv: 1906.08237.

[29] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context," *arXiv:1901.02860 [cs, stat]*, June 2019. arXiv: 1901.02860.

[30] C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid, "VideoBERT: A Joint Model for Video and Language Representation Learning," *arXiv:1904.01766 [cs]*, Sept. 2019. arXiv: 1904.01766.

[31] A. Wang and K. Cho, "BERT has a Mouth, and It Must Speak: BERT as a Markov Random Field Language Model," *arXiv:1902.04094 [cs]*, Apr. 2019. arXiv: 1902.04094 version: 2.

[32] A. Baevski, S. Edunov, Y. Liu, L. Zettlemoyer, and M. Auli, "Cloze-driven Pretraining of Self-attention Networks," *arXiv:1903.07785 [cs]*, Mar. 2019. arXiv: 1903.07785.

[33] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," *arXiv:1910.01108 [cs]*, Oct. 2019. arXiv: 1910.01108.

[34] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations," *arXiv:1909.11942 [cs]*, Oct. 2019. arXiv: 1909.11942 version: 3.

[35] X. Liu, P. He, W. Chen, and J. Gao, "Multi-Task Deep Neural Networks for Natural Language Understanding," *arXiv:1901.11504 [cs]*, May 2019. arXiv: 1901.11504.

[36] W. de Vries, A. van Cranenburgh, A. Bisazza, T. Caselli, G. van Noord, and M. Nissim, "BERTje: A Dutch BERT Model," *arXiv:1912.09582 [cs]*, Dec. 2019. arXiv: 1912.09582.

[37] M. Polignano, P. Basile, and M. de Gemmis, "ALBERTO: Italian BERT Language Understanding Model for NLP Challenging Tasks Based on Tweets," p. 6, 2019.

[38] L. Martin, B. Muller, P. J. O. Suárez, Y. Dupont, L. Romary, E. V. de la Clergerie, D. Seddah, and B. Sagot, "CamemBERT: a Tasty French Language Model," *arXiv:1911.03894 [cs]*, May 2020. arXiv: 1911.03894.

[39] H. Le, L. Vial, J. Frej, V. Segonne, M. Coavoux, B. Lecouteux, A. Allauzen, B. Crabbé, L. Besacier, and D. Schwab, "FlauBERT: Unsupervised Language Model Pre-training for French," *arXiv:1912.05372 [cs]*, Mar. 2020. arXiv: 1912.05372.

[40] G. Lample and A. Conneau, "Cross-lingual Language Model Pretraining," *arXiv:1901.07291 [cs]*, Jan. 2019. arXiv: 1901.07291.

[41] N. Kitaev, L. Kaiser, and A. Levskaya, "Reformer: The Efficient Transformer," *arXiv:2001.04451 [cs, stat]*, Jan. 2020. arXiv: 2001.04451.

[42] D. R. So, C. Liang, and Q. V. Le, "The Evolved Transformer," *arXiv:1901.11117 [cs, stat]*, May 2019. arXiv: 1901.11117.