ARATRANS THE NEW TRANSFORMER MODEL TO GENERATE ARABIC TEXT

Abdelhamid Atassi, LARI Laboratory, Ibn Tofail University
Ikram El Azami, LARI Laboratory, Ibn Tofail University

ABSTRACT

The automated generation of coherent text is an area of natural language processing (NLP) that has received a lot of attention in recent years. Several state-of-the-art language models have been developed capable of automatically generating text structures of a quality close to that of human-generated text.

Indeed, many potential use cases are seemingly benign (i.e. automated summarizing of long texts, generation of sporting event recaps, generation of text for entertainment purposes, etc.), while other applications such as propaganda generation and fake news have been identified as real risks associated with this technology.

In this paper, we demonstrate a new architecture dedicated to the generation of the text essentially based on the basic architecture of neural network of the transformer type and we compare the results of our model with the basic model GPT-2 in English and GPT-2 in Arabic, to show that generating the automatic text of tweets from our AraTrans model gives better results compared to a powerful GPT-2 model or a GPT-2 model pre-trained on Arabic text.

Keywords: RNN, Convolutional Neural Network, CNN, Adam, AdamA, Kaggle, NLP, BERT, GPT, GPT-2, T5, Transformer, Transformer-XL, Word-based, WordPiece, Character-based, Bye-Pair Encoding, Nvidia, Nvidia Tesla, Huggface, Tockenization, Learning rate, Fine-tuning, Arabic, Wikipedia.

INTRODUCTION

A transformer is a type of neural network architecture that appeared in the article «Attention Is All You Need» (Vaswani et al., 2017), this model architecture consists of a multi-head self-attention mechanism combined with an encoder-decoder structure. This mechanism can achieve results that outperform various other models by exploiting the Recurrent Neural Networks (RNNs) or Convolutional Neural Networks (CNNs) in both evaluation score and training time.

Indeed, a key advantage of a Transformer over other Neural Network (NN) structures is that a more distant context around a word is considered more efficiently in terms of computation. The expression « make... more difficult » would be recognized in the sentence « make the registration or voting process more difficult » even if « more difficult » is a rather remote dependence on make. The calculation of the relevant context around a word can be done in parallel, saving important training resources.

Initially, the Transformer was developed for NLP tasks and applied by Vaswani et al. 2017 to machine translation. However, it has also been adapted for image recognition as well as other areas.

The structure of the Transformer model has largely replaced other implementations of the NLP model such as NNCs. Current NLP models use the Transformer architecture in part or as a whole.

The Generative Pre-Training Transformer (GPT) model (Radford et al., 2018) uses only the Transformer structure decoder (unidirectional), while BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019) is based on the Transformer (bidirectional) encoder.

Citation Information: Atassi, A., & Azami, I.E. (2022). Aratrans the new transformer model to generate arabic text. *Journal of Management Information and Decision Sciences, 25*(S2), 1-7.

T5 (Raffel et al., 2020), counts to it, uses a structure of transform encoder-decoder very similar to the original implementation. These general architectures also differ in the number and size of the elements that make up an encoder or a decoder (i.e. the number of layers, the hidden size and the number of self-attention heads they use).

Prepare Dataset for Training

Dataset

The initial training dataset used is in the form of 4GB of press articles dated 20.01.2018 downloaded from the kaggle platform, to do the first training, and fixed the length of the sentences to 1024 characters.

Fine-tuning (Yosinski et al., 2014), which consists in thawing the entire model (or part of it), and re-training it on new data with a very low learning rate. This can potentially make significant improvements, gradually adapting pre-trained features to new data.

In fact, the fine-tuning dataset contains 10K tweets from the kaggle platform as text divide on lines and each line corresponds to a tweet. Each line of text will serve as an individual text sequence when entering the template for training. Cleaned tweet data is saved in csv format which we will load for model learning.

Tokenization

Tokenization is the process of converting a string of characters or words into a series of subcomponents (tokens) that can be used to reconstruct the original string of characters or words. A simple example of a set of Tokens is the English alphabet. Each character represents a token, and these tokens can be organized significantly to form words. Similarly, the vocabulary of the English language represents a series of tokens that can be combined to form sentences.

In fact, there are three different tokenization techniques. Each of these techniques works differently and has its own advantages and disadvantages:

1) Word-based tokenization is the most commonly used tokenization technique. It divides a piece of text into words based on a delimiter. The most commonly used delimiter is space. Each word is then represented with an identifier and each identifier contains a lot of information, since a word in a sentence usually contains a lot of contextual and semantic information. The technique seems impressive but this type of tokenization leads to a massive corpus that leads to a big vocabulary. This huge vocabulary size leads to a huge integration matrix for input and output layers, making the model heavier and requiring more computational resources.
2) Character-based tokenization, character-based tokenizers divide plain text into individual characters. The logic behind this tokenization is that a language has many different words but also has a fixed number of characters. The result is a very limited vocabulary. This type of tokenization is quite simple and can greatly reduce the time and complexity of the memory. However, a character usually has no meaning or information as a word does.
3) Subword-based tokenization, another popular tokenization is subword-based tokenization, which is a solution between word- and character-based tokenization. The main idea is to solve the problems encountered by word-based tokenization (very large vocabulary size, large number of out-of-vocabulary tokens (OOV) and different meaning of very similar words) and character-based tokenization (very long sequences and less significant individual tokens).

Most of the models that have obtained cutting-edge results use a sort of sub word tokenization algorithm. Some common tokenization algorithms based on sub words are WordPiece (Ynghui Wu et al., 2016) used by BERT and DistilBERT, Unigram by XLNet and ALBERT, and Bye-Pair Encoding by GPT-2 and RoBERTa.

Citation Information: Atassi, A., & Azami, I.E. (2022). Aratrans the new transformer model to generate arabic text. *Journal of Management Information and Decision Sciences, 25*(S2), 1-7.

Sub-word tokenization allows the model to have a decent vocabulary size and also to be able to learn meaningful representations independent of context. It is even possible for a model to process a word he has never seen before because decomposition can lead to known sub words. See Figure 1.

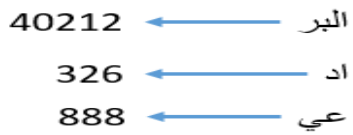

FIGURE 1
AN EXAMPLE OF A TOKEN AND THE CORRESPONDING ID

In our case, we use the Bye-Pair Encoding (Bostrom et al., 2020) used by GPT-2, because our model will be compared with a model of the GPT-2 type. See figure 2

Ids tokens : [40212, 326, 888, 929, 279, 706, 9291, 6946, 1251, 2051, 295]

Tokens : البرادعي يستقوى بامريكا مرة اخرى

FIGURE 2
APPLY TOKENIZATION TO A SENTENCE

With the use of the tokenizer of the GPT-2 model, some special tokens are added. These will be used to represent the beginning (<BOS>) and end (<EOS>) of each text sequence. An additional fill token (<PAD>) is created so that all entries can be filled to the same length for model learning. See figure 3.

Ids tokens : [50257, 40212, 326, 888, 929, 279, 706, 9291, 6946, 1251, 2051, 295]

Tokens : <BOS>البرادعي يستقوى بامريكا مرة اخرى

FIGURE 3
THE ADDITION OF THE <BOS> TOKEN AT THE BEGINNING OF THE SENTENCE

The MAX_LEN constant refers to the maximum acceptable length of each sample of text supplied to the model for learning. Text samples containing less than this value will be filled to meet this length, while samples exceeding this value will be truncated. Selecting this value is critical because some pre-trained text templates can only process text sequences up to a specific length. In our case this length was set to 1024 during initial training and 200 during Fine tuning, in order to give the model a wider learning during the initial learning phase, and because the old max size of the characters used by twitter was 140 characters and which has now become 280 [9] characters following the recent update. However, according to statistics, the use of 280 characters does not exceed 1%.

Batch

After organizing a dataset in form of a pair containing the tokens indexes and the corresponding mask of each tweet, see FIG. 4, as well as dividing the dataset into two parts, 80% of training data and 20% of validation data, the choice of batch data in a random manner with a length of 2 at each epoch, the number of epochs was fixed at 200.

Citation Information: Atassi, A., & Azami, I.E. (2022). Aratrans the new transformer model to generate arabic text. *Journal of Management Information and Decision Sciences, 25*(S2), 1-7.

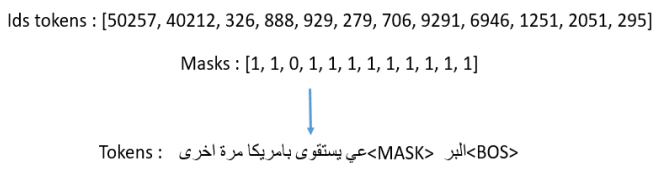

FIGURE 4
SAMPLE BATCH, CONTAINS THE IDS AND MASKS CORRESPONDING TO IT

Since our validation data set will only be used to evaluate model performance, we do not need to worry about randomizing data when creating batches. Therefore, we will use a sequential validation batch. This sampling simply creates batches based on the input sequence rather than applying randomization.

Architecture

Transformers have been introduced in the context of machine translation in order to avoid recursion and allowing parallel computation to reduce learning time and also to reduce the performance decrease due to long dependencies. The main features are:

- Non sequential: sentences are processed as a whole rather than word by word.
- Attention: this is the new « unit » used to calculate similarity scores between words in a sentence.
- Positional encoding: another innovation introduced to replace recurrence. The idea is to use fixed or learned weights that encode information about a specific position of a token in a sentence.

Thus, the first point is the main reason why transformers do not suffer from prolonged dependency problems. Original transformers do not rely on past hidden states to capture dependencies with previous words, they process a sentence as a whole, so there is no risk of losing (or « forgetting ») past information. In addition, the multi-head attention and the positional encodings both provide information about the relationship between the different words.

Most models that rely on transforming on the market take the encoder or decoder part with slight modifications, for example GPT-2 relies on decoder on the other hand BERT relies on encoder but most models are tested on the translation task and most searches do the tests on the translation task or question answer, except for the Transformer-XL model (Dai et al., 2019) that uses the basic architecture of transforming with small modifications (i.e. uses both parts to encode and decode it).

In our case, we have kept the same structure but with modifications at the level of the two inputs of the model, the first input is fed by a sequence of tokens but with some masked tokens and the second with the same sequence of tokens but without mask. Six layers in the encoding part and six layers in the decoding part. See figure 5.

Citation Information: Atassi, A., & Azami, I.E. (2022). Aratrans the new transformer model to generate arabic text. *Journal of Management Information and Decision Sciences, 25*(S2), 1-7.

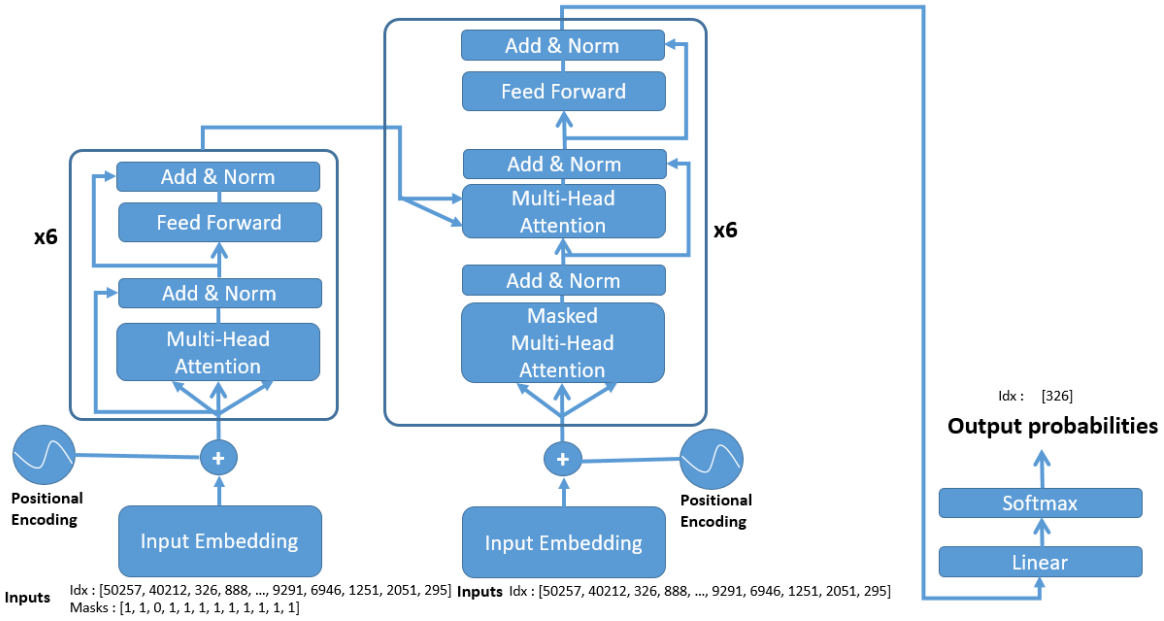

FIGURE 5
THE TRANSFORMATIVE ARCHITECTURE USED TO DRIVE OUR ARATRANS MODEL

We will use the AdamW optimizer (Loshchilov et al., 2019) to update model weights during training. AdamW is a modified version of Adam, a very popular adaptive gradient descent algorithm used for machine learning model formation. It is designed to decouple weight decay from model weight gradient updates. This improves the regularization of the optimizer and increases the ability of the trained model to generalize.

Learning rate used by the optimizer establishes the amplitude according to which the weights of the model are updated during each pass in the gradient descent algorithm is set to 5e-4.

Research has shown that reducing the learning rate gradually during the training process can improve overall model performance and reduce training time. There are several methods that can be used to reduce the learning rate, but we will use a linear step-by-step method. To do this, we will implement a linear learning rate planner. This type of learning rate scheduler starts with a learning rate of 0 and increases it linearly until it reaches a user-defined learning rate after a number of training steps known as a warm-up period. We need to specify this preheat period when setting our optimizer. Once the warm-up period has elapsed, the learning rate scheduler will decrease the learning rate linearly over the remaining training steps until the value reaches 0.

Finally, we will need to create a start of text that can be used to generate text using our trained template. We will use our start sequence token (<BOS>) as the start for all generated text.

RESULTS AND EVALUATION

With an i9, 32 GB machine we used a Nvidia Tesla K80 24GB GDDR.5 CUDA Cores 4992 card, we managed to get a well-trained general model and with a fine-tuning of a tweet dataset we get a model specialized in the generation of Arabic tweets, the two comparison models exist on the Huggface platform, the first is the GPT2-Small-Arabic model is a fine-tuning on the Arabic Wikipedia dataset which is a basic model gpt2 -small, the second is a gpt2-small model.

Both models give the same results even with a fine-tuning model GPT2-Small-Arabic.

Citation Information: Atassi, A., & Azami, I.E. (2022). Aratrans the new transformer model to generate arabic text. *Journal of Management Information and Decision Sciences, 25*(S2), 1-7.

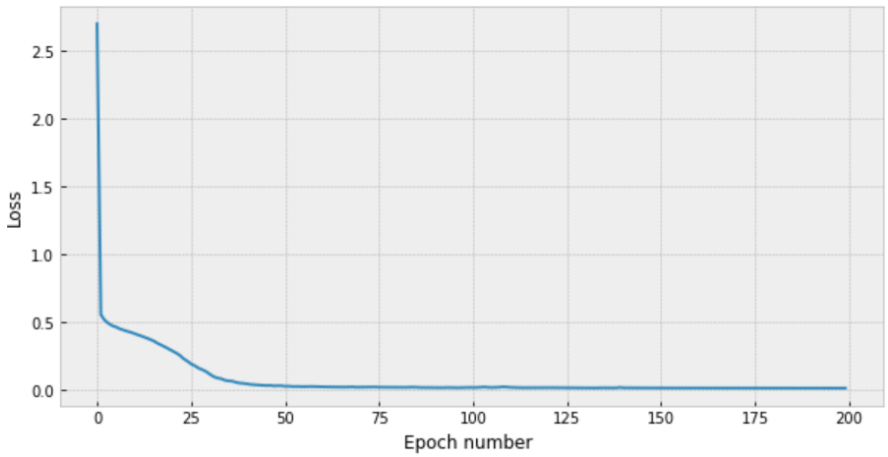

FIGURE 6
THE MODELS GPT2-SMALL-ARABIC AND GPT2-SMALL HAVE THE SAME RESULTS, BOTH MODELS DURING TRAINING, SHOW THAT THEY ARRIVE AT ZERO FROM THE 50TH EPOCH

With our 25-inch model, we are almost at zero loss, see FIG. 7, but with the other two 50-inch models, we are almost at zero. See figure 6.

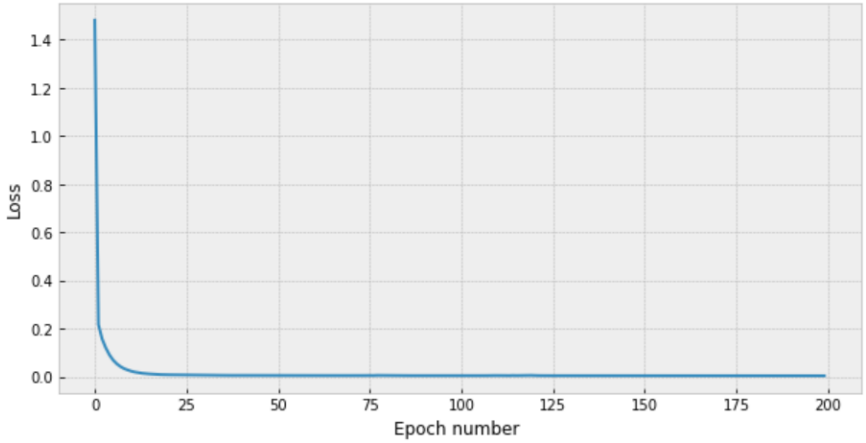

FIGURE 7
THE RESULTS OF OUR ARATRANS MODEL, DURING TRAINING, SHOW THAT THE MODEL REACHES ZERO FROM THE 25TH EPOCH

There is semantics when generating tweets from the AraTrans template. See figure 8

1 وكامل الاحترام لـ #عباس اللي مسك لسانه وما شتمش
2 الهتافات تتصاعد في المحلة ضد الإعلان الدستوري والرئيس مرسي
3 لا ده بيقوللك وسوري كمان وأعلى حاجة أن بيقوللك عبد البيادة وفي نفس الجملة بيهين الجيش

FIGURE 8
SOME EXAMPLES WHEN GENERATING TWEETS FROM THE ARATRANS TEMPLATE

CONCLUSION

Storage is the essential phase in the training of a model, and the choice of sub-words makes it possible to preserve a significant representation of a word in a sentence, the mask of words in a random way gives the model the possibility to be trained in a more optimal way and

Citation Information: Atassi, A., & Azami, I.E. (2022). Aratrans the new transformer model to generate arabic text. *Journal of Management Information and Decision Sciences, 25*(S2), 1-7.

to construct a model based on the architecture of the transform (the encoder and the decoder are used), we take advantage of this architecture also the parallelism which is a very important point, considering the number of data that the model must learn, we note that a model based on data in English even if in fine-tuning with an Arabic database, it takes the double epochs to train to a specific dataset, and this is not the case if it has a basic dataset in Arabic. The other works are based on the generation of structure and not meaning, through the training of poetry of the well-known poets. In this work, however, tweets were used to generate meaningful text.

Training with a basic dataset and with a basic architecture to transform requires huge resources and a lot of time, our future research focuses on optimizing the architecture to optimize resources and time and also do other training to optimize the meaning of Arabic text to be close to the language of a human being.

REFERENCES

Vaswani, A. (2017). Attention is all you need.
Radford. (2018). Generative pre-trained transformer.
Devlin, J. (2019). Pre-training of deep bidirectional transformers for language understanding.
Raffel, C. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer.
kaggle.com
Yosinski, J. (2014). How transferable are features in deep neural networks?
Wu, Y. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation.
Bostrom, K. (2020). Byte pair encoding is suboptimal for language model pretraining.
Dai, Z. (2019). Transformer-XL: Attentive language models beyond a fixed-length context. Crossref, Google scholar, Indexed at
Loshchilov, I. (2019). Decoupled weight decay regularization.
https://huggingface.co/akhooli/gpt2-small-arabic
https://www.kaggle.com/abedkhooli/arabic-wiki-data-dump-2018
https://huggingface.co/transformers/model_doc/gpt2.html

Received: 30-Dec-2021, **Manuscript No.** JMIDS- 21-8873; **Editor assigned:** 02-Jan-2022, **PreQC No.** JMIDS- 21-8873 (PQ); **Reviewed:** 15-Jan-2022, **QC No.** JMIDS- 21-8873; **Revised:** 23-Jan-2022, **Manuscript No.** JMIDS- 21-8873(R); **Published:** 30-Jan-2022

Citation Information: Atassi, A., & Azami, I.E. (2022). Aratrans the new transformer model to generate arabic text. *Journal of Management Information and Decision Sciences, 25*(S2), 1-7.