**<https://dsstream.com/roberta-vs-bert-exploring-the-evolution-of-transformer-models/>**

[**https://habr.com/ru/articles/680986/**](https://habr.com/ru/articles/680986/)

**RoBERTa**

*RoBERTa — это простая, но очень популярная альтернатива/преемник BERT. Она улучшает BERT за счет тщательной и разумной оптимизации обучающих гиперпараметров для BERT. Несколько простых и понятных изменений в совокупности повышают производительность RoBERTa и позволяют ей превзойти BERT практически во всех задачах, для которых он был разработан.*

*Самое интересное, что во время публикации Роберты другой популярный новый трансформер,* [*XLNet*](https://towardsdatascience.com/how-to-use-xlnet-from-the-hugging-face-transformer-library-ddd0b7c8d0b9)*, также был представлен в исследовательской работе. Однако изменения, внесенные в XLNet, реализовать значительно сложнее, чем в RoBERTa, и это только увеличивает популярность последней среди сообщества AI/NLP.*

*Как я уже упоминал, RoBERTa использует ту же архитектуру, что и BERT. Однако, в отличие от BERT, во время предобучения она обучается только генерации пропущенного токена (BERT также предобучался предсказанию следующего предложения). Ниже приведены некоторые изменения в гиперпараметрах, благодаря которым RoBERTa достигла производительности.*

* *Более длительное время обучения и больший объем обучающих данных (в 10 раз больше — от 16GB к 160GB)*
* *Размер батча от 256 к 8000 и больший словарь — от 30k to 50k*
* *В качестве входных данных используются более длинные последовательности, но RoBERTa по-прежнему имеет ограничение на максимальное количество токенов — 512, как и у BERT*
* *Динамическое маскирование позволяет маскирующей схеме меняться при каждой подаче последовательности на модель. Отличие от BERT в том, что везде используется одна и та же маскирующая схема.*

*Знание того, как использовать BERT из библиотеки Hugging Face помогает понять, как закодирована RoBERTa (и другие модели из этой статьи).* [*Здесь*](https://towardsdatascience.com/how-to-use-bert-from-the-hugging-face-transformer-library-d373a22b0209) *вы можете узнать об этом больше. Если следовать коду из этого текста, использовать RoBERTa из Hugging Face довольно просто.*

from transformers import RobertaModel

import torch

import torch.nn as nn

class RoBERTa\_Model(nn.Module):

 def init(self, classes):

   super(RoBERTa\_Model, self).init()

   self.roberta = RobertaModel.from\_pretrained('roberta-base')

   self.out = nn.Linear(self.roberta.config.hidden\_size, classes)

   self.sigmoid = nn.Sigmoid()

 def forward(self, input, attention\_mask):

   , output = self.roberta(input, attention\_mask = attention\_mask)

   out = self.sigmoid(self.out(output))

   return out

Приведенный выше код показывает, как можно построить модель RoBERTa Pytorch общего назначения. Если сравнить его с кодом для модели основанной на BERT, то увидите, что мы буквально заменяем BERT на RoBERTa! В этом есть смысл, ведь RoBERTa — это практически BERT, но лучше обученный.

Вскоре вы поймете, что это относится и к ALBERT с DistilBERT, ведь эти модели — модификации BERT. Код «обнимающего лица» работает так, что для использования любой модели, достаточно взять код BERT из вышеприведенной статьи и заменить все термины BERT на термины RoBERTa. То есть, мы импортируем модель RoBERTa, используем правильный идентификатор модели 'RoBERTa-base' и импортируем правильный токенизатор RoBERTa.

Поэтому, если вы хотите проводить моделирование языка по маске, создавать экстрактивную вопросно-ответную систему или что-то еще с помощью RoBERTa, вы можете использовать данные в статье фрагменты кода просто заменить термины BERT на RoBERTa, DistilBERT или ALBERT (в зависимости от того, что вы хотите использовать).