Charformer: Fast Character Transformers via Gradient-based Subword Tokenization

Орлов Александр
Солодуха Мария
Княжевский Владимир

Charformer

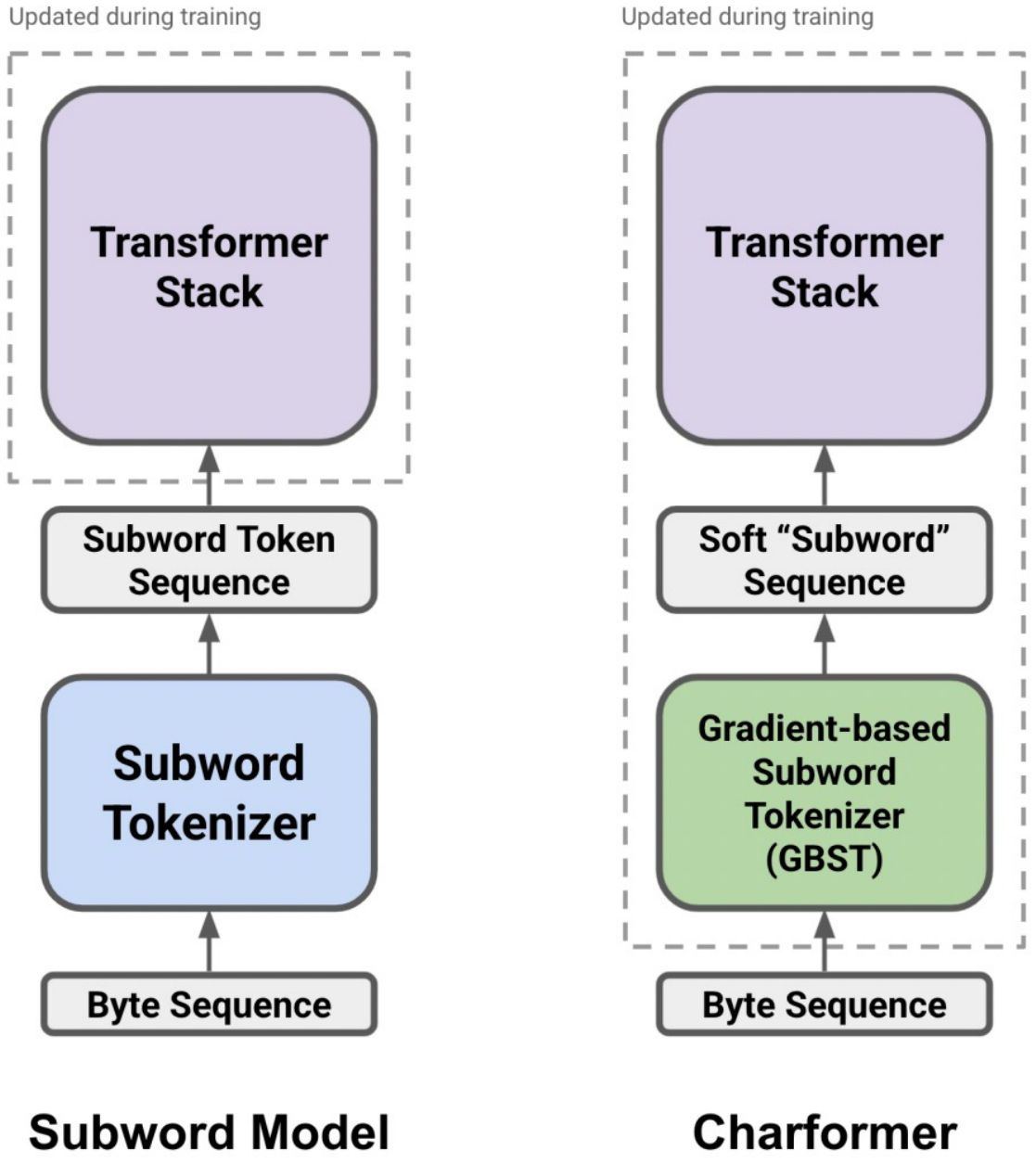
- Трансформер с токенайзером на основе градиентов
- Пайплайн обычного токенайзера
  1. Нормализация
  2. Пре-токенизация
  3. Токенизация (предобученная модель)
  4. Постпроцессинг (токены)

Проблемы обычных токенайзеров

- Разделение на токены по частоте, без учета лексики
- Некоторые языки не имеют разделения на слова
- Проблема мультиязычных моделей – разные языки имеют разное количество токенов
- Проблема с файнтюнингом – распределение слов на обучении может отличаться

Charformer

- Charformer пробует решить эти проблемы
- Работает с байтами (символами)
- Разделяет слова с помощью блоков подслов-кандидатов
- Выучивает интерпретируемые подслова
- Вычислительно эффективен



Gradient-based subword tokenizer (GBST)

1. Текст
2. Байты
3. Эмбеддинги (фиксированный словарь размера 256)
4. Матрица $X \in \mathbb{R}^{L \times d}$, где L - кол-во байтов, d - размерность эмбеддинга

Основная идея: научить модель выполнять сегментацию данных по подсловам, выбирая по ней наиболее подходящий для всех символов блок подслов

Gradient-based subword tokenizer (GBST)

- Блок подслов
- Функция проекция F (Avg Pooling)
- Объединяем все блоки подслов
- Авторы предлагают брать s равным b
- Считаем X_b для различных b

$$X_{i:i+b} \in \mathbb{R}^{b \times d}$$

$$F : \mathbb{R}^{b \times d} \rightarrow \mathbb{R}^{d}$$

$$X_b = [F(X_{i:i+b}); F(X_{(i+s):(i+s)+b}); \ldots]$$

Проблемы

1. В таком пайплайне обучения мы рассматриваем не все возможные подслова
   - Можно уменьшить s, но это заметно увеличит объем вычислений
   - Вместо этого авторы предлагают применить 1D свертку к матрице X
2. Мы не учитываем порядок символов
   - При применении свертки мы бонусом получаем учет порядка в функции F

Получение представлений

- Пусть $X_{b,i}$ - блок длины b, в который попадает i-ый символ

Тогда введем модель, которая будет оценивать этот блок.	$F_R : \mathbb{R}^d \to \mathbb{R}$
С помощью неё можем оценивать все наши блоки.	$p_{b,i} = F_R(X_{b,i})$
Отсюда можем получить матрицу вероятностей	$P_i = \mathrm{softmax}(p_{1,i}, p_{2,i}, ..., p_{M,i}),$ $P \in \mathbb{R}^{L \times M}$
Также добавим self-attention блок, чтобы учесть вероятности соседних подслов	$\hat{P} = \mathrm{softmax}(PP^T)P$
Наконец получаем новое представление входных данных	$\hat{X} = \sum_b^M \hat{P}_{b,i} X_{b,i}$

Усечение

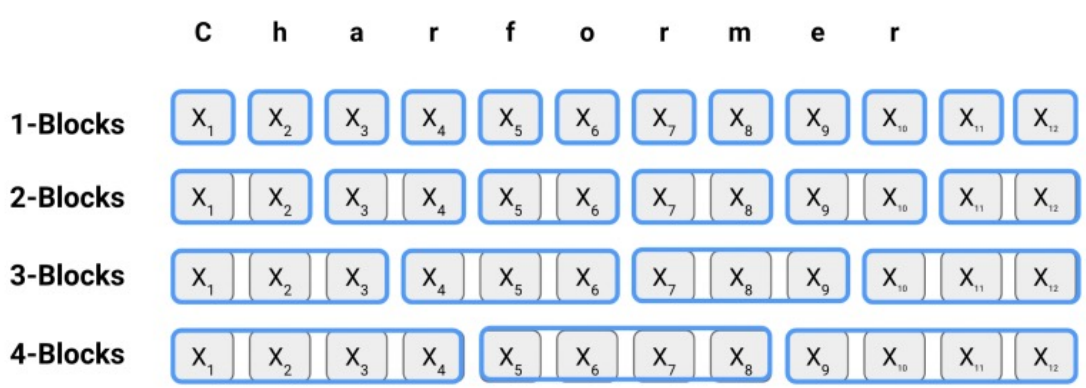
- С помощью Avg Pooling получаем векторные представления блоков символов

$$F_D : \mathbb{R}^{L \times d} \rightarrow \mathbb{R}^{\frac{L}{d_s} \times d}$$

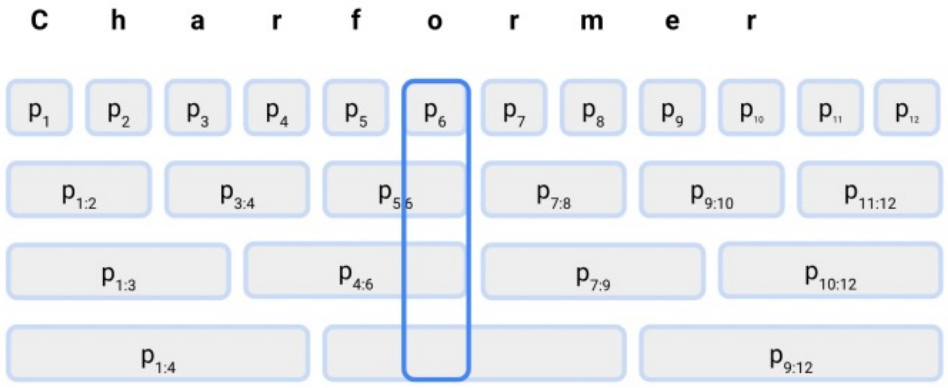
- В итоге на вход трансформеру мы будем подавать:

$$F_D(\hat{X}),\ \hat{X} = [\hat{X}_1, ..., \hat{X}_L]$$

Пример



(a) Formation of subword blocks to be scored by F_R. Offsets and/or pre-GBST convolutions not shown.



(b) Block scores that have been expanded back to length L. Softmax is taken over block scores at each position i to form block weights for constructing latent subword representations.

Figure 2: Illustration of subword block formation and scoring.

Пример

- Далее получившиеся на предыдущей картинке вероятности передаем по построенному пайплайну

$$P_i = \mathrm{softmax}(p_{1,i}, p_{2,i}, \ldots, p_{M,i})$$

$$\hat{P} = \mathrm{softmax}(PP^T)P$$

$$\hat{X} = \sum_{b}^{M} \hat{P}_{b,i} X_{b,i}$$

Выход GBST

$$F_D(\hat{X})$$

Эксперименты

Стандартные английские датасеты

Model	$\|\theta\|$	SST-2	MNLI	QNLI	MRPC	QQP	STSB	COLA	AVG
$\text{BERT}_{Base,Subword}$	110M	92.7	84.4/-	88.4	86.7/-	-	-	-	-
$\text{T5}_{Base,Subword}$	220M	92.7	84.2/84.6	90.5	88.9/92.1	91.6/88.7	88.0	53.8	84.3
Byte-level T5_{Base}	200M	91.6	82.5/82.7	88.7	87.3/91.0	90.9/87.7	84.3	45.1	81.5
Byte-level T5+Conv_{Base}	205M	89.8	81.1/82.5	89.2	83.6/89.2	90.7/87.7	85.0	47.1	81.2
Byte-level T5+LASC_{Base}	205M	90.0	80.0/80.8	87.1	82.8/88.1	89.0/85.4	83.7	25.3	77.0
CHARFORMER_{Base}	203M	91.6	82.6/82.7	89.0	87.3/91.1	91.2/88.1	85.3	42.6	81.4
Byte-level T5_{SBase}	133M	91.2	83.9/83.7	90.9	85.5/89.2	91.1/88.1	85.7	49.3	82.6
$\text{CHARFORMER}_{SBase}$	134M	91.5	83.7/84.4	91.0	87.5/91.4	91.4/88.5	87.3	51.8	83.6

Эксперименты

Классификация комментариев с Wiki

Model	Civil Comments	Wiki Comments
$\text{T5}_{Base,Subword}$	81.2 / -	91.5 / -
Byte-level T5_{Base}	82.8 / 78.7	93.2 / 75.4
Byte-level T5+LASC_{Base}	82.9 / 78.2	93.0 / 75.0
CHARFORMER_{Base}	83.0 / 78.8	92.7 / 79.7
$\text{CHARFORMER}_{SBase}$	83.0 / 78.9	93.5 / 75.5

Эксперименты

Классификация больших документов

Model	IMDb	News
$\text{T5}_{Base,Subword}$	94.2	93.5
Byte-level T5_{Base}	<u>91.5</u>	93.6
Byte-level T5+LASC_{Base}	91.1	93.5
CHARFORMER_{Base}	<u>91.5</u>	<u>94.0</u>
$\text{CHARFORMER}_{SBase}$	94.4	94.1

Эксперименты

Многоязычный перевод текстов

		In-Language	Translate-Train-All				Zero-Shot	
Model	$\|\theta\|$	TyDiQA-GoldP	XQuAD	MLQA	XNLI	PAWS-X	XNLI	PAWS-X
mBERT$_{Base}$ (Subword)	179M	77.6/68.0	-/-	-/-	-	-	65.4	81.9
mT5$_{Base}$ (Subword)	582M	80.8/70.0	75.3/59.7	67.6/48.5	75.9	89.3	75.4	86.4
Byte-level T5$_{Base}$	200M	75.6/65.4	68.6/54.3	61.8/44.4	69.4	87.1	57.4	80.9
Byte-level T5+LASC$_{Base}$	205M	70.6/59.7	66.8/52.1	58.8/41.1	67.9	84.8	55.2	79.0
CHARFORMER$_{Base}$	203M	75.9/65.6	70.2/55.9	62.6/44.9	71.1	87.2	57.6	81.6
CHARFORMER$_{SBase}$	134M	79.1/68.8	73.6/59.0	66.3/48.5	72.2	88.2	66.6	85.2
CHARFORMER$_{SBase,LongPT}$	134M	81.2/71.3	74.2/59.8	67.2/49.4	72.8	88.6	67.8	83.7

Производительность

Модели на основе подслов

Model	Batch Size	L	d_s	$\|\theta\|$	Speed (steps/s)	FLOPS
mT5$_{Base}$ (Subword)	1024	1024	-	582M	1.54	1.3×10^{15}
CHARFORMER$_{SBase}$	1024	2048	2	134M	1.98	4.3×10^{14}
CHARFORMER$_{SBase,LongPT}$	2048	2048	2	134M	1.01	4.3×10^{14}

Производительность

Модели на основе байтов

Model	L	d_s	$\|\theta\|$	Speed (steps/s)	FLOPS	Peak Mem.
T5_{Base} (Subword)	512	-	220M	9.3	1.1×10^{13}	-
Byte-level T5_{Base}	1024	1	200M	8.2	2.9×10^{13}	3.09GB
Byte-level T5+LASC_{Base}	1024	4	205M	15	9.9×10^{12}	1.62GB
CHARFORMER_{Base}	1024	2	206M	11	1.6×10^{13}	1.95GB
CHARFORMER_{Base}	1024	3	203M	15	1.1×10^{13}	1.63GB
$\text{CHARFORMER}_{SBase}$	1024	2	134M	14	1.3×10^{13}	1.73GB
$\text{CHARFORMER}_{SBase}$	1024	3	134M	20	8.7×10^{12}	1.34GB

Спасибо за внимание!