ML challenge Yandex Cup 2021 (CV)

14 окт 2021, 17:16:45
старт: 20 сен 2021, 10:00:00
финиш: 19 окт 2021, 16:00:00
до финиша: 4д. 22ч.
начало: 20 сен 2021, 10:00:00
конец: 19 окт 2021, 16:00:00
длительность: 29д. 6ч.

A. CV

Введение

Один из ярких современных трендов в машинном обучении — использование больших объемов данных, собранных в автоматическом режиме без ручной разметки. Например, модели, основанные на трансформерах, совершили переворот в сфере NLP за счет предобучения на огромных корпусах текстовых данных. Предобученные модели, такие как BERT и GPT, повсеместно используются для эффективного дообучения на самых разных целевых задачах. Идею обучения таких «универсальных» моделей активно продолжают развивать и в других доменах, в том числе в домене изображений, видео, и для мультимодальных данных.

В этой задаче предлагаем вам обучить такую универсальную модель, которая способна решать задачу классификации изображений в режиме zero-shot. Например, по одному только названию класса (без дообучения) модель должна уметь отличать изображения варежек от изображений перчаток, фотографии Москвы от фотографий Казани, фотографии Яндекс.Станции Макс от фотографий Яндекс.Станции Мини. Модель будет оцениваться по точности классификации на нескольких, не известных заранее наборах классов, доступ к названиям которых будет дан лишь во время автоматической оценки на приватном наборе данных. Участники должны будут загрузить код своего решения в тестирующую систему, где он будет запущен в стандартном изолированном окружении.

На каких же данных обучать такую модель? К сожалению, пока объем открытых данных в этой области оставляет желать лучшего (например, датасеты MS-COCO и Visual Genome содержат всего порядка 100 тысяч размеченных пар «изображение + текст»), а наборы пар «изображение + заголовок», которые можно собрать скрапингом интернета, часто получаются шумными и малоинформативными. Поэтому в рамках CV-трека ML-чемпионата мы выложим в открытый доступ многомиллионный набор мультимодальных данных (релевантных пар «текстовый запрос + изображение»), собранных по данным логов Яндекс.Картинок. Пользователи, как правило, кликают на самые релевантные изображения из поисковой выдачи — это обеспечивает дополнительную фильтрацию нерелевантных запросу изображений и уменьшает шум в данных. Ваша задача — извлечь из этих данных максимально полезный сигнал и обучить модель, которая продемонстрирует максимальную обобщающую способность.

Задача

Обучающие данные

В этой задаче мы предлагаем вам обучить мультимодальную модель на парах `[текст поискового запроса, релевантное запросу изображение]`, полученных сопоставлением наиболее релевантных изображений и пользовательских запросов по реальным кликовым данным. Отметим, что предлагаемый датасет сильно отличается от "Image Captioning" датасетов (таких как MS COCO) спецификой сбора данных: тексты представляют из себя не подписи к изображениям, а реальные поисковые запросы пользователей, что определяет специфику данных.

В датасете представлено более 5 миллионов уникальных изображений и более 20 миллионов уникальных пар `[текст, изображение]`.

Оценка качества

Обученная вами модель будет тестироваться на задаче zero-shot классификации изображений, то есть вам необходимо обучить модель, способную классифицировать изображения, имея в распоряжении лишь список с русскоязычными названиями классов.

Все датасеты, на которых производится оценка качества, разбиты на два набора, "публичный", и "приватный". Во время проведения соревнования лидерборд будет строиться по оценкам на публичном наборе данных, после завершения чемпионата все решения будут автоматически перетестированы на приватном наборе данных. Публичный зачет является **иллюстративным** и может быть использован для предварительной оценки качества решения и работоспособности модели в системе Яндекс.Контест. Окончательная оценка будет определяться качеством модели на приватном наборе данных. В качестве окончательного решения, которое будет оценено на приватном наборе данных, будет использоваться **последняя успешная посылка** (со статусом "OK"). **Будьте внимательны!** Если ваша лучшая посылка не будет последней, она не будет использована для финальной оценки!

Важно, что **все** тестовые данные (и публичные, и приватные) будут доступны только во время запуска кода участников в изолированном окружении системы Яндекс.Контест, названия классов в этих наборах данных опубликованы **не будут**.

Формат данных

Train

Исходные данные для обучения представлены в виде файла `metadata.json` (в формате json-lines) следующей структуры:

```
{"image": 1, "queries": ["запрос1", "запрос2", "запрос3"]}
{"image": 2, "queries": ["запрос1", "запрос2", "запрос3"]}
...
```

В отдельном файле `images.json` даны ссылки на исходные изображения:

```
{"image": 1, "url": "http://path.to/image1.jpg"}
{"image": 2, "url": "http://path.to/image2.jpg"}
...
```

Доступность всех изображений по предоставленным ссылкам не гарантируется. **Все права на изображения принадлежат их правообладателям. Распространение, коммерческое и личное использование данных вне соревнования недопустимо.**

Скачать данные можно по ссылкам: images.json metadata.json

Eval

Данные, на которых замеряется качество zero-shot классификации, представляют из себя несколько датасетов с изображениями. Каждый датасет расположен в **отдельной директории** с содержимым вида:

```
.
├── classes.json
└── img
    ├── 01204c5c-bdcd-4535-b981-318d12d16b40.jpg
    ├── 0135b8ce-1f9e-485a-81a1-82c302d44128.jpg
    ├── 0168b6e5-530b-4fde-801a-56f5c2d0762c.jpg
    ├── 01975fb0-ab0c-4b67-b159-e704d52b7660.jpg
...
```

В файле `classes.json` дан список русскоязычных названий классов (нумерация с 0). Для каждого из файлов директории `img` необходимо предсказать один (и только один) из классов. В качестве итоговой оценки используется средняя точность (precision) по всему набору датасетов, умноженная на 100. Код замера оценки можно найти в репозитории с бейзлайн-решением.

Для локальной оценки качества и рабоспособности модели можно использовать открытые датасеты, такие как CIFAR, ImageNet, Caltech. Небольшая подвыборка датасетов `Caltech101` и `Caltech256`, на которой можно локально провалидировать свое решение, доступна прямо в репозитории.

Формат отправки решения

Для отправки в тестирующую систему необходимо подготовить архив, в **корневой** директории которого присутсвуют файлы `setup.sh` и `predict.sh`. Первый файл `setup.sh` будет вызван без аргументов и может быть использован для настройки окружения, установки дополнительных пакетов (которые необходимо добавить в архив). Второй файл `predict.sh` будет вызван с двумя аргументами: первый — путь к директории с датасетами, второй — путь к json-файлу, в который необходимо вывести предсказания модели на всех датасетах. Пример формата выходного файла можно найти здесь. Библиотеки, доступные по-умолчанию, описаны в docker-файле. Полный пример решения, готового к отправке, можно найти в репозитории.

Baseline-решение

Baseline-решение представляет из себя классическую двухбашенную мультимодальную модель, обученную отличать релевантные пары "текст+изображение" от нерелевантных (contrastive target). В качестве энкодера изображений используется `resnet50` (претренированный на imagenet), в качестве энкодера текстов используется модель Bag-of-Words. Код обучения, предобученные веса, скрипты для запуска предсказания классов на наборе данных и оценки качества, а также более подробное техническое описание доступны в репозитории с baseline-решением.

Ограничения в системе Яндекс.Контест

1. Отправляемые архивы не должны иметь объем, превышающий 700 Мб.
2. Код инференса модели должен уметь использовать **несколько CPU-ядер** (при наличии), в противном случае имеется риск выхода за максимальное время при тестировании на приватных данных. Код, использующий для инференса распространенные фреймворки, такие как PyTorch и Tensorflow, использует несколько CPU-ядер по умолчанию.
3. Временное ограничение на 1 посылку — 23 минуты. При тестировании на **публичном** наборе данных используется виртуальное окружение с 1 vCPU, 8 GB RAM. Для оценки: в текущем окружении Яндекс.Контеста PyTorch-модель ResNet50 (бейзлайн) работает со скоростью 3 изображения в секунду, что при общем количестве изображений в публичном наборе ~2200 дает время работы 12.5 минут.
4. В изолированном окружении нет доступа к сети. Любые дополнительные пакеты необходимо добавлять в архив с вашей моделью.
5. Изначальное ограничение числа посылок равно 5. Далее каждый день лимит посылок будет увеличиваться на 5. Таким образом, за время всего соревнования можно будет отправить суммарно не более 150 посылок.
6. Посылки можно отправлять не чаще, чем 1 раз в 10 минут.

Объем тестовых данных

В публичной части тестовых данных используется несколько датасетов суммарным объемом 2200 изображений, каждое из которых необходимо классифицировать на несколько классов (в среднем по датасетам — 20 классов). Объем приватной части тестовых данных не регламентирован и подобран так, чтобы решение, удовлетворяющее ограничениям выше, проходило по времени.

Примечания

Возможные ошибки:

При получении ошшибки `SystemError: google/protobuf/pyext/descriptor.cc:358: bad argument to internal function` стоит проверить, что правильно подключены переменные окружния, как в файле `predict.sh` бейзлайна. Так же стоит вставить импорт `tensorflow` вверх `predict.py` файла, чтобы он импортился перед `torch` и `transformers`.

Выбрать Файл не выбран

Отправить