Использование обучения с подкреплением для решения задачи распознавания диктора в интерактивном режиме

Головин Вячеслав Сергеевич

2023

Содержание

Введение

Данная работа посвящена интерактивному подходу к решению задачи распознавания диктора. Оригинальный метод был предложен в [1], и значительная часть работы посвящена его описанию и практической реализации. С момента публикации этой статьи (2020 год) уже прошло достаточное количество времени, но она не стала популярной — по данным *Google Scholar* на момент написания этого отчёта она была процитирована 6 раз[1]. Тем не менее, нам (автору дипломной работы, его научному руководителю и коллегам из лаборатории Huawei CBG AI) она показалась заслуживающей внимания. На это есть ряд причин.

В первую очередь стоит отметить оригинальность предложенного подхода. Исторически большинство работ, в той или иной степени затрагивающие задачу распознавания диктора, посвящены способам как можно лучше определять диктора на основе уже существующих аудиозаписей. здесь, наверное, нужно привести примеры таких работ Рассматриваемая работа ставит проблему иначе — какие слова или фразы должен произнести диктор, чтобы уже существующая система смогла распознать его как можно быстрее и надёжнее. Чем-то такой подход напоминает концепцию активного обучения (*англ.* active learning) — разметки только тех данных, которые являются наиболее важными для решающей функции.

Другой причиной интереса к работе стала возможность её потенциального использования в конечном продукте — системе аутентификации пользователя на мобильном устройстве или персональном ассистенте. Предополагается, что такая система будет спрашивать пользователя произнести ту или иную фразу, пока она не станет уверена, что перед ней действительно находится настоящий владелец прибора. В таком случае логично делать не случайные запросы, а такие, которые позволят системе как можно быстрее идентифицировать пользователя.

Более подробное описание метода дано в главе 1. Следующая глава посвящена практической реализации описанного метода и полученным результатам. Глава 3 в свою очередь посвящена модификациям оригинального подхода, направленными на повышение точности и адаптации метода под сформулированную выше практическую задачу.

[1]Из этих цитат 1 приходится на кандидатскую диссертацию её первого автора.

1 Распознавание диктора в интерактивном режиме

1.1 Задача распознавания диктора

Распознавание диктора является одной из задач обработки речи — обширного научного и исследовательского направления с долгой и богатой историей. Как и в случае со многими другими научными направлениями, в последние годы обработка речи стала активно использовать методы машинного обучения, в частности нейросетевые модели. привести примеры моделей?

Задача распознавания диктора, как нетрудно понять из названия, заключается в определении личности человека по аудиозаписи его речи. Если говорить чуть более строго, задачей является сопоставление некоторой аудиозаписи речи неизвестного человека с некоторым набором дикторов. В случае решения задачи *идентификации* этот набор состоит из нескольких дикторов, при этом мы точно знаем, что один из них произнес анализируемую нами речь. Соответственно, в таком случае задачей системы является правильный выбор диктора. В случае решения задачи *верификации* нам известна информация только об одном диктора. Таким образом, от нас требуется определить, произнёс ли он речь на предоставленной нам аудиозаписи. криво написано, можно переписать

1.2 Интерактивный режим

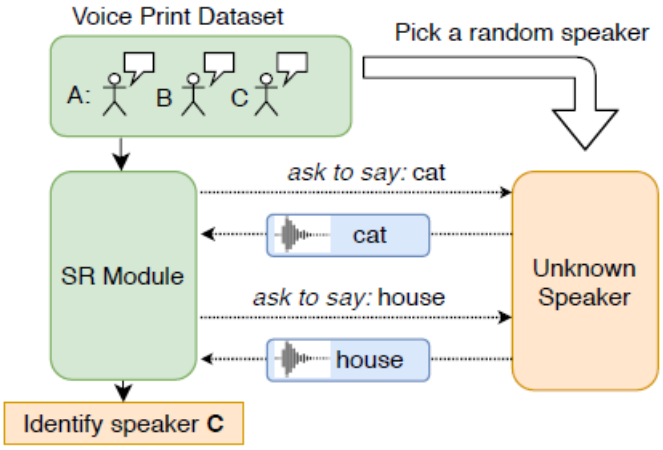

Рис. 1. Схема интерактивной игры по определению диктора [1]

2 Детали реализации и результаты

2.1 Данные для обучения и извлечение эмбеддингов

Здесь мы практически полностью повторяем описанный в [1] подход. Единственным (но очень существенным) отличием является использованная размерность эмбеддингов. Перед тем как перейти к обсуждению этого момента, расскажем про исходные данные.

Итак, для обучения и тестирования моделей мы использовали датасет TIMIT[2]. Он составлен из аудиозаписей речи 630 дикторов, говорящих на 8 основных диалектах американского английского языка. Эти дикторы поделены на обучающую (`train`) и тестовую (`test`) выборки, в первую входят 468 дикторов, во вторую — 162. Для обучения нейросетевых моделей мы также создавали валидационную выборку, в которую выделялись 20% дикторов из обучающей.

Каждый из дикторов произносит 10 фонетически насыщенных предложений. При этом 2 из 10 предложений являются общими для всех дикторов [2] , остальные 8 уникальны для каждого диктора. Такое разделение позволяет без особых затруднений подготовить данные, необходимые для описанной в 1.2 игры:

- 2 общих предложения можно использовать для получения аудиозаписей слов. Для этого разделим аудиозаписи этих предложений по временным отметкам, предоставленным создателями датасета. В результате получим 20 аудиозаписей слов[3] для каждого диктора.

- 8 уникальных для каждого диктора предложений можно использовать для получения голосовых подписей — эмбеддингов дикторов — просто при помощи усреднения эмбеддингов аудиозаписей этих предложений.

В качестве векторов признаков использовались эмбеддинги `x-vector` [3]. Весь процесс преобразования аудиозаписей в векторы признаков осуществлялся с помощью библиотеки Kaldi [4]. На первом этапе рассчитывались мел-частотные кепстральные коэффициенты[4] и производилось детектирование голосовой активности (*англ.* VAD — voice activity detection).

[2]Общие предложения:
She had your dark suit in greasy wash water all year.
Don't ask me to carry an oily rag like that.

[3]Аналогично [1] мы не используем слово *an*.

[4]Параметры аналогичны использованным в [1] и определяются требованиями предобученной модели.

Полученные векторы признаков поступали на вход предобученной нейронной сети [5]. В качестве эмбеддингов использовались данные со второго 512-мерного слоя.

Здесь, как уже было сказано ранее, мы отступаем от оригинальной работы [1], где использовались 128-мерные эмбеддинги. На это есть две причины. Во-первых, из приведенных в [1] комментариев неочевидно[5], как производилось понижение размерности. Во-вторых, мотивация такого преобразования тоже неочевидна. Уже первые проведенные нами эксперименты показали, что при использовании 512-мерных эмбеддингов точность идентификации оказывается существенно выше приведенных в [1] значений.

2.2 Обучение `Guesser`

Первой обучается нейронная сеть `Guesser`, выполняющая выбор из K дикторов при помощи T аудиозаписей произнесенных слов. Как уже было сказано ранее, эта нейросеть тренируется в режиме обучения с учителем, дикторы и произносимые слова выбираются случайно, в качестве функции используется кросс-энтропия. Процесс вычисления значения функции потерь для одной игры можно записать следующим образом:

Листинг 1. Рассчёт функции потерь `Guesser`

```
speaker_ids = speakers.sample(size=K)
G = voice_prints.get(speaker_ids)
target = randrange(0, K)
word_inds = randrange(0, V, size=T)
X = word_vocab.get(speaker=speaker_ids[target],
                   words=word_inds)
probabilities = guesser.forward(G, X)
loss = cross_entropy(probabilities, target)
```

Из-за увеличения относительно [1] размерности эмбеддингов пропорционально увеличились и размерности слоёв `Guesser`. Из-за этого нам пришлось изменить гиперпараметры, в частности мы сильно уменьшили темп обучения (*learning rate*).

Как и в оригинальной статье, для сравнения моделей будем строить графики *word* и *guest sweep*. Т. е. будем обучать модель в режиме с $K = 5$ дикторами и $T = 3$ запрашиваемыми словами, а затем будем тестировать

[5]Цитата: *We then process the MFCCs features through a pretrained X-Vector network to obtain a high quality voice embedding of fixed dimension 128, where the X-Vector network is trained on augmented Switchboard, Mixer 6 and NIST SREs.*

её в режимах с отличным числом дикторов или слов. Здесь и далее, если это не оговорено отдельно, для расчёта точности проводятся 20000 игр среди дикторов из тестовой выборки, эксперименты повторяются по 5 раз с различным `seed` генератора случайных чисел.

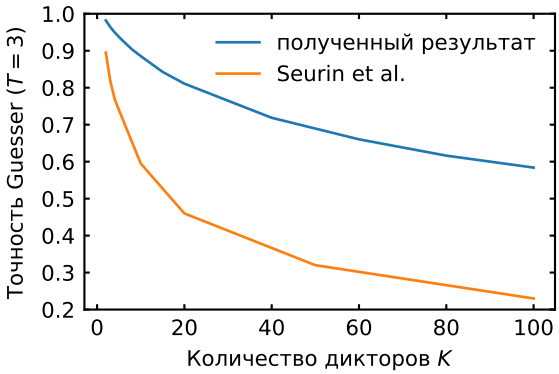

Рис. 2. Зависимость точности `Guesser` обученного нами и авторами [1] от числа дикторов K. Модели обучены в режиме $K = 5$, $T = 3$.

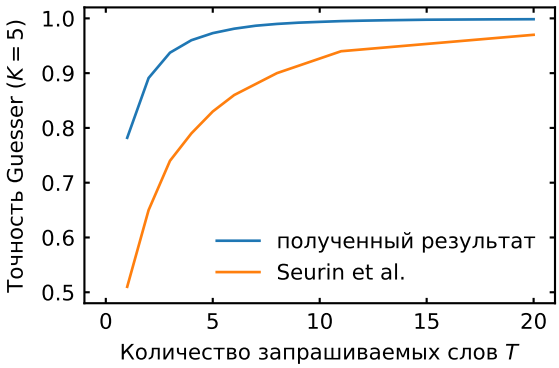

Рис. 3. Зависимость точности `Guesser` обученного нами и авторами [1] от числа запрошенных слов T. Модели обучены в режиме $K = 5$, $T = 3$.

По приведенным на графиках результатам видно, что увеличение размерности эмбеддингов существенно улучшает точность идентификации, разница особо велика в режимах с большим числом дикторов K.

2.3 Обучение `Enquirer`

Для обучения `Enquirer` — модели выбора слов — уже нужна обученная модель `Guesser`. На этом этапе уже используется обучение с подкреплением, псевдокод для 1 игры приведён ниже.

Листинг 2. Интерактивная игра для обучения `Enquirer`

```
speaker_ids = speakers.sample(size=K)
G = voice_prints.get(speaker_ids)
target = randrange(0, K)

g_hat = G.mean(dim=0)
x_i = start_tensor
X = []
for i in range(T):
    probs = enquirer.forward(g_hat, x_i)
    if training:
        word_inds = multinomial(probs).sample()
    else:
        probs[previous_actions] = 0.0
        word_ind = argmax(probs)
    x_i = word_vocab.get(speaker=speaker_ids[target],
                         word=word_ind)
    X.append(x_i)

prediction = guesser.predict(G, X)
reward = 1 if prediction == target else 0
```

Как видно из приведенного псевдокода, награда выдается в том случае, когда `Guesser` правильно угадывает диктора. Для обучения мы использовали алгоритм PPO [6] — здесь мы снова повторяем подход авторов [1]. В целом выбор метода выглядит разумным — PPO зарекомендовал себя как простой и универсальный алгоритм, позволяющий достигать хороших результатов. Однако некоторые особенности нашей задачи — дискретное пространство действий, малая длительность эпизодов — выглядят лучше подходящими для off-policy алгоритмов. К сожалению, у нас не нашлось времени, чтобы проверить эту гипотезу.

Приведенные результаты свидетельствуют о том, что `Enquirer` действительно успешно обучается — точность оказываются заметно выше, чем с случае случайного выбора слов. Как и в случае повышения размерности эмбеддингов, особенно большое различие наблюдается в режимах с большим числом дикторов.

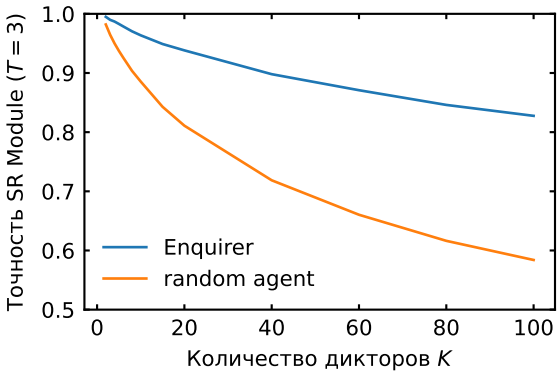

Рис. 4. Зависимость точности SR-систем с различными методами выбора слов — нейросетевым агентов (`Enquirer`) и случайным (random agent) — от числа дикторов K. Модели обучены в режиме $K = 5$, $T = 3$.

2.4 Эвристическая модель выбора слов

Очевидно, что агент, выбирающий запрашиваемые слова случайным образом, не является тяжелым противником для нейросетевого агента. Для более трезвой оценки возможностей последнего, логично сравнивать его с каким-то более сложным алгоритмом.

Здесь мы снова немного отходим от оригинальной статьи. И опять основной причиной является тот факт, что в [1] отсутствует точное описание использованного в качестве бейзлайна эвристического алгоритма выбора слов. Из приведенных в работе слов[6] общий подход понятен — сэмплирование производится не из всех 20 слов, а из тех, которые в среднем показывают самую высокую точность. При этом остаются непонятными следующие детали:

1. Из скольки слов производится сэмплирование, и меняется ли это число в зависимости от числа запрашиваемых слов T?

2. Производится ли сэмплирование равномерно, или вероятность выбрать слово пропорциональна достигаемой при выборе этого слова средней точности?

Именно такие вопросы возникли у нас при создании эвристического агента. Первым же этапом стала оценка слов — расчёт средней точности, которая достигается случайным агентом в тех играх, когда он выбрал то

[6]Цитата: *We curated a list of the most discriminant words (words that increase globally the recognition scores) and sample among those instead of the whole list.*

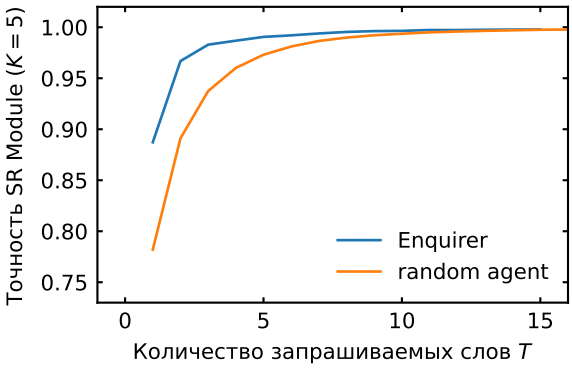

Рис. 5. Зависимость точности SR-систем с различными методами выбора слов — нейросетевым агентов (`Enquirer`) и случайным (random agent) — от числа запрашиваемых слов T. Модели обучены в режиме $K = 5$, $T = 3$.

или иное слово. Для этого мы протестировали `Guesser` в 100000 эпизодов с $K = 5$, и $T = 3$, а также случайным выбором слов без повторений. Мы рассчитывали точность для каждого слова, учитывая только те эпизоды, в которых это слово было выбрано. Фактически мы оценивали условную вероятность связки `Guesser`–случайный агент правильно выбрать диктора при условии, что одно слово уже было выбрано. ссылка на рисунок

После этого мы стали тестировать различные модификации эвристического агента. Как следует из сформулированных выше вопросов, эти агенты отличались числом использованных слов и методом сэмплирования. Эксперименты показали, что наилучшие результаты достигаются при использовании “детерминированного” агента, всегда выбирающего одни и те же слова с наибольшей средней точностью. В таком случае говорить о каком-либо сэмплировании неуместно, поэтому такой агент, по всей видимости, отличается от использованного в оригинальной статье.

В данном случае преимущество `Enquirer` проявляется только в режимах с большим числом дикторов. В стандартном режиме с $K = 5$ дикторами и $T = 3$, в отличие от [1], мы не наблюдаем сколько-нибудь существенной разницы между двумя агентами.

В таком случае возникает резонный вопрос — не сходится ли `Enquirer` к такой же политике, что использует эвристический агент? Ответ на этот вопрос — отрицательный, в целом `Enquirer` выбирает из 5 слов (ещё 2 используются редко), в то время как эвристический агент всегда использует 3 тех же слова. Из этого можно предположить, что `Enquirer` обучен

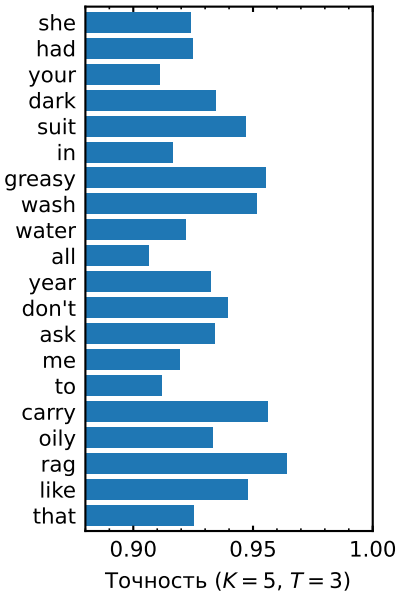

Рис. 6. Средняя точность `Guesser` на валидационной выборке в тех эпизодах, когда соответствующее слово было выбрано (остальные выбирались случайно).

недостаточно хорошо, возможно, другие гиперпараметры или алгоритм обучения позволили бы улучшить результаты.

2.5 Обучение в других режимах

Другой логичный вопрос, возникающий при обсуждении графиков *word* и guest sweep — является ли стандартный режим ($K = 5$ дикторов и $T = 3$ запрашиваемых слова) оптимальным для обучения моделей? Не будут ли результаты лучше, если мы будем обучать и тестировать модели в одном и том же режиме? Мы также провели ряд экспериментов и пришли к следующим выводам:

- Общее правило — более тяжелые режимы позволяют улучшить точность. В первую очередь это касается увеличения числа дикторов, ситуация с уменьшением числа слов менее однозначная.
- Основные улучшения наблюдаются в работе `Guesser`, в то же время `Enquirer` оказывается нечувствительным к режиму обучения.

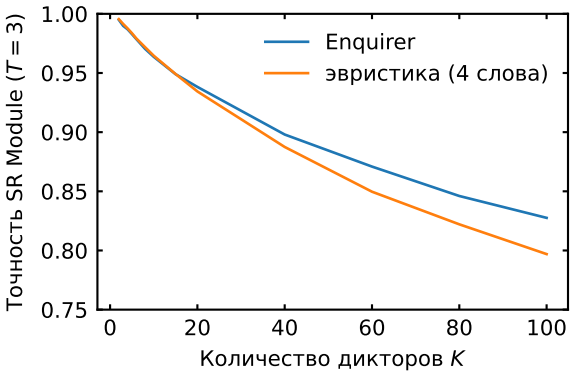

Рис. 7. Зависимость точности SR-систем с различными методами выбора слов — нейросетевым и эвристическим агентами — от числа дикторов K. Модели обучены в режиме $K = 5$, $T = 3$.

- Обучение при $T = 1$ является специфической задачей. Во-первых, только в этом режиме обученная модель показывает хорошие (лучше чем другие) результаты только в этом же режиме тестирования. Во-вторых, `Enquirer` в среднем показывает плохие результаты в данном режиме, часто уступая максимально простой политике, всегда выбирающей одно и то же слово.

3 Модификации метода

3.1 От идентификации к верификации

`Enquirer` менять вообще не нужно, `Guesser` — совсем немного.

3.2 `CodebookEnquirer`

вроде работает

3.3 Добавление шума

обучается норм, результаты такие же

3.4 Альтернативные эмбеддинги

внезапно эмбеддинги 2017 года оказались не очень

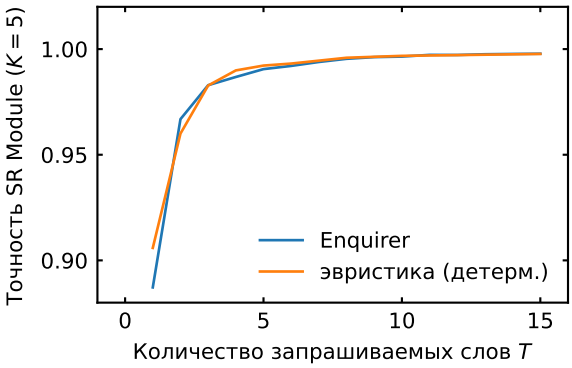

Рис. 8. Зависимость точности SR-систем с различными методами выбора слов — нейросетевым и эвристическим агентами — от числа запрашиваемых слов T. Модели обучены в режиме $K = 5$, $T = 3$.

Заключение

все работает, но хотелось бы большего

Список литературы

[1] M. Seurin, F. Strub, P. Preux и O. Pietquin, *A machine of few words – interactive speaker recognition with reinforcement learning*, 2020. arXiv: 2008.03127 [eess.AS].

[2] Garofolo, John S. и др., *TIMIT acoustic-phonetic continuous speech corpus*, 1993. DOI: 10.35111/17GK-BN40. url: https://catalog.ldc.upenn.edu/LDC93S1.

[3] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey и S. Khudanpur, «X-Vectors: Robust DNN embeddings for speaker recognition,» в *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, апр. 2018. DOI: 10.1109/icassp.2018.8461375. url: https://www.danielpovey.com/files/2018_icassp_xvectors.pdf.

[4] D. Povey и др., «The kaldi speech recognition toolkit,» в *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, IEEE Catalog No.: CFP11SRW-USB, Hilton Waikoloa Village, Big Island, Hawaii, US: IEEE Signal Processing Society, дек. 2011.

[5] «SRE16 Xvector Model.» (2017), url: http://kaldi-asr.org/models/m3 (дата обр. 18.05.2023).

[6] J. Schulman, F. Wolski, P. Dhariwal, A. Radford и O. Klimov, *Proximal policy optimization algorithms*, 2017. arXiv: 1707.06347 [cs.LG].