

# Interactive Speaker Recognition

Применение обучения с подкреплением для решения задачи распознавания диктора

Вячеслав Головин  
Евгений Шуранов (руководитель)

Huawei CBG AI и ФКН ВШЭ СПб

10.05.2023

**Задача:** повышение точности систем верификации / идентификации диктора. Такая система может, например, быть использована для подтверждения личности на мобильных устройствах.

### **Требования к системе:**

- короткие запросы (не раздражаем пользователя),
- разнообразные запросы (не боимся спуфинга),
- высокая точность (без комментариев).

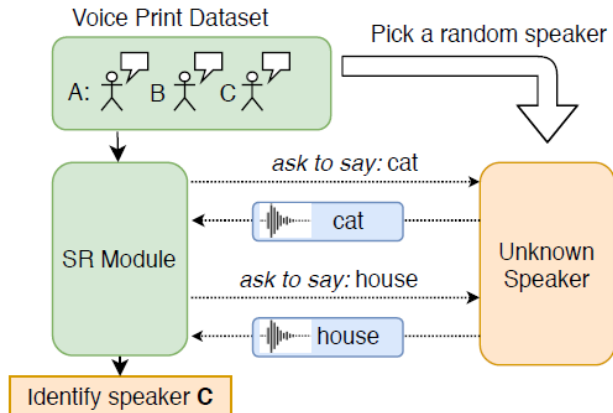
**Предлагаемое решение:** использование RL-агента для выбора запрашиваемых слов.

### **Новизна дипломной работы:**

- переход от идентификации к верификации,
- более гибкая система для выбора слов.

# Interactive Speaker Recognition

Метод был предложен в статье *A Machine of Few Words — Interactive Speaker Recognition with Reinforcement Learning*, Mathieu Seurin et al., INTERSPEECH 2020, arXiv:2008.03127v1.

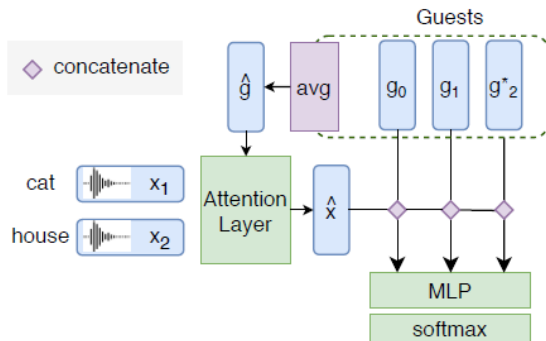


Важные особенности:

- 1 Рассматривается только задача идентификации.
- 2 Набор слов строго фиксирован.
- 3 Разные нейронные сети для двух задач SR Module — запроса слов (Enquirer) и идентификации диктора (Guesser).

# Блок Guesser

## Архитектура



Входные данные:

- эмбеддинги дикторов  
 $G = [g_1; g_2; \dots g_K]$
- эмбеддинги слов  
 $X = [x_1; x_2; \dots x_T]$

Выходные данные:

- вероятности  
 $\{P(g_i = g^*) \mid i = 1..K\}$

## Обозначения

- $K$  количество гостей / дикторов  
 $T$  количество запрашиваемых слов

# Блок Guesser

## Псевдокод 1 итерации обучения

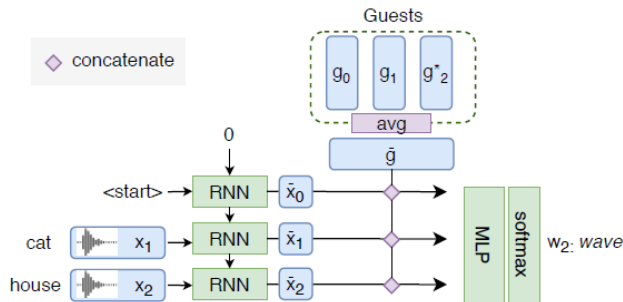
```
speaker_ids = speakers.sample(size=K)
G = voice_prints.get(speaker_ids)
target = randrange(0, K)
word_inds = randrange(0, V, size=T)
X = word_vocab.get(speaker=speaker_ids[target],
                  words=word_inds)
probabilities = guesser.forward(G, X)
loss = cross_entropy(probabilities, target)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

### Обозначения

- $K$  количество гостей / дикторов
- $T$  количество запрашиваемых слов
- $V$  размер словаря — число доступных для запроса слов

# Блок Enquirer

## Архитектура



Входные данные:

- среднее эмб. дикторов

$$\hat{g} = \frac{1}{K} \sum_{i=1}^K g_k$$

- эмбединги слов

$$X = [x_1; x_2; \dots; x_t]$$

Выходные данные:

- вероятность выбрать каждое из слов

## Обозначения

- $K$  количество гостей / дикторов  
 $T$  количество запрашиваемых слов  
 $t$  количество запрошенных слов,  $0 \leq t \leq T$

# Блок Enquirer

## Псевдокод 1 эпизода ISR-игры

```
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:
        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
```

## Входные данные

Для обучения использовался датасет **TIMIT**:

- 630 дикторов из США, 8 акцентов;
- каждый диктор произносит 10 предложений: 8 уникальных и 2 общих.

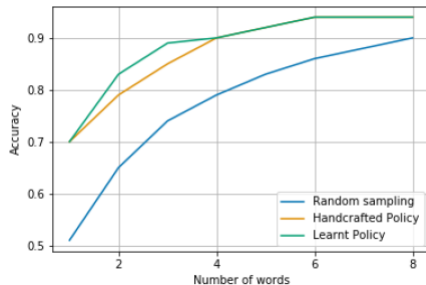
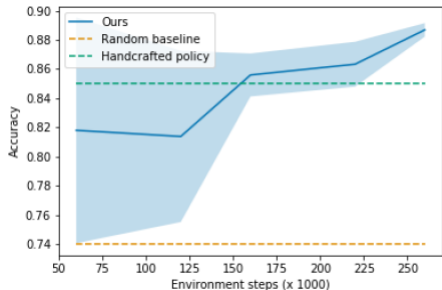
В качестве эмбедингов использовались **x-vectors**, полученные с помощью нейронной сети, обученной на аугментированных датасетах для распознавания диктора *Switchboard*, *Mixer 6* и *NIST*.

- Эмбединги дикторов  $g$  получались с помощью усреднения эмбедингов 8 уникальных предложений.
- Эмбединги слов  $x$  извлекались с помощью 2 общих предложений, т.е. сначала вырезались записи одиночных слов, которые затем пропусклись через нейронную сеть.



# Результаты из статьи

$K = 5$  дикторов и  $T = 3$  слова

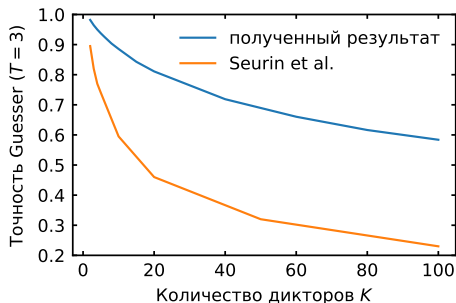
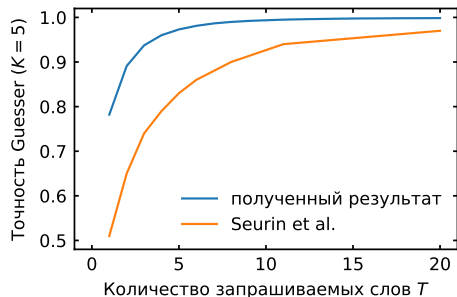


RL-агент при выборе запрашиваемых слов учитывает контекст — он опережает не только случайного агента, но и эвристического, выбирающего из подмножества “лучших” слов.

Преимущество RL-агента невелико и проявляется только при небольшом числе запрашиваемых слов.

# Обучение и тестирование Guesser

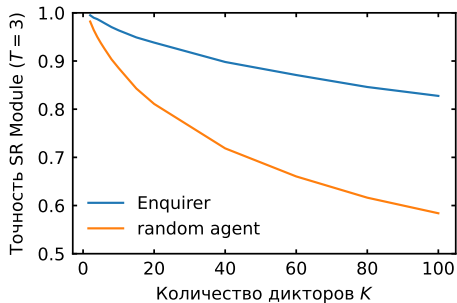
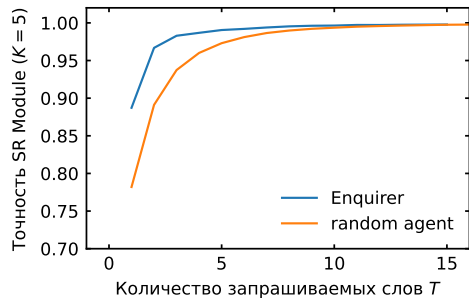
$K = 5$  дикторов и  $T = 3$  слова при обучении



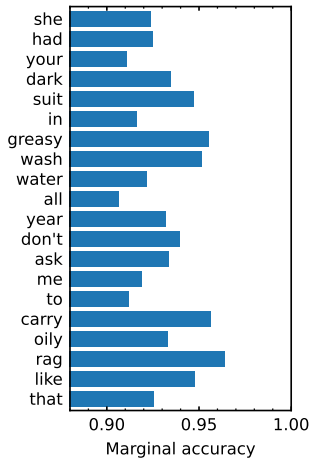
Вероятно, главная причина расхождения результатов — увеличение размерности эмбедингов (512 вместо 128 в статье). Неизвестно, как и зачем в статье производилось понижение размерности.

# Обучение и тестирование Enquirer

$K = 5$  дикторов и  $T = 3$  слова при обучении

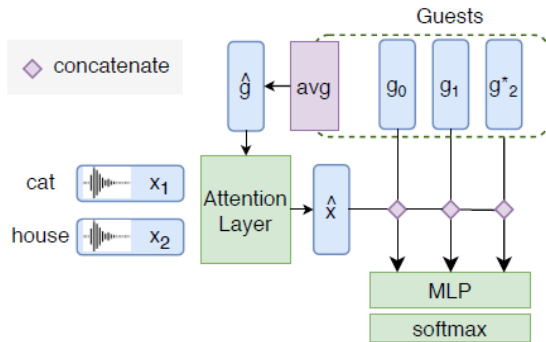


# Эвристический агент



# Верификация

## Как преобразовать Guesser



- только 1 диктор  $\implies \hat{g} = g_0$
- softmax  $\rightarrow$  sigmoid
- при  $T = 3$  точность 0.91  
( $\sim$  как в классификации с  $K = 7$  гостями)
- MLP  $\rightarrow$  CosineSimilarity тоже работает, но хуже