Распознавание речи: как сделать Speech-to-Text своими руками

Иван Бондаренко
(МФТИ, Data Monsters, НГУ)

Интервью надо расшифровать. Как?

- Вручную

Интервью надо расшифровать. Как?

- ~~Вручную~~
- Автоматически

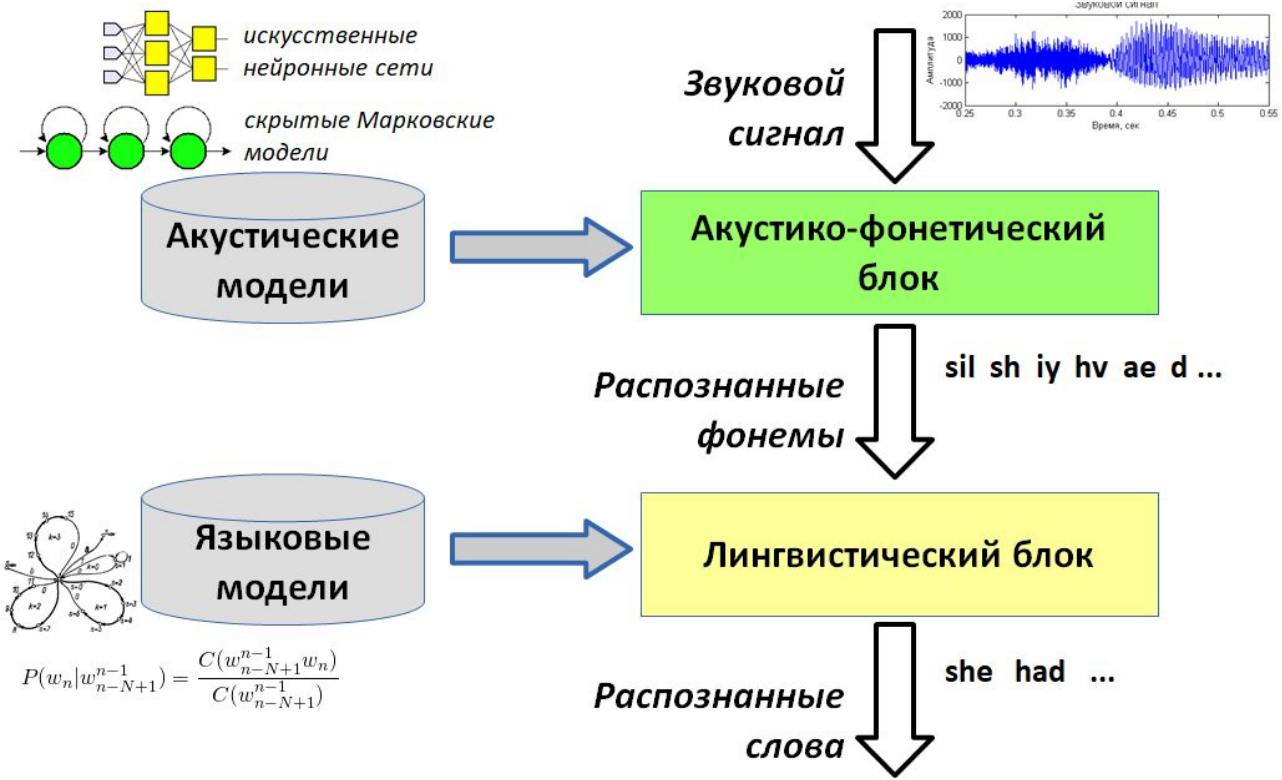
Распознавание речи

- Пропиетарные системы
  - Nuance, Google, Яндекс, ЦРТ...
- Open-source системы
  - CMU Sphinx
  - HTK
  - Kaldi
  - ...

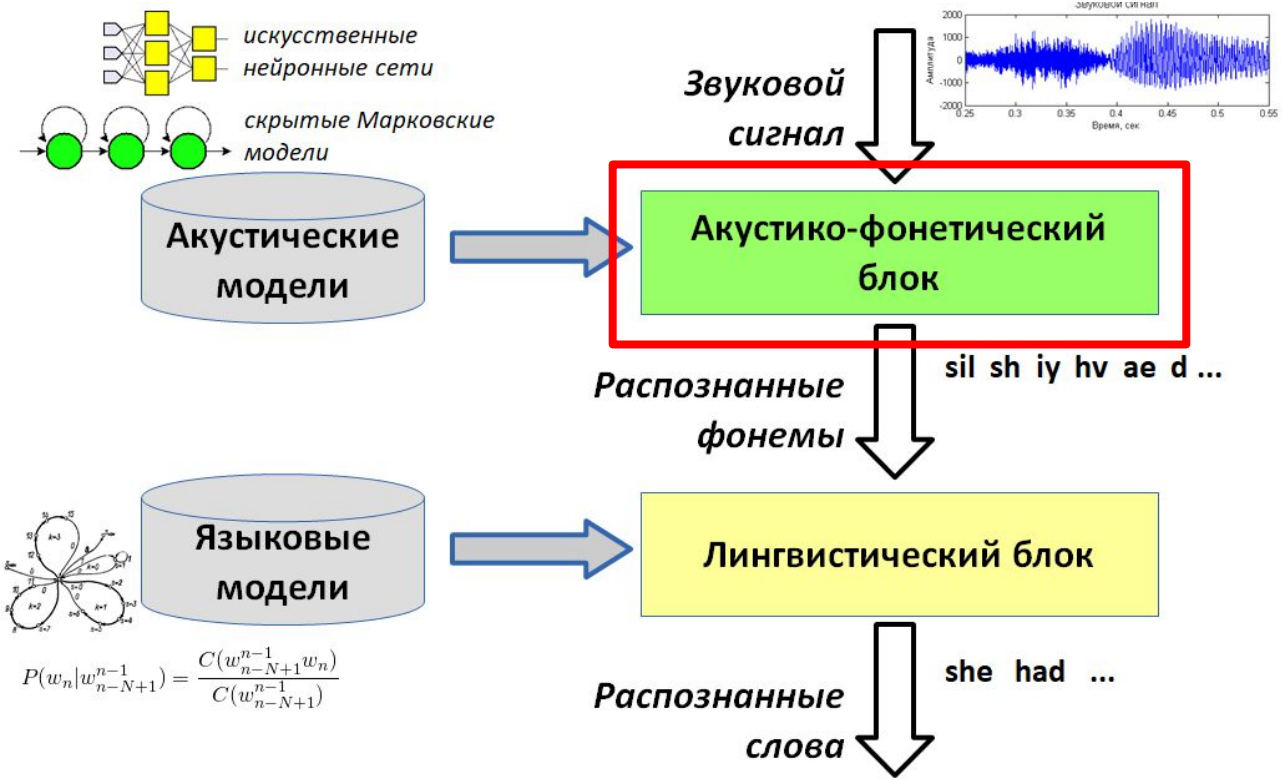
CMU Sphinx

https://cmusphinx.github.io/

Классическая структура системы распознавания речи



Классическая структура системы распознавания речи



$$P(w_n|w_{n-N+1}^{n-1}) = \frac{C(w_{n-N+1}^{n-1}w_n)}{C(w_{n-N+1}^{n-1})}$$

Акустико-фонетический блок

- CMU Sphinx
  - https://cmusphinx.github.io/wiki/tutorialam/#training
- Речевой корпус
  - http://voxforge.org/ru

Подготовка к запуску обучения

```
├─ etc
│   ├─ your_db.dic                    (Словарь транскрипций)
│   ├─ your_db.phone                  (Список фонем)
│   ├─ your_db.lm                     (Языковая модель)
│   ├─ your_db.filler                 (Список филлеров - неречевых квазифонем)
│   ├─ your_db_train.fileids          (Список звукозаписей для обучения)
│   ├─ your_db_train.transcription    (Аннотации звукозаписей для обучения)
│   ├─ your_db_test.fileids           (Список звукозаписей для тестирования)
│   └─ your_db_test.transcription     (Аннотации звукозаписей для тестирования)
└─ wav
    ├─ speaker_1
    │   └─ file_1.wav                 (Звукозаписи в формате WAV PCM)
    └─ speaker_2
        └─ file_2.wav
```

Создание конфига

sphinxtrain -t my_speech_system setup

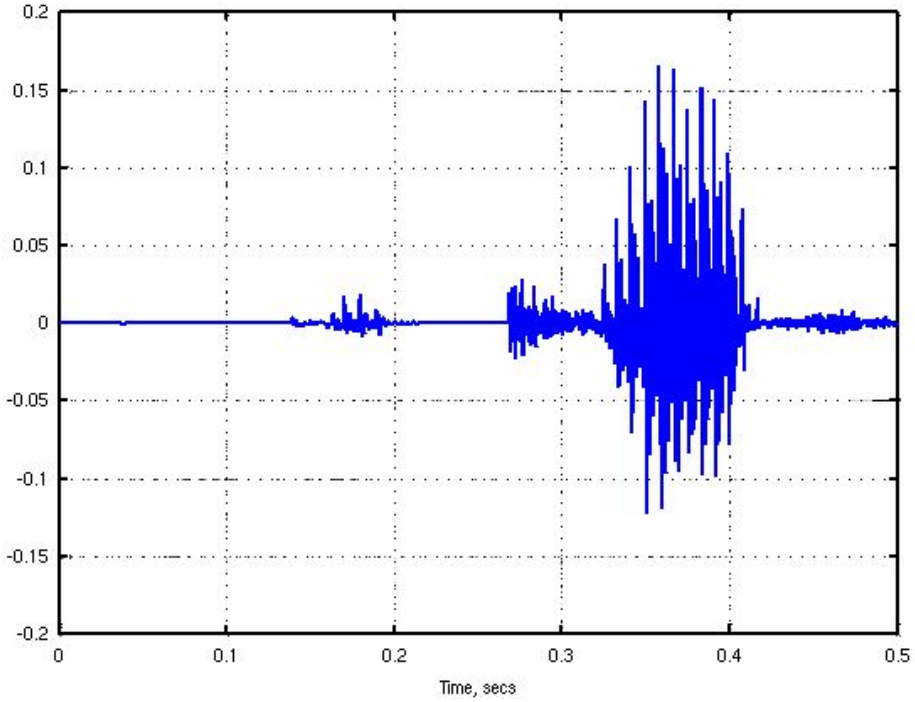
```
├─ etc
│   ├─ sphinx_train.cfg          (Конфигурационный файл)
│   ├─ your_db.dic               (Словарь транскрипций)
│   ├─ your_db.phone             (Список фонем)
│   ├─ your_db.lm                (Языковая модель)
│   ├─ your_db.filler            (Список филлеров - неречевых квазифонем)
│   ├─ your_db_train.fileids     (Список звукозаписей для обучения)
│   ├─ your_db_train.transcription (Аннотации звукозаписей для обучения)
│   ├─ your_db_test.fileids      (Список звукозаписей для тестирования)
│   └─ your_db_test.transcription  (Аннотации звукозаписей для тестирования)
└─ wav
    ├─ speaker_1
    │   └─ file_1.wav            (Звукозаписи в формате WAV PCM)
    └─ speaker_2
        └─ file_2.wav
```

sphinx_train.cfg

```
# Feature extraction parameters
$CFG_WAVFILE_SRATE = 8000.0;
$CFG_NUM_FILT = 15; # For wideband speech it's 25, for telephone 8khz reasonable value
is 15
$CFG_LO_FILT = 200; # For telephone 8kHz speech value is 200
$CFG_HI_FILT = 3500; # For telephone 8kHz speech value is 3500
$CFG_TRANSFORM = "dct"; # Previously legacy transform is used, but dct is more accurate
$CFG_LIFTER = "22"; # Cepstrum lifter is smoothing to improve recognition
$CFG_VECTOR_LENGTH = 13; # 13 is usually enough
```

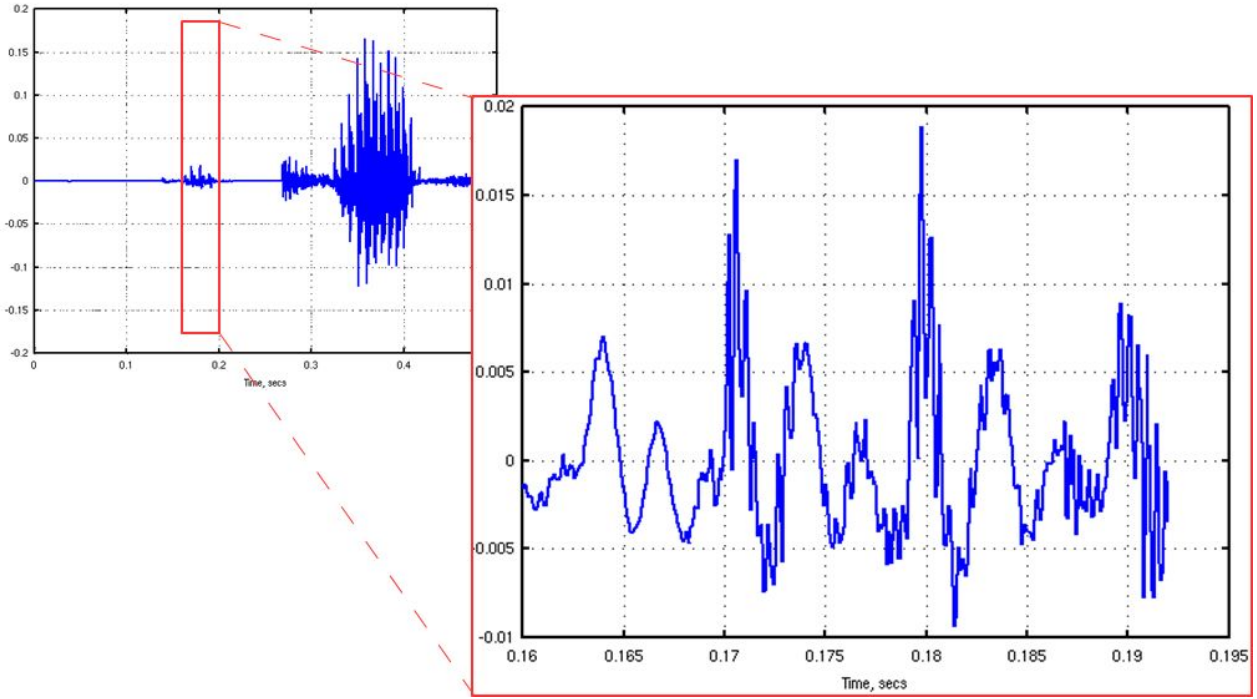
Звуковой сигнал

```
$CFG_WAVFILE_SRATE = 8000.0;
```

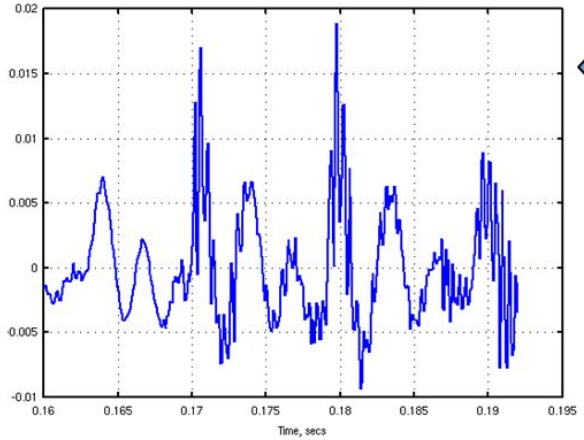


Оконный анализ звукового сигнала

Начало английской фразы
«Be careful not to plow over the flower beds»

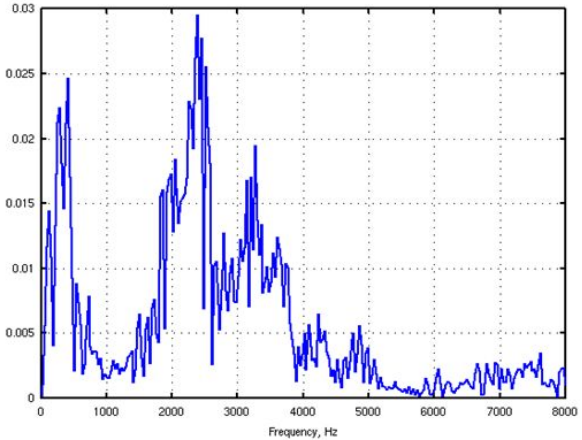


Сигнал и его спектр



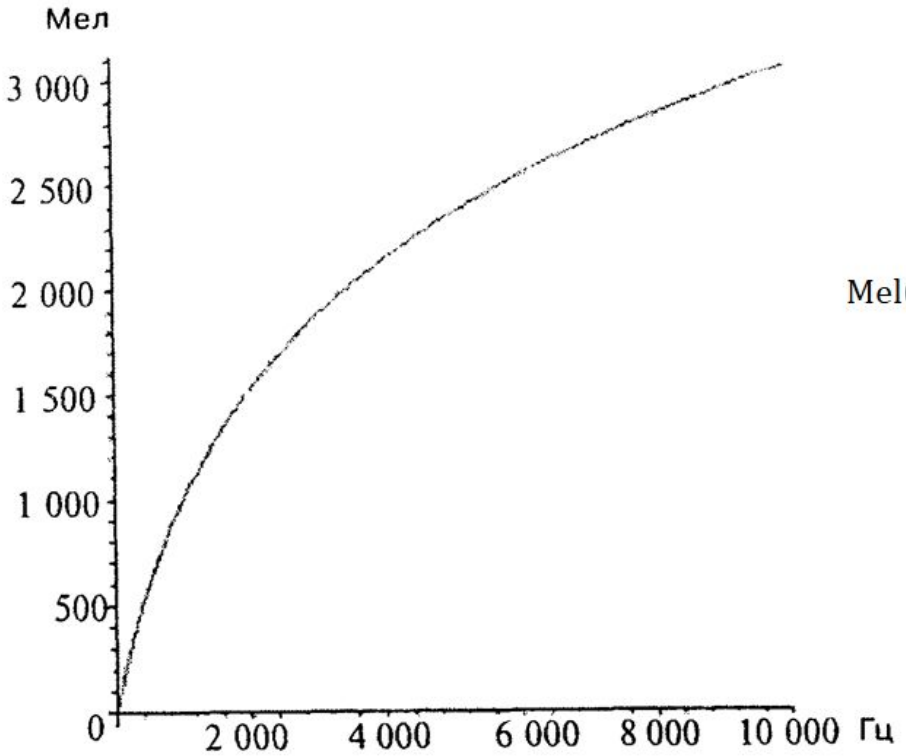
Фрагмент речевого сигнала длиной 25 миллисекунд

Энергетический спектр, вычисленный на этом фрагменте



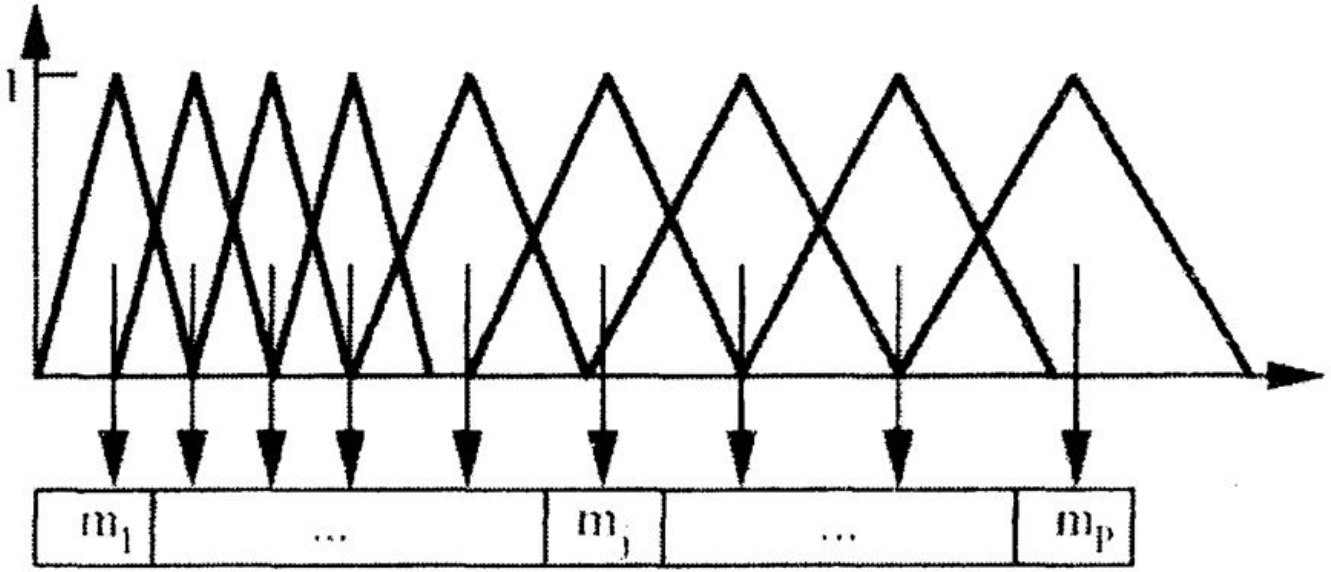

```
$CFG_LO_FILT = 200;
$CFG_HI_FILT = 3500;
```

Мел-шкала



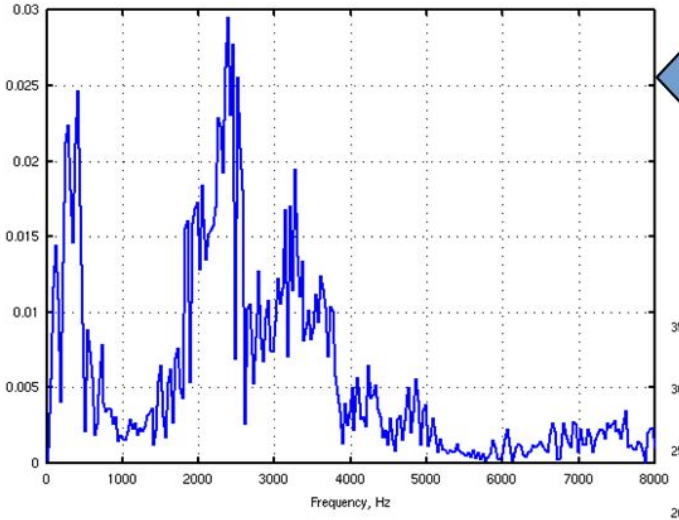
$$\mathrm{Mel}(f) = 2595 \cdot \log_{10}\left(1 + \frac{f}{700}\right)$$

Гребёнка треугольных фильтров



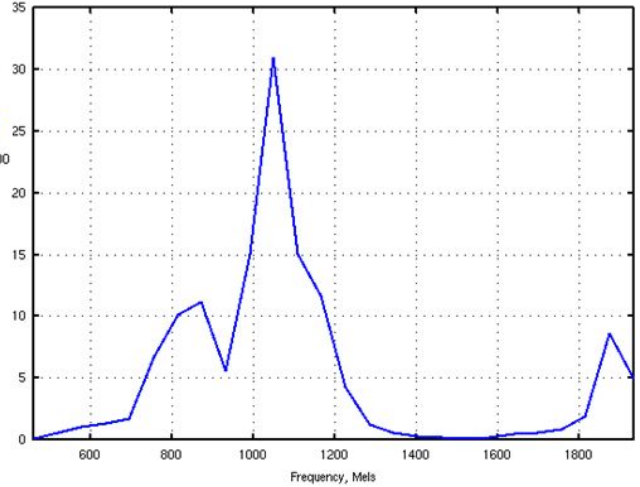
$CFG_NUM_FILT = 15;

Исходный и сглаженный спектры



Исходный энергетический спектр

Сглаженный мел-спектр



sphinx_train.cfg

```
# Variables used in main training of models
$CFG_DICTIONARY     = "$CFG_LIST_DIR/$CFG_DB_NAME.dic";
$CFG_RAWPHONEFILE   = "$CFG_LIST_DIR/$CFG_DB_NAME.phone";
$CFG_FILLERDICT     = "$CFG_LIST_DIR/$CFG_DB_NAME.filler";
```

Фонемы

```
$CFG_RAWPHONEFILE    = "$CFG_LIST_DIR/$CFG_DB_NAME.phone";
```

мама мыла раму

M A0 M A M Y0 L A R A0 M U

Где взять фонемы?

- Прочитать труды лингвистов и придумать систему фонем

Где взять фонемы?

- ~~Прочитать труды лингвистов и придумать систему фонем~~
- взять отсюда https://github.com/nsu-ai/voxforge_ru_sphinx

https://github.com/nsu-ai/voxforge_ru_sphinx/blob/ruscorpora-ngrams/voxforge_ru/etc/voxforge_ru.phone

Словарь транскрипций

```
$CFG_DICTIONARY     = "$CFG_LIST_DIR/$CFG_DB_NAME.dic";


...
мама M A0 M A
папа P A0 P A
мыла M Y0 L
мыл M Y0 L
раму R A0 M U
синхрофазотрон S0 I N KH R A F A Z A T R O0 N
...
```

Где взять словарь транскрипций?

- Изучить закономерности устной речи и выписать 800 000 транскрипций вручную

Где взять словарь транскрипций?

- ~~Изучить закономерности устной речи и выписать 800 000 транскрипций вручную~~
- взять отсюда https://github.com/nsu-ai/voxforge_ru_sphinx

https://github.com/nsu-ai/voxforge_ru_sphinx/blob/ruscorpora-ngrams/voxforge_ru/etc/voxforge_ru.dic

более **800 тысяч** словоформ

А если нужен миллион слов?

Транскрипции для недостающих слов можно нагенерировать автоматически!

1) с помощью словарей и правил “буква-фонема” https://github.com/nsu-ai/russian_g2p
2) с помощью машинного обучения https://github.com/nsu-ai/russian_g2p_neuro

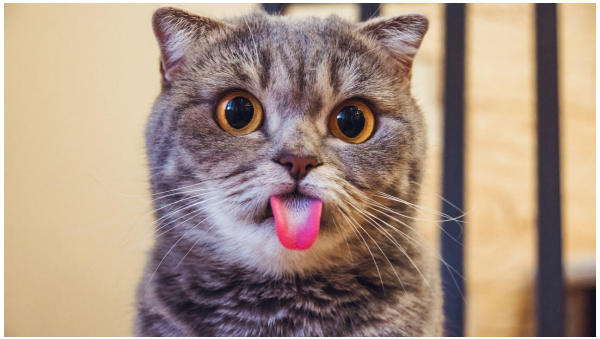
Филлеры (квазифонемы)

```
$CFG_FILLERDICT     = "$CFG_LIST_DIR/$CFG_DB_NAME.filler";
```

```
<s> SIL
</s>    SIL
<sil>   SIL
```

Виды филлеров

Мяу-мяу!



Ту-ту!

Би-би!

Где взять записи филлеров?

- Собрать самому

Где взять записи филлеров?

- Собрать самому
- Скачать в интернете
  https://www.kaggle.com/c/freesound-audio-tagging/data

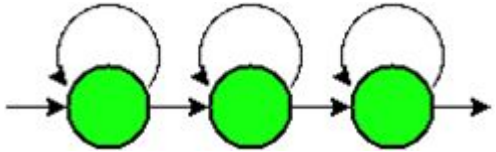
sphinx_train.cfg

```
# Variables used in characterizing models

$CFG_HMM_TYPE = '.cont.'; # Sphinx 4, PocketSphinx
#$CFG_HMM_TYPE  = '.semi.'; # PocketSphinx
#$CFG_HMM_TYPE  = '.ptm.'; # PocketSphinx (larger data sets)

if (($CFG_HMM_TYPE ne ".semi.")
    and ($CFG_HMM_TYPE ne ".ptm.")
    and ($CFG_HMM_TYPE ne ".cont.")) {
  die "Please choose one CFG_HMM_TYPE out of '.cont.', '.ptm.', or '.semi.', " .
    "currently $CFG_HMM_TYPE\n";
}

# This configuration is fastest and best for most acoustic models in
# PocketSphinx and Sphinx-III.  See below for Sphinx-II.
$CFG_STATESPERHMM = 3;
$CFG_SKIPSTATE = 'no';
```

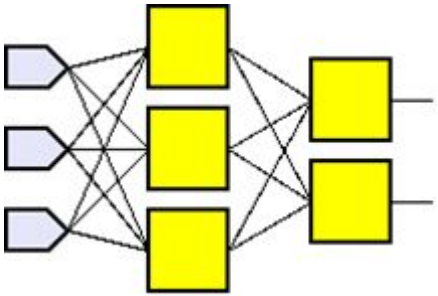
Распознавание фонем

Два подхода к распознаванию фонем:

1) генеративный (СММ, ИКДП)

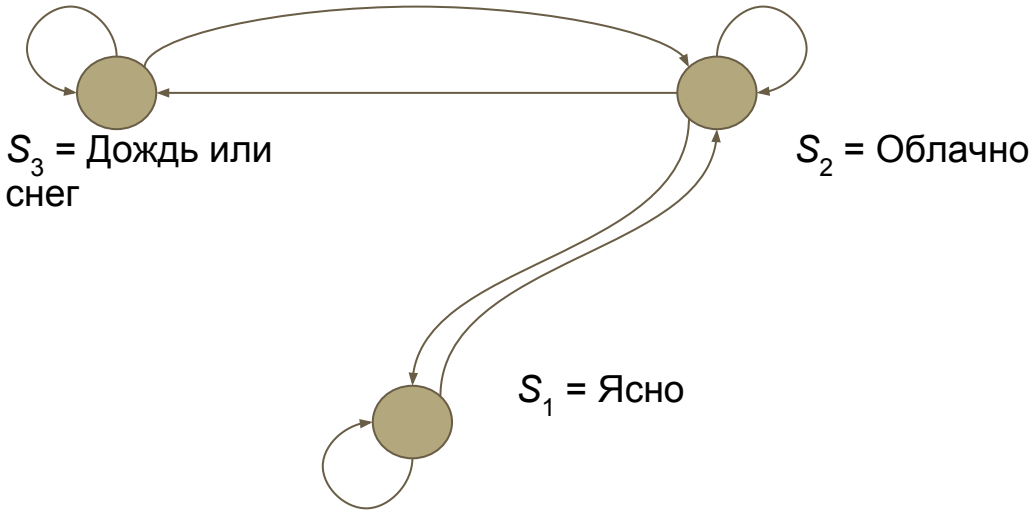


2) дискриминативный (ИНН)



Марковский процесс

«Будущее» процесса не зависит от «прошлого» при известном «настоящем».



Вопросы к марковской модели

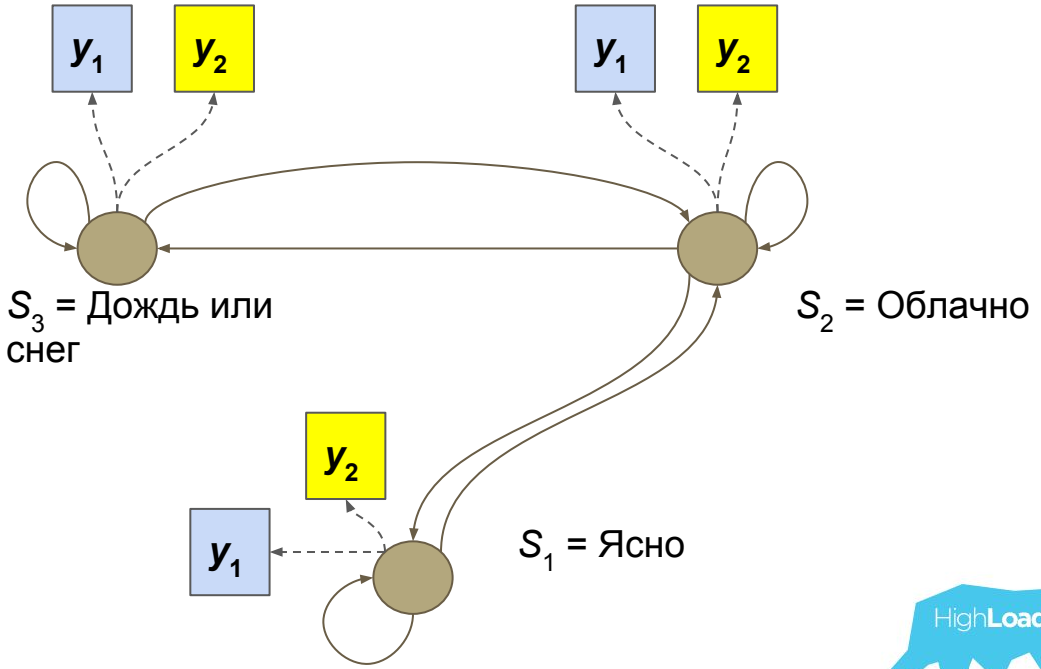
- Какова вероятность последовательности состояний погоды “ясно-облачно-дождь-облачно-ясно”, если известно, что сейчас ясно?
- Какова вероятность того, что солнечная погода будет оставаться солнечной ровно 5 часов подряд?
- Какова наиболее вероятная длительность дождливого периода?
- …

Скрытая марковская модель

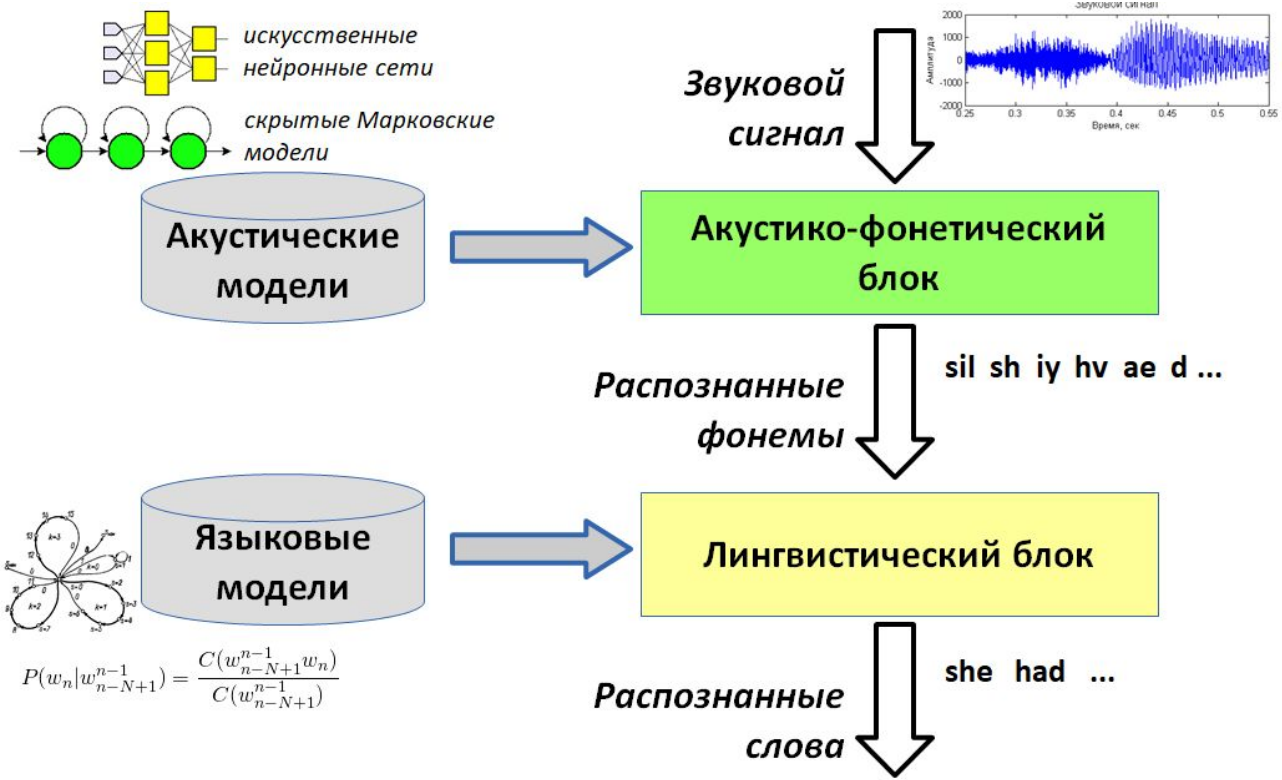
Состояния модели скрыты, мы о них лишь догадываемся по наблюдениям

y_1 =

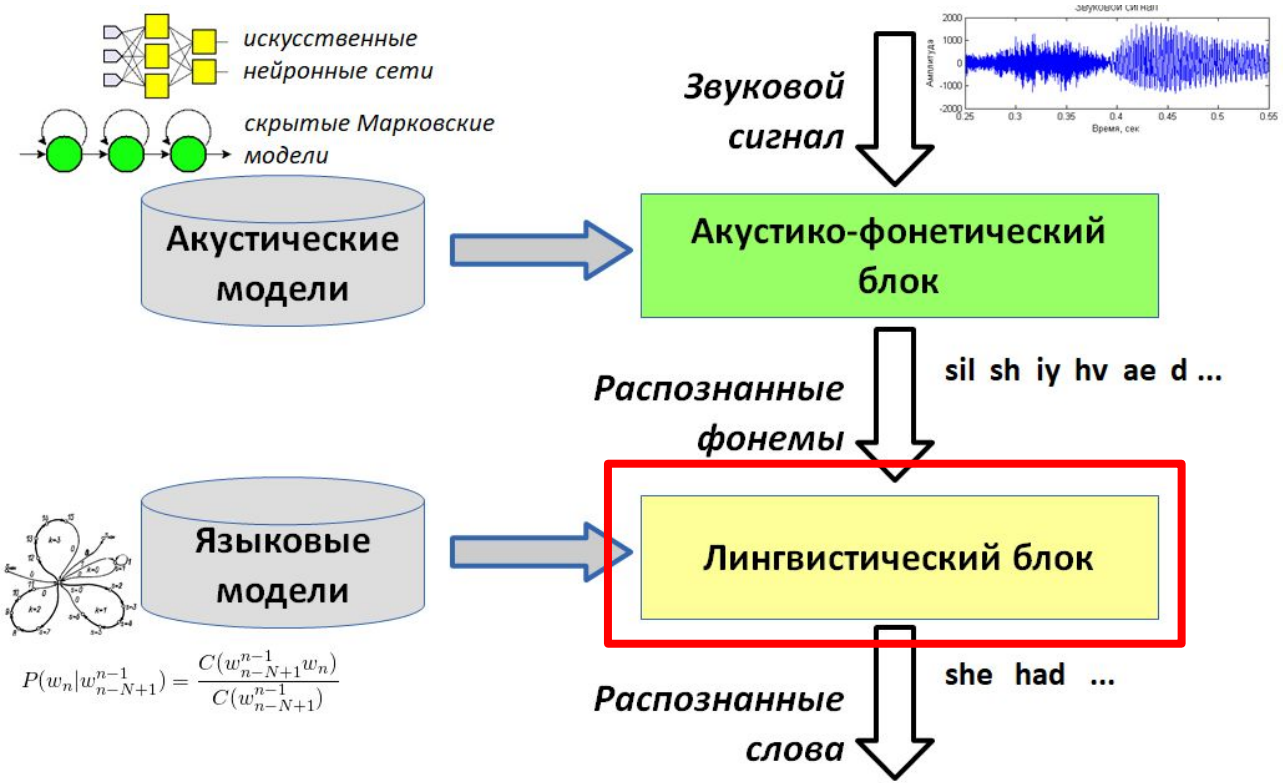
y_2 =



Классическая структура системы распознавания речи



Классическая структура системы распознавания речи



$$P(w_n|w_{n-N+1}^{n-1}) = \frac{C(w_{n-N+1}^{n-1}w_n)}{C(w_{n-N+1}^{n-1})}$$

Языковые модели

```
$DEC_CFG_LANGUAGEMODEL  = "$CFG_BASE_DIR/etc/${CFG_DB_NAME}.lm.bin";
# Or can be JSGF or FSG too, used if uncommented
# $DEC_CFG_GRAMMAR  = "$CFG_BASE_DIR/etc/${CFG_DB_NAME}.jsgf";
# $DEC_CFG_FSG  = "$CFG_BASE_DIR/etc/${CFG_DB_NAME}.fsg";
```

Языковые модели

```
$DEC_CFG_LANGUAGEMODEL  = "$CFG_BASE_DIR/etc/${CFG_DB_NAME}.lm.bin";
# Or can be JSGF or FSG too, used if uncommented
# $DEC_CFG_GRAMMAR  = "$CFG_BASE_DIR/etc/${CFG_DB_NAME}.jsgf";
# $DEC_CFG_FSG  = "$CFG_BASE_DIR/etc/${CFG_DB_NAME}.fsg";
```

Детерминированные Вероятностные

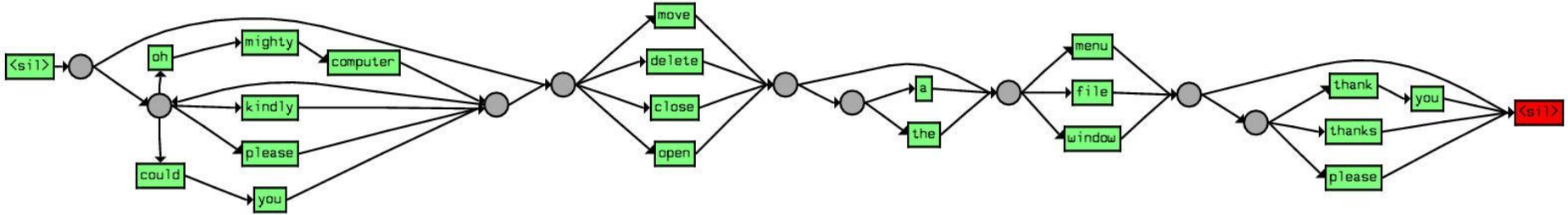
Языковые модели

```
$DEC_CFG_LANGUAGEMODEL  = "$CFG_BASE_DIR/etc/${CFG_DB_NAME}.lm.bin";
# Or can be JSGF or FSG too, used if uncommented
# $DEC_CFG_GRAMMAR  = "$CFG_BASE_DIR/etc/${CFG_DB_NAME}.jsgf";
# $DEC_CFG_FSG  = "$CFG_BASE_DIR/etc/${CFG_DB_NAME}.fsg";
```

Детерминированные

Вероятностные

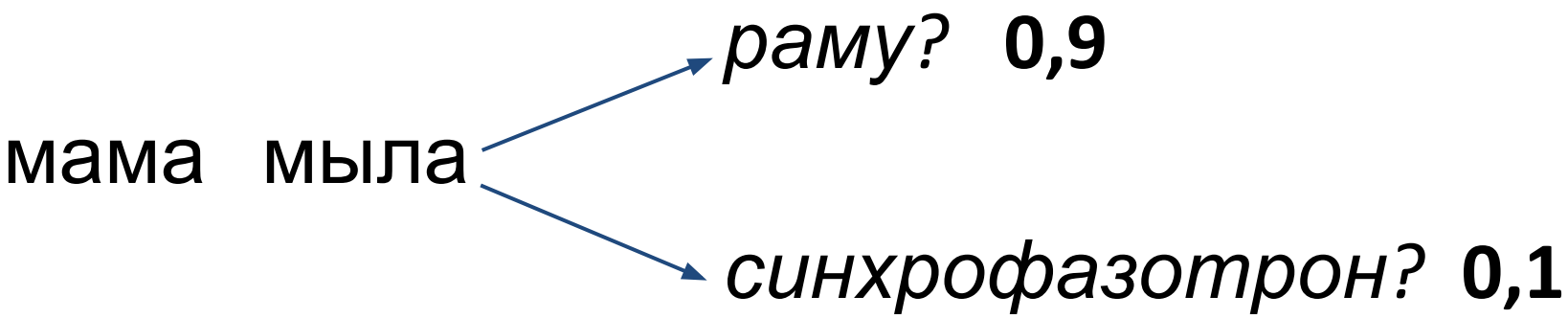
Детерминированные модели



https://www.w3.org/TR/jsgf/

Вероятностные модели

N-граммы



https://cmusphinx.github.io/wiki/arpaformat/

Как построить N-граммы?

- SRILM http://www.speech.sri.com/projects/srilm/
  - ngram-count
    http://www.speech.sri.com/projects/srilm/manpages/ngram-count.1.html
- Большой-пребольшой текстовый корпус
  - Национальный корпус русского языка (НКРЯ) http://www.ruscorpora.ru/corpora-freq.html

Где взять готовые N-граммы?

взять отсюда https://github.com/nsu-ai/voxforge_ru_sphinx

https://github.com/nsu-ai/voxforge_ru_sphinx/blob/ruscorpora-ngrams/voxforge_ru/etc/voxforge_ru.lm.bin

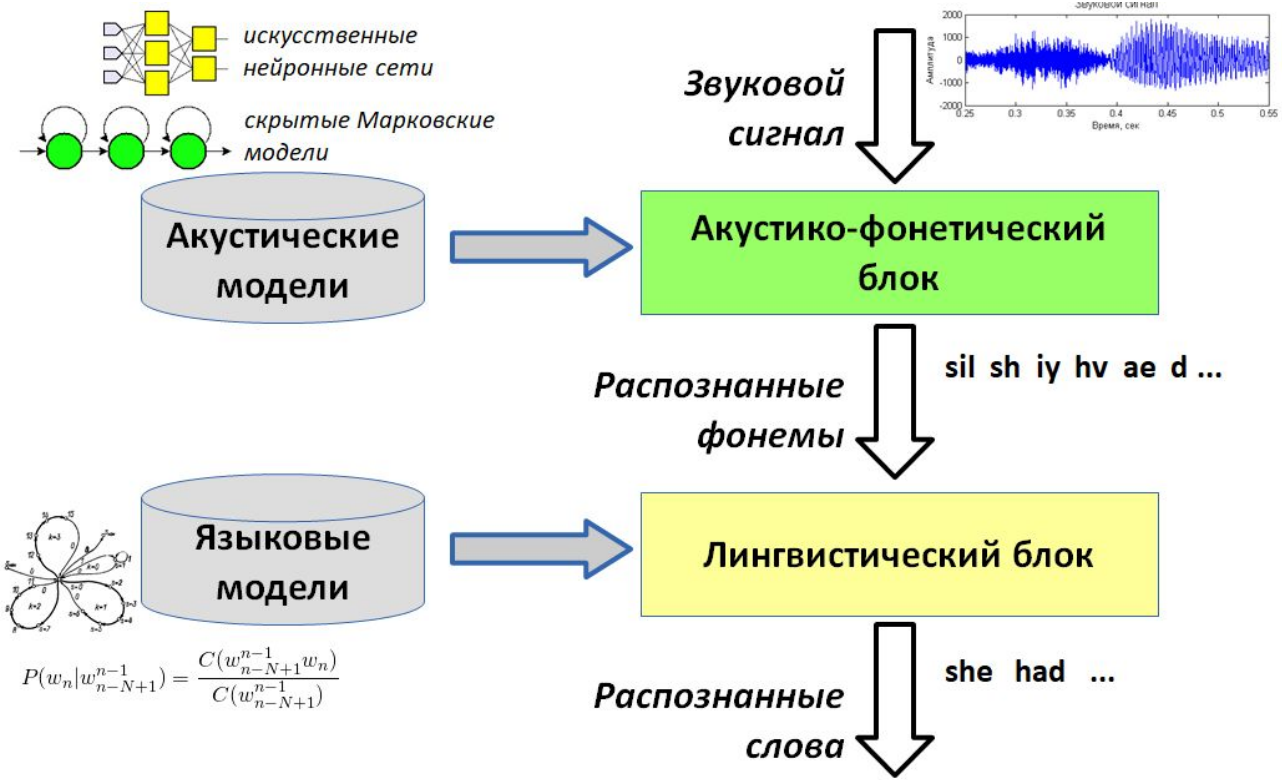
А если у меня специфичная задача?

- Голосовой чат-бот для выдачи кредита
- Голосовой чат-бот для заказа пиццы
- ...

Специфичной задаче - особые N-граммы!

1. Собираем особые тексты (кулинарные, финансовые…)
2. Строим особые N-граммы с помощью ngram-count
3. Объединяем их с базовыми N-граммами с помощью ngram-merge
   http://www.speech.sri.com/projects/srilm/manpages/ngram-merge.1.html

Классическая структура системы распознавания речи



Что такое хорошо?

Word error rate:

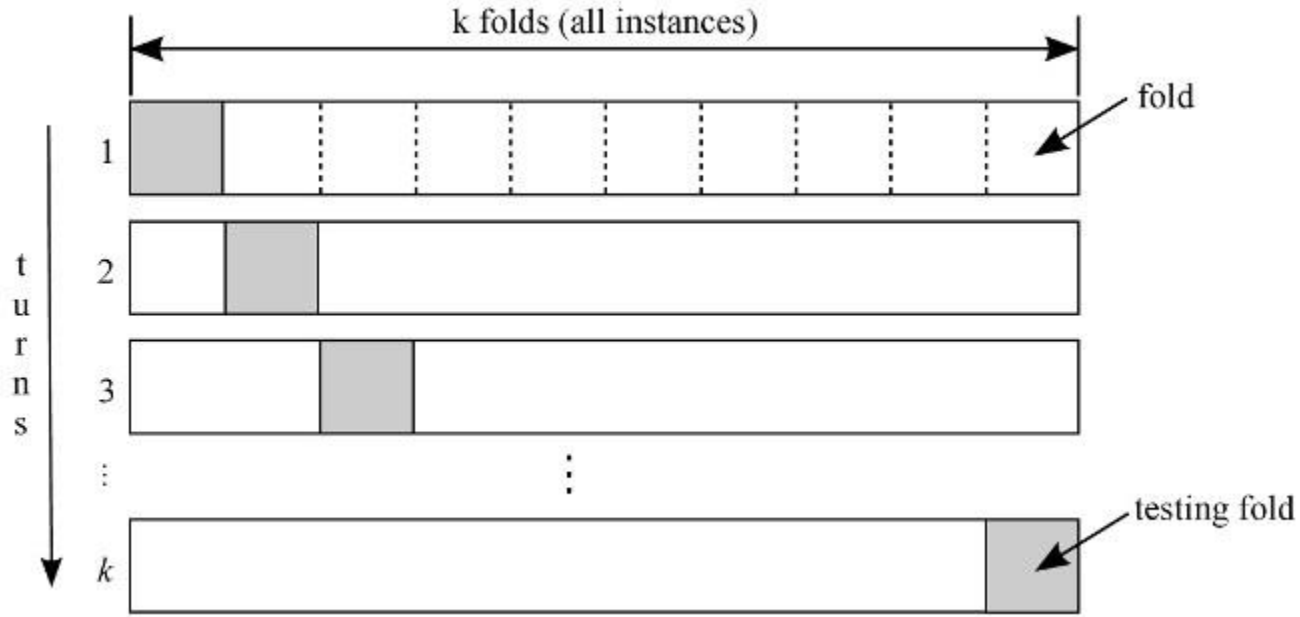
число вставок ***I***, замен ***S*** и удалений ***D***, необходимое для приведения распознанной фразы к эталонной фразе

$$WER = \frac{S + D + I}{N}$$

Какой WER мы хотим?

- голосовое командное управление
  - WER < 5%
- голосовой поиск
  - WER < 10%
- стенографирование звукозаписей
  - WER < 20-30%

Кроссвалидация



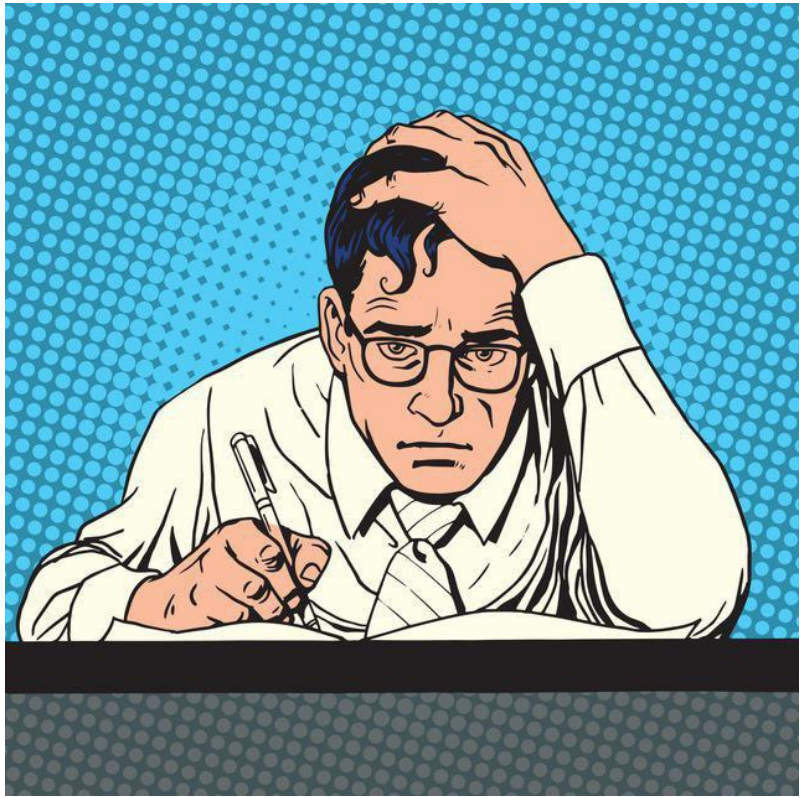
Кроссвалидация на Voxforge-Ru

https://github.com/nsu-ai/voxforge_ru_sphinx/tree/ruscorpora-ngrams

Experiment	1	2	3	4	5	6	7	8	9	10
WER, %	17,70	29,15	23,99	26,12	27,58	19,24	15,69	17,22	19,36	11,99

Итоговый WER = **20,80% ± 5,34%**

Что мы забыли?

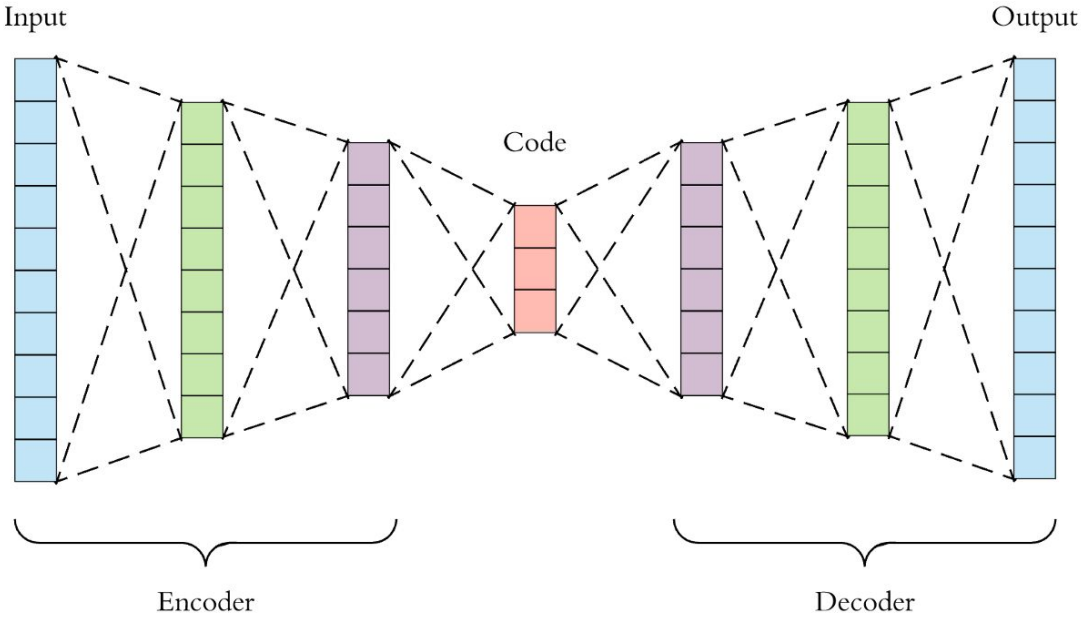


Как происходит интервью?

Шумоподавление!

Как подавить шумы?

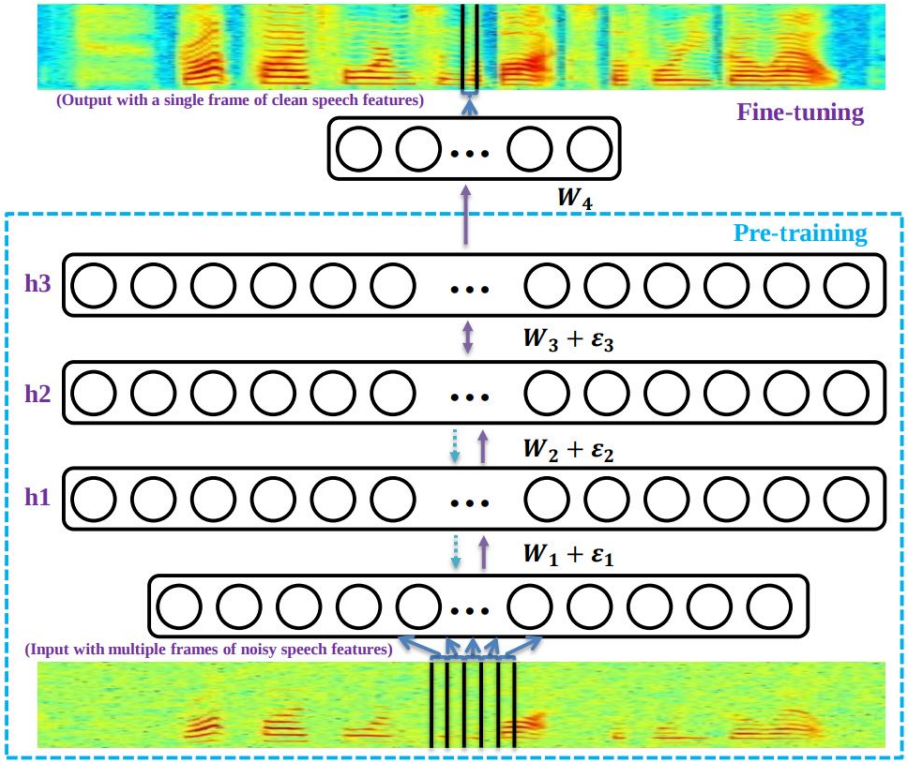
Глубокими нейронными сетями!



Регрессионная нейронная сеть

Yong Xu, Jun Du, Li-Rong Dai, and Chin-Hui Lee

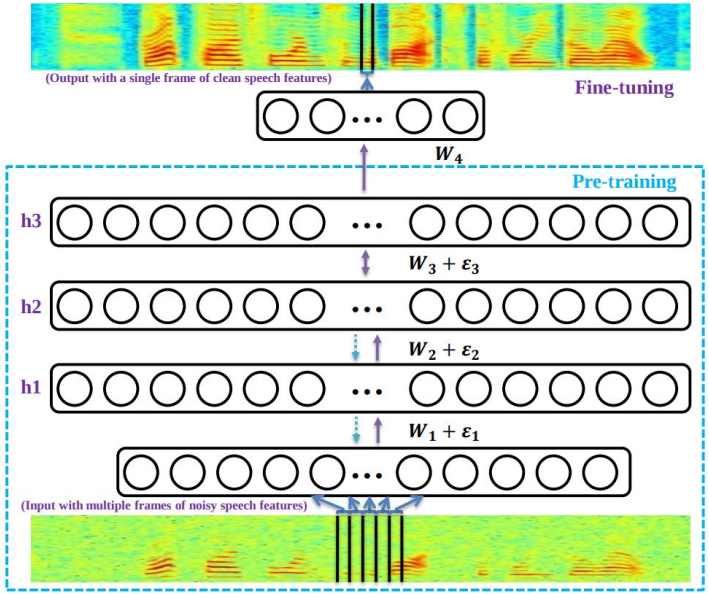
A Regression Approach to Speech Enhancement Based on Deep Neural Networks



Регрессионная нейронная сеть

Yong Xu, Jun Du, Li-Rong Dai, and Chin-Hui Lee

A Regression Approach to Speech Enhancement Based on Deep Neural Networks



SE-DNN: https://github.com/yongxuUSTC/sednn

Итоги

1. Распознавание речи можно сделать самому

Итоги

1. Распознавание речи можно сделать самому
2. Open Source - это круто!

Итоги

1. Распознавание речи можно сделать самому
2. Open Source - это круто!
3. **“Модульный подход”** VS “Нейросетевой End-to-end”

Спасибо моим ребятам!

Оля Яковенко, лингвист

Даниил Водолазский, математик

Маша Боровикова, лингвист

Спасибо вам за внимание!

Я открыт для диалога 😉

Иван Бондаренко

bond005@yandex.ru