Распознавание речи. Декодер

П. А. Холявин

p.kholyavin@spbu.ru

20.03.2024

Задача распознавания речи

Если $O = o_1, o_2, \ldots, o_n$ – звуковая последовательность,
$W = w_1, w_2, \ldots, w_n$ – последовательность слов, то

$$\hat{W} = \underset{W \in L}{argmax}\, P(W|O)$$

$$\hat{W} = \underset{W \in L}{argmax}\, \frac{P(O|W)P(W)}{P(O)} = \underset{W \in L}{argmax}\, P(O|W)P(W)$$

Задача декодера

модели звуков

↓

модели слов

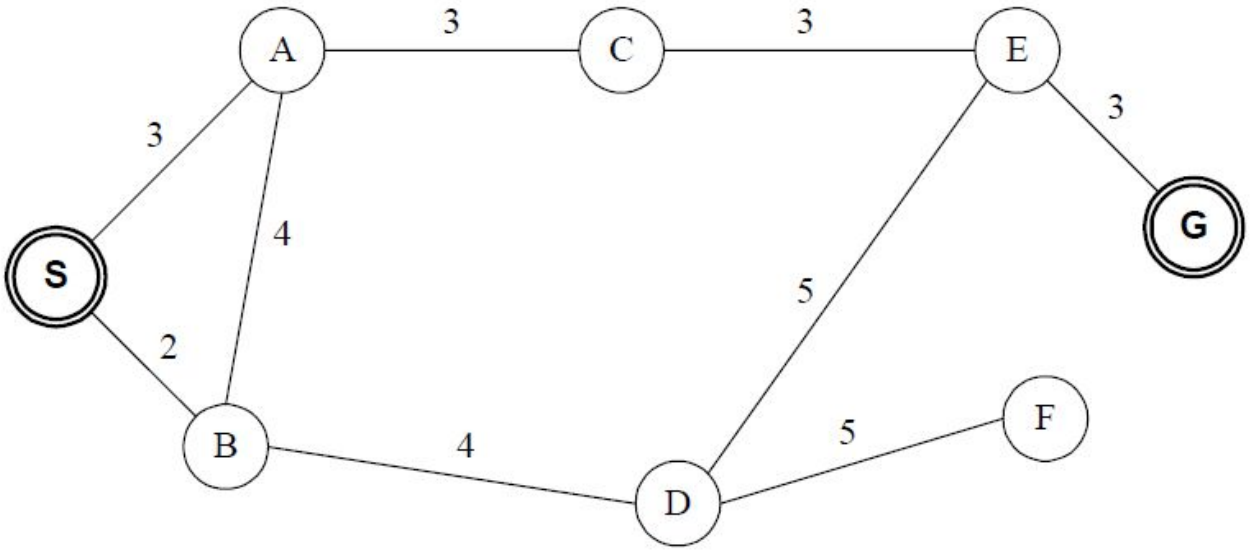
↓

сеть распознавания (lattice)

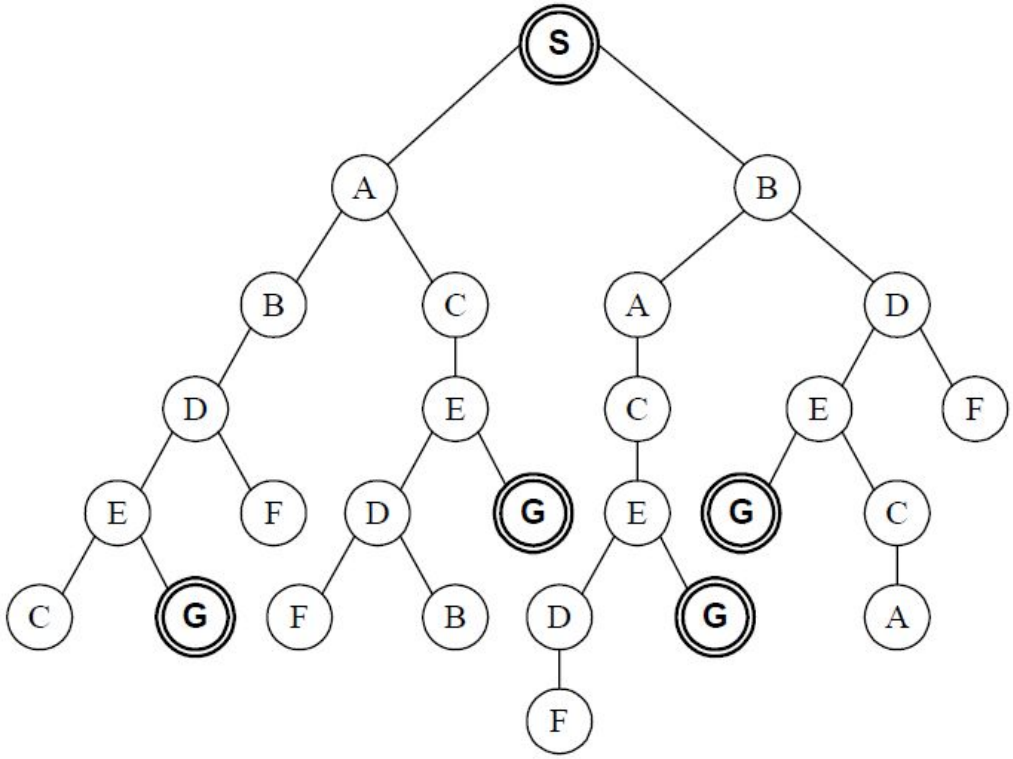
Декодер ищет наиболее вероятный путь

Поиск пути по графу

Задача коммивояжёра



Поиск пути по графу



Алгоритмы поиска:

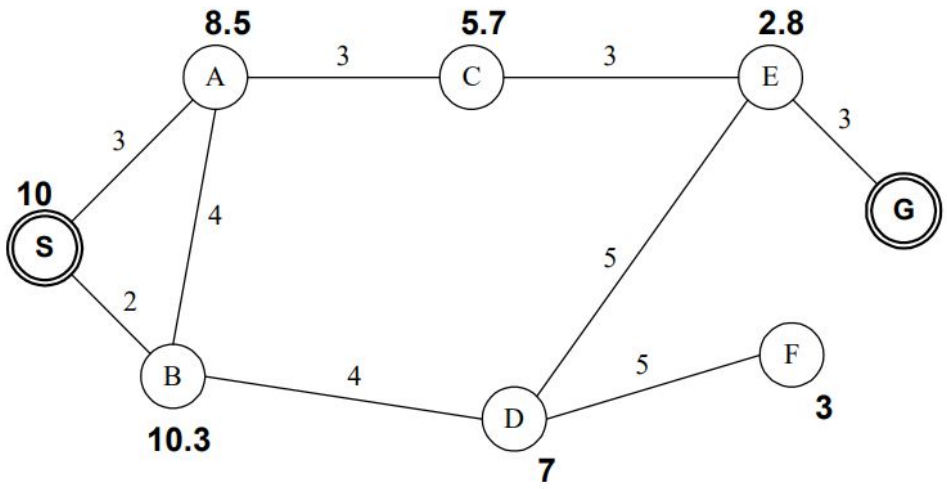
- в глубину
- в ширину
- эвристические (с функцией оценки текущей гипотезы):

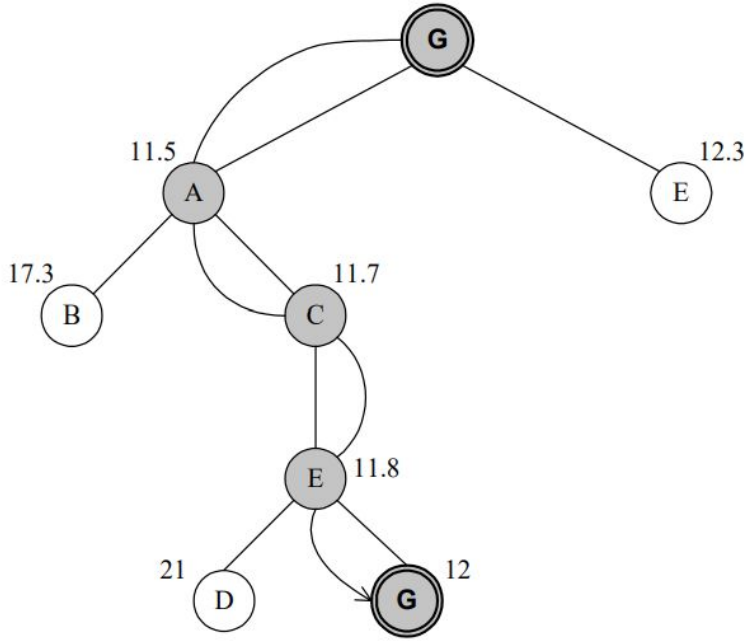
- по первому наилучшему совпадению
- лучевой поиск

Принцип ранней рекомбинации

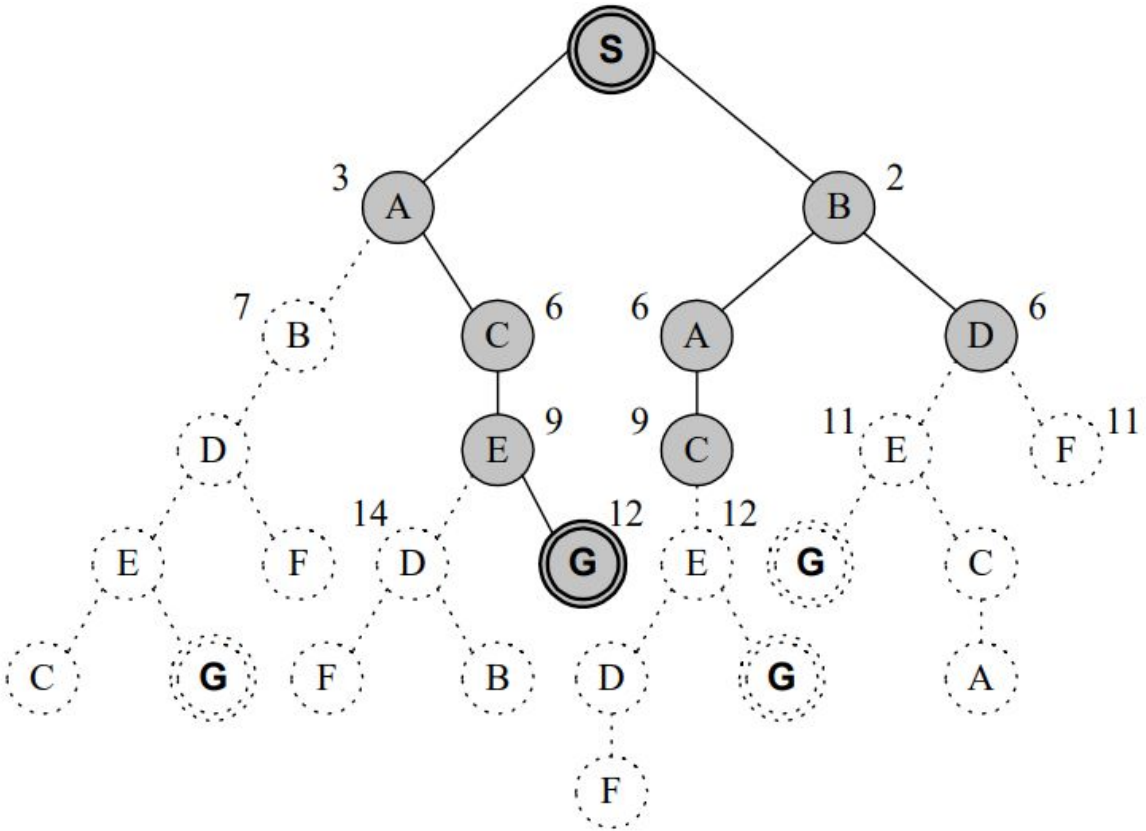
если несколько гипотез в сети имеют общий узел, следует оставить наилучшую гипотезу до этого узла и отбросить остальные, поскольку при дальнейшем развитии процесса у этих гипотез уже не будет возможности превзойти сохранённую

A* search


8.5
5.7
2.8
A
C
E
3
3
3
3
10
S
4
5
2
G
B
4
5
F
D
10.3
7
3


G
11.5
12.3
A
E
17.3
11.7
B
C
E
11.8
21
12
D
G

Beam search



Задача декодера

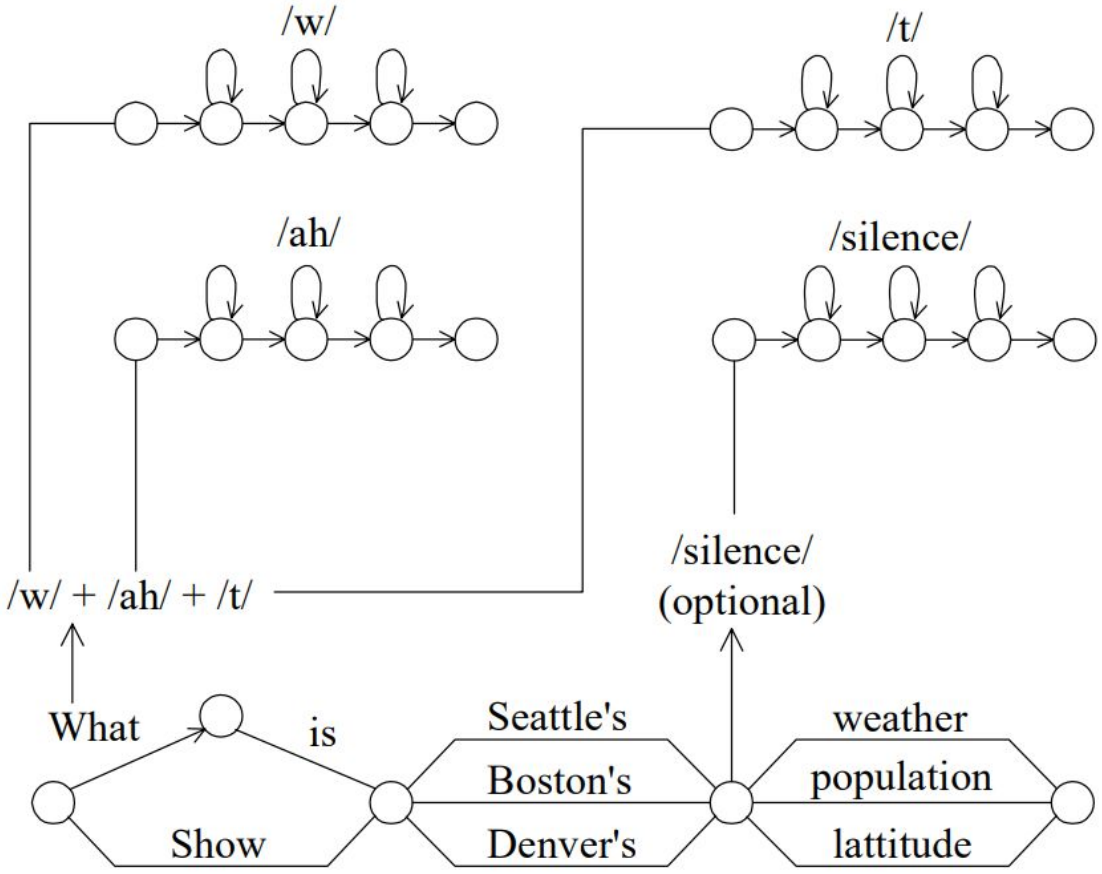
$$\hat{W} = \arg\max_{W} \{P(W) \cdot \sum_{s_1^T} P(x_1^T, s_1^T \mid w_1^N)\}$$

Аппроксимация Витерби

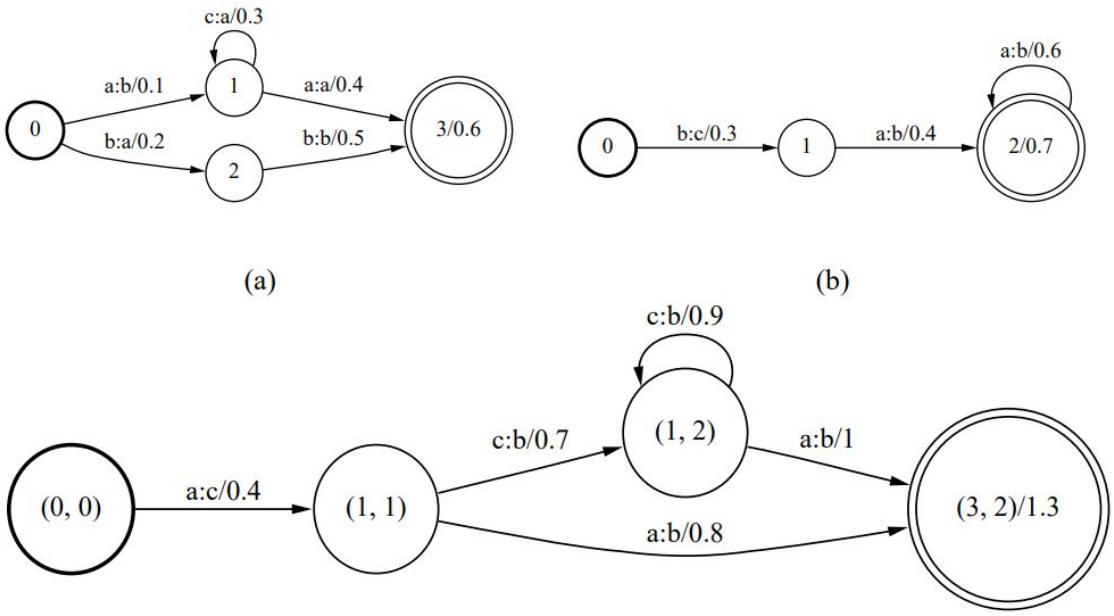
$$\hat{W} = \arg\max_{W} \{P(W)^{\alpha} \cdot \underset{s_1^T}{Max}[P(x_1^T, s_1^T \mid w_1^N)]\}$$

α – вес языковой модели

Объединение моделей



Композиция преобразователей



HCLG = H o C o L o G

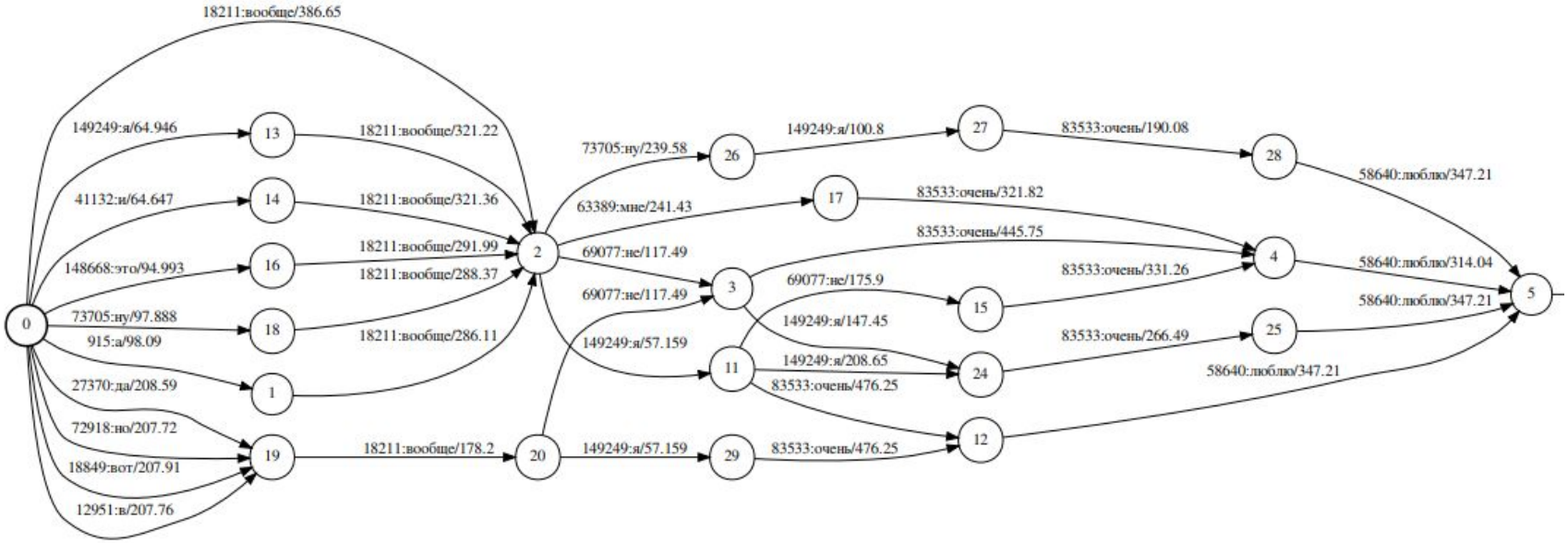
H: переходы между состояниями AM →контекстные аллофоны

C: контекстные аллофоны → звуки

L: звуки → слова

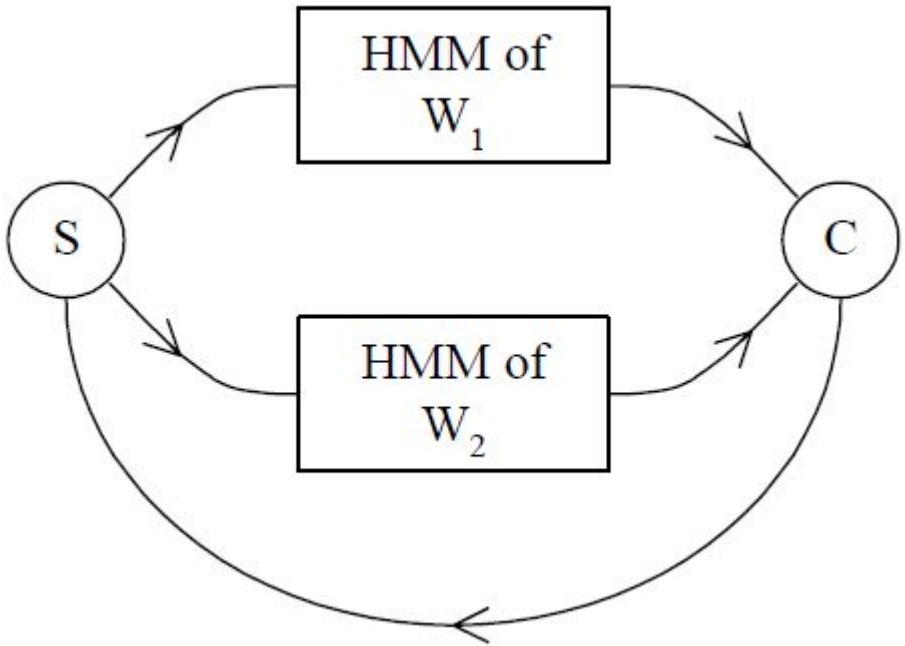
G: слова → слова

Сеть распознавания



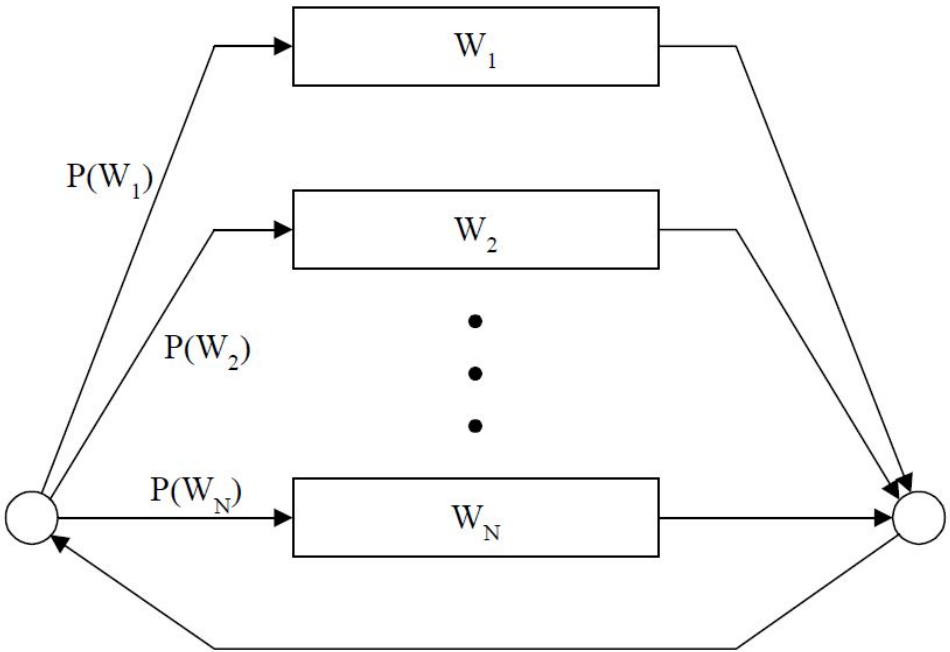
Распознавание слитной речи

Равновероятные униграммы



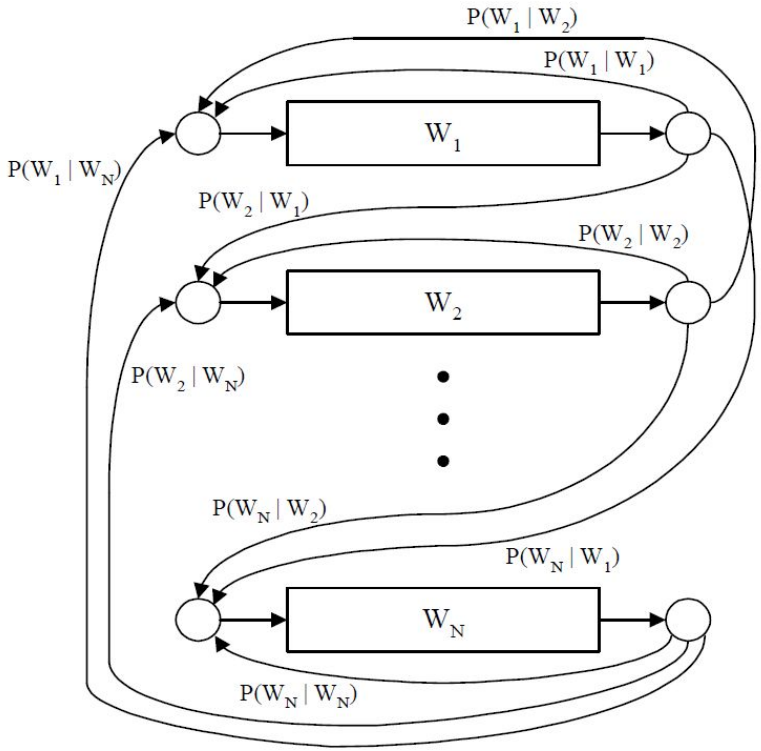
Распознавание слитной речи

Униграммы



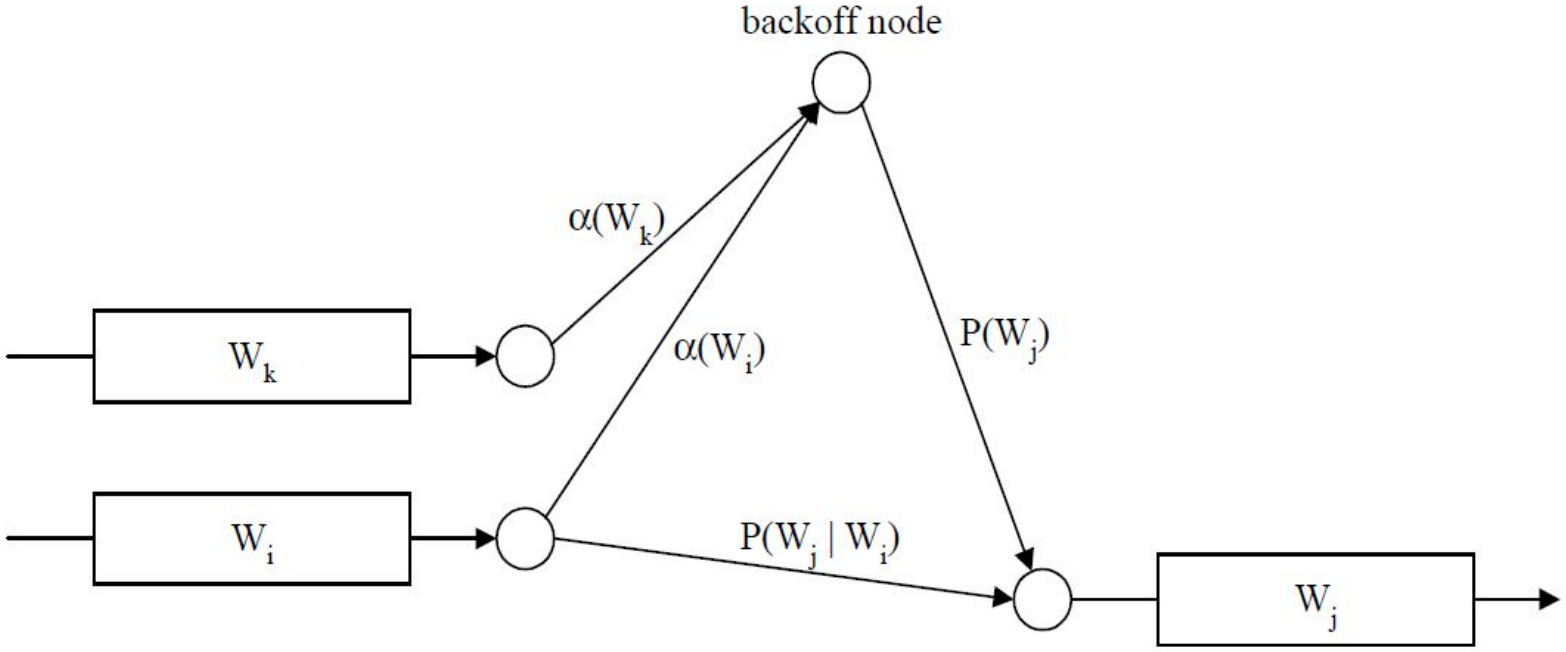
Распознавание слитной речи

Биграммы



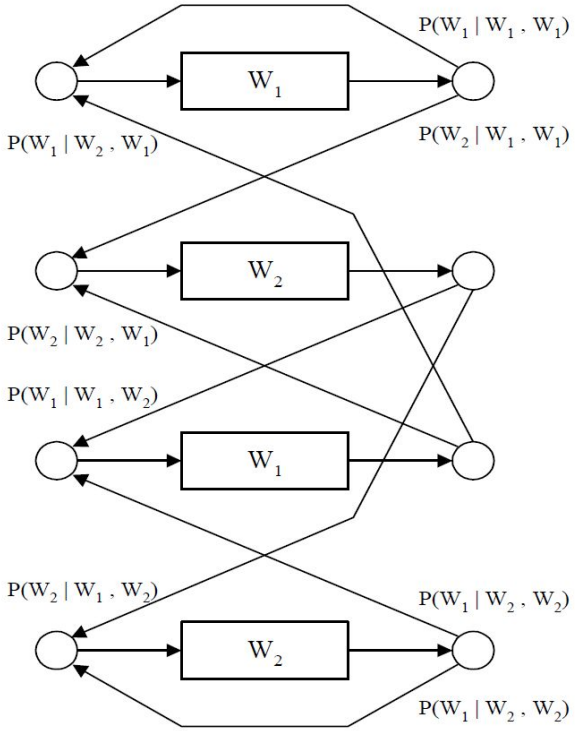
Распознавание слитной речи

Биграммы с откатом

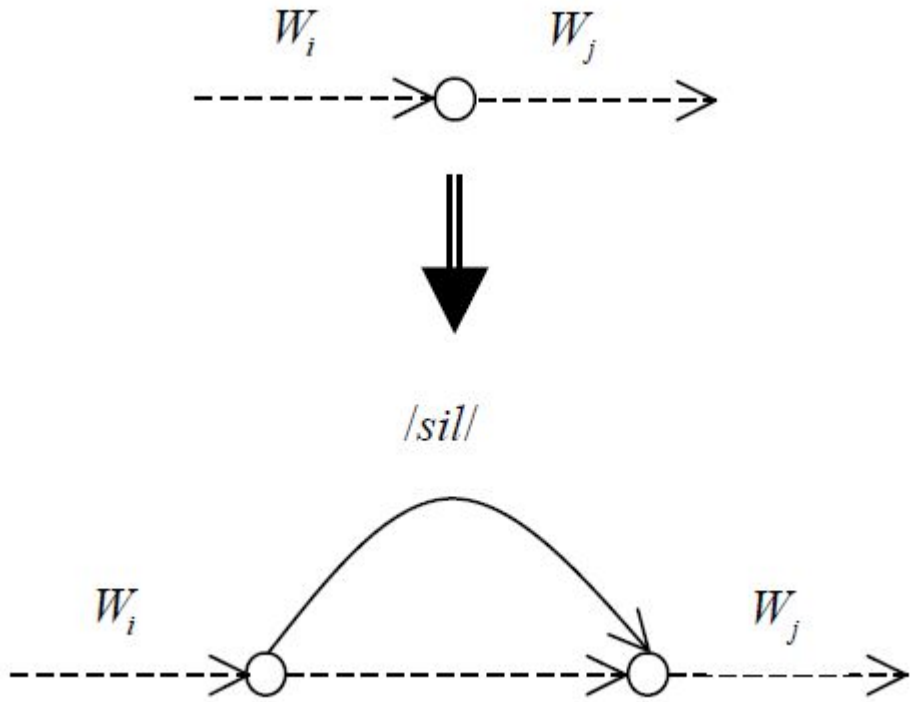


Распознавание слитной речи

Триграммы



Паузы



Лучевой поиск Витерби

ALGORITHM 12.6 TIME-SYNCHRONOUS VITERBI BEAM SEARCH

Initialization: For all the grammar word states w which can start a sentence,

$$D(0; I(w); w) = 0$$

$$h(0; I(w); w) = null$$

Induction: For time $t = 1$ to T do

For all active states do

Intra-word transitions according to Eq. (12.17) and (12.18)

$$D(t; s_t; w) = \min_{s_{t-1}} \{d(\mathbf{x}_t, s_t \mid s_{t-1}; w) + D(t-1; s_{t-1}; w)\}$$

$$h(t; s_t; w) = h(t-1, b_{\min}(t; s_t; w); w)$$

For all active word-final states do

Inter-word transitions according to Eq. (12.21), (12.22) and (12.23)

$$D(t; \eta; w) = \min_{v} \{\log P(w \mid v) + D(t; F(v); v)\}$$

$$h(t; \eta; w) = \langle v_{\min}, t \rangle :: h(t, F(v_{\min}); v_{\min})$$

if $D(t; \eta; w) < D(t; I(w); w)$

$$D(t; I(w); w) = D(t; \eta; w) \text{ and } h(t; I(w); w) = h(t; \eta; w)$$

Pruning: Find the cost for the best path and decide the beam threshold

Prune unpromising hypotheses

Termination: Pick the best path among all the possible final states of grammar at time T

Obtain the optimal word sequence according to the backtracking pointer $h(t; \eta; w)$

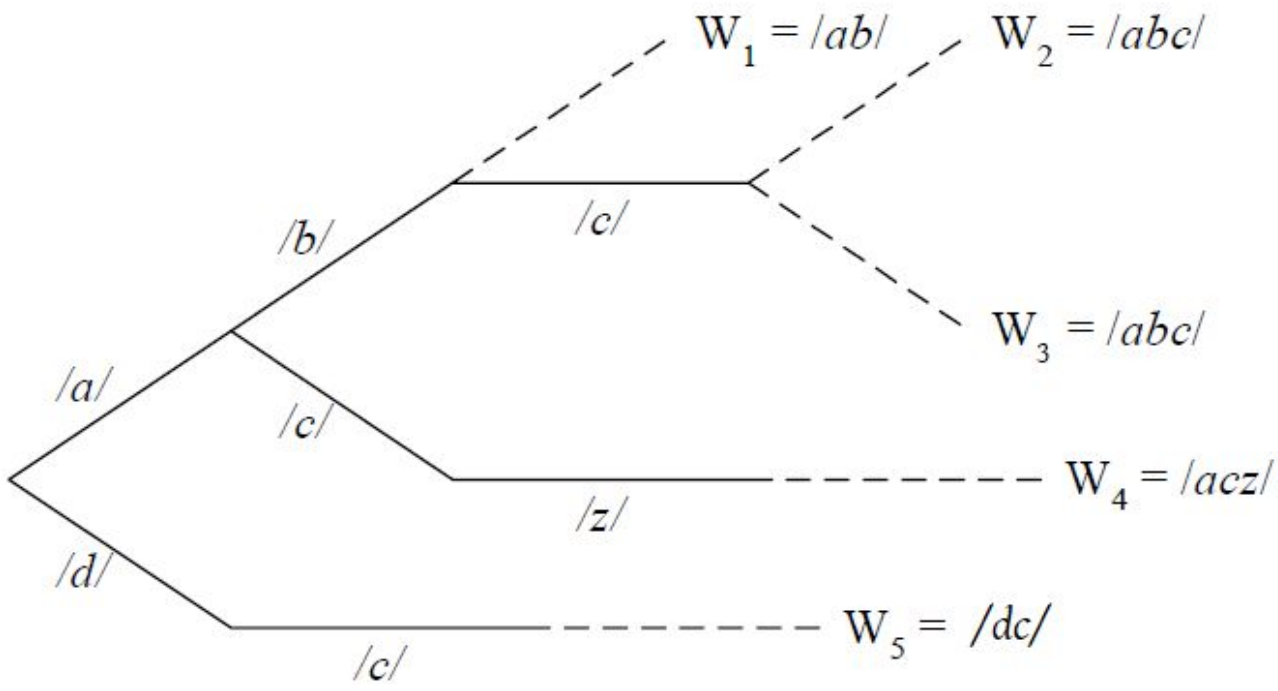
A*-поиск: выбор функции оценки

1. After the final training iteration, perform Viterbi forced alignment[9] with each training utterance to get an optimal time alignment for each word.
2. Randomly select an interval to cover the number of words ranging from two to ten. Denote this interval as $[i \ldots j]$
3. Compute the average acoustic cost per frame within this selected interval according to the following formula and save the value in a set Λ.

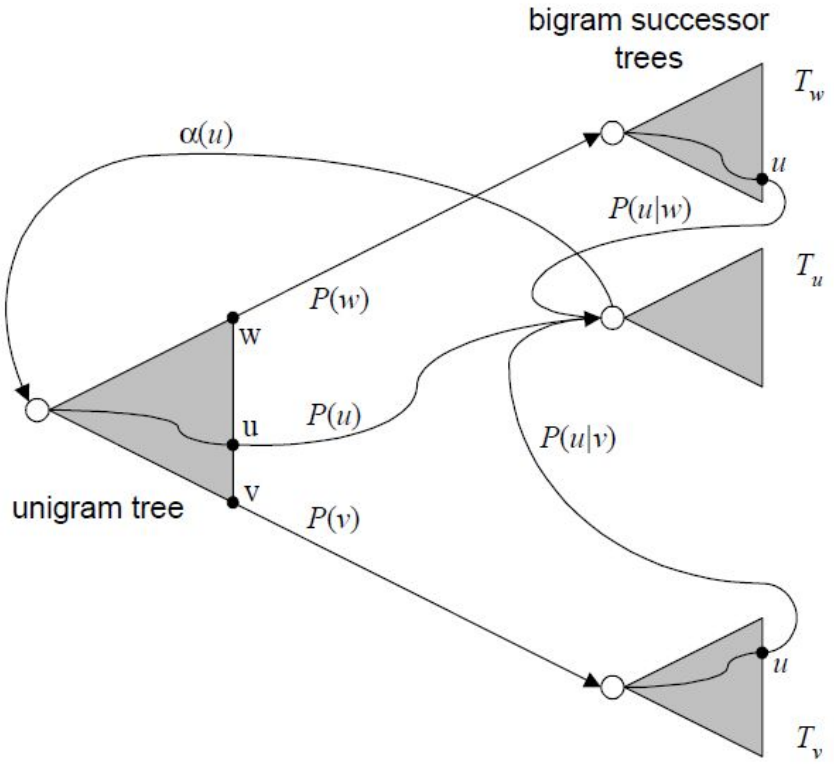
$$\frac{-1}{j-i} \log P(\mathbf{x}_i^j \mid \mathbf{w}_{i \ldots j}) \qquad (12.25)$$

   where $\mathbf{w}_{i \ldots j}$ is the word string corresponding to interval $[i \ldots j]$
4. Repeat Steps 2 and 3 for the entire training set.
5. Define $\psi_{\min}$ and ψ_{avg} as the minimum and average value found in set Λ.

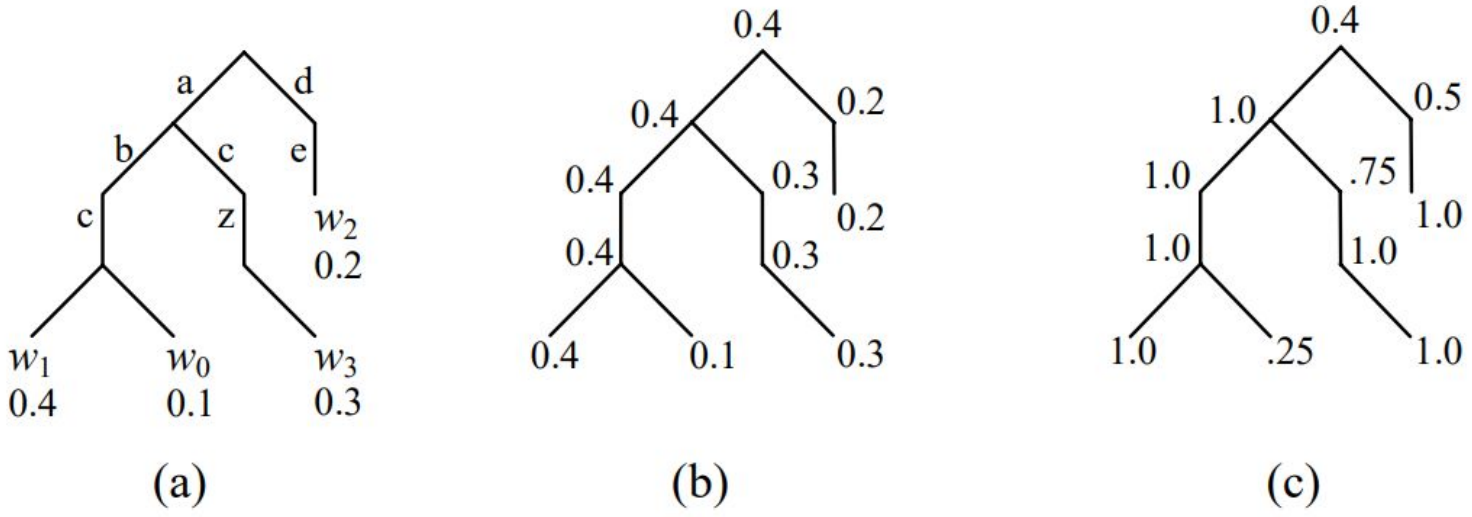
Представление лексикона как дерева



Оптимизация для N-грамм



Факторизация дерева



Проблема внесловарных слов

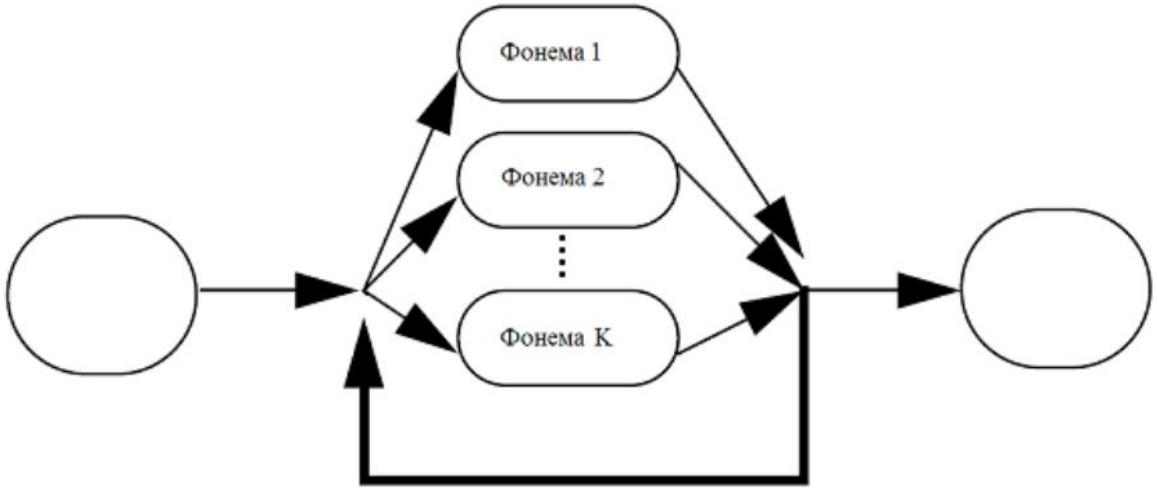
OOV – out of vocabulary

1. Определение наличия OOV-слова и его положения

2. Определение его «фонемного» состава

Способы решения:
а) модели заполнения (общие модели слова)
б) фиксированные сочетания фонем

Общая модель слова



Темы для докладов

1. Wavelet-преобразование
2. Настройка системы на диктора
3. Устойчивость систем APP к шуму
4. LSTM в APP
5. Трансформеры в APP
6. Wav2vec: общая архитектура
7. Долговременные (TRAP, TempoRAL Patterns) признаки
8. CRF в APP
9. HTK Speech Recognition Toolkit
10. Kaldi ASR
11. CMU Sphinx
12. Частные случаи APP: детская речь, патологическая речь, ...
13. Сбор данных для обучения системы APP
14. Computer-aided pronunciation training
15. Любая статья Interspeech по APP
16. Преобразования признаков: LDA, PCA, DMC
17. Постобработка текста в APP
18. Другая тема, интересующая лично вас

Спасибо за внимание!