

1. Для начала поэкспериментируем с разным количеством лучей на обычном beam_search

```
=====
Results | beam_width=5 | distances = [10, 5, 5, 8, 2, 16, 19, 15], avg_distance = 10.0
=====
Results | beam_width=10 | distances = [10, 5, 5, 8, 2, 17, 19, 15], avg_distance = 10.125
=====
```

Судя по результатам, при увеличении количества лучей страдает качество. При анализе кода так и не понял, в чем проблема – все таки топ 1 должен быть одинаковым и в топ 5 и в топ 10. Но код при этом выглядит логично.

Предполагаю, что при поиске в топ 10 на последующих шагах всплывает более вероятная последовательность с точки зрения модели (что впрочем не является более валидным результатом).

2. Следующий эксперимент с разными гиперпараметрами с перевзвешиванием LM

```
=====
Results |  $\alpha=0.5$ ,  $\beta=0.5$ , beam_width=5 | distances = [10, 5, 5, 8, 2, 16, 19, 15], avg_distance = 10.0
=====
Results |  $\alpha=0.5$ ,  $\beta=0.5$ , beam_width=10 | distances = [10, 5, 5, 8, 2, 17, 19, 15], avg_distance = 10.125
=====
Results |  $\alpha=0.5$ ,  $\beta=5$ , beam_width=5 | distances = [10, 5, 5, 8, 2, 15, 19, 15], avg_distance = 9.875
=====
Results |  $\alpha=0.5$ ,  $\beta=5$ , beam_width=10 | distances = [10, 5, 5, 8, 2, 15, 19, 15], avg_distance = 9.875
=====
Results |  $\alpha=1.0$ ,  $\beta=0.5$ , beam_width=5 | distances = [10, 5, 5, 8, 2, 16, 19, 15], avg_distance = 10.0
=====
Results |  $\alpha=1.0$ ,  $\beta=0.5$ , beam_width=10 | distances = [10, 5, 5, 8, 2, 17, 19, 15], avg_distance = 10.125
=====
Results |  $\alpha=1.0$ ,  $\beta=5$ , beam_width=5 | distances = [10, 5, 5, 8, 2, 16, 19, 15], avg_distance = 10.0
=====
Results |  $\alpha=1.0$ ,  $\beta=5$ , beam_width=10 | distances = [10, 5, 5, 8, 2, 15, 19, 15], avg_distance = 9.875
=====
Results |  $\alpha=3$ ,  $\beta=0.5$ , beam_width=5 | distances = [10, 5, 5, 8, 2, 16, 19, 15], avg_distance = 10.0
=====
Results |  $\alpha=3$ ,  $\beta=0.5$ , beam_width=10 | distances = [10, 5, 5, 8, 2, 17, 19, 15], avg_distance = 10.125
=====
Results |  $\alpha=3$ ,  $\beta=5$ , beam_width=5 | distances = [10, 5, 5, 8, 2, 16, 19, 15], avg_distance = 10.0
=====
Results |  $\alpha=3$ ,  $\beta=5$ , beam_width=10 | distances = [10, 5, 5, 8, 2, 17, 19, 15], avg_distance = 10.125
=====
```

Влияние LM видно только при высоком beta. Т.к. тут перевзвешивание проходит уже на лучших выходах после beam_search, влияние все равно не особо сильное, но там где beta высокий – результаты лучше

Примечание: в первых экспериментах коэффициенты брал совсем небольшие, поэтому влияния не было видно совсем.

3. Чего не скажешь о следующем эксперименте, где LM помогает перевзвесить выход до выбора top лучей:

```

=====
Results |  $\alpha=0.5$ ,  $\beta=0.5$ , beam_width=5 | distances = [11, 6, 6, 8, 2, 20, 25, 20], avg_distance = 12.25
=====
Results |  $\alpha=0.5$ ,  $\beta=0.5$ , beam_width=10 | distances = [12, 9, 6, 7, 2, 20, 25, 21], avg_distance = 12.75
=====
Results |  $\alpha=0.5$ ,  $\beta=5$ , beam_width=5 | distances = [11, 11, 7, 12, 3, 20, 17, 17], avg_distance = 12.25
=====
Results |  $\alpha=0.5$ ,  $\beta=5$ , beam_width=10 | distances = [15, 36, 8, 12, 3, 25, 17, 25], avg_distance = 17.625
=====
Results |  $\alpha=1.0$ ,  $\beta=0.5$ , beam_width=5 | distances = [18, 20, 19, 15, 7, 23, 24, 24], avg_distance = 18.75
=====
Results |  $\alpha=1.0$ ,  $\beta=0.5$ , beam_width=10 | distances = [31, 22, 28, 16, 8, 35, 33, 26], avg_distance = 24.875
=====
Results |  $\alpha=1.0$ ,  $\beta=5$ , beam_width=5 | distances = [13, 12, 10, 12, 5, 17, 21, 21], avg_distance = 13.875
=====
Results |  $\alpha=1.0$ ,  $\beta=5$ , beam_width=10 | distances = [13, 31, 10, 15, 5, 17, 21, 25], avg_distance = 17.125
=====
Results |  $\alpha=3$ ,  $\beta=0.5$ , beam_width=5 | distances = [83, 153, 104, 92, 89, 45, 68, 75], avg_distance = 88.625
=====
Results |  $\alpha=3$ ,  $\beta=0.5$ , beam_width=10 | distances = [138, 206, 141, 113, 153, 55, 68, 63], avg_distance = 117.125
=====
Results |  $\alpha=3$ ,  $\beta=5$ , beam_width=5 | distances = [51, 75, 70, 38, 55, 25, 50, 53], avg_distance = 52.125
=====
Results |  $\alpha=3$ ,  $\beta=5$ , beam_width=10 | distances = [43, 106, 94, 45, 37, 51, 44, 56], avg_distance = 59.5
=====

```

Здесь заметно влияние LM, которая при больших коэффициентах α (влияние скоров модели) сильно портит результат. Лучше всего работает малое количество лучей и отсутствие вклада модели. Возможно тут как раз стоит посмотреть, как себя ведет LM побольше и получше.

4. Чем и займемся. Посмотрим как себя поведет модель с большим количеством параметров "lm/4-gram.arpa" на перевзвешивании после beam_search

```

=====
Results |  $\alpha=0.5$ ,  $\beta=0.5$ , beam_width=5 | distances = [10, 5, 5, 8, 2, 16, 19, 15], avg_distance = 10.0
=====
Results |  $\alpha=0.5$ ,  $\beta=0.5$ , beam_width=10 | distances = [10, 5, 5, 8, 2, 17, 19, 15], avg_distance = 10.125
=====
Results |  $\alpha=0.5$ ,  $\beta=5$ , beam_width=5 | distances = [10, 5, 5, 8, 2, 15, 19, 15], avg_distance = 9.875
=====
Results |  $\alpha=0.5$ ,  $\beta=5$ , beam_width=10 | distances = [10, 5, 5, 8, 2, 15, 19, 15], avg_distance = 9.875
=====
Results |  $\alpha=1.0$ ,  $\beta=0.5$ , beam_width=5 | distances = [10, 5, 5, 8, 2, 16, 19, 15], avg_distance = 10.0
=====
Results |  $\alpha=1.0$ ,  $\beta=0.5$ , beam_width=10 | distances = [10, 5, 5, 8, 2, 17, 19, 15], avg_distance = 10.125
=====
Results |  $\alpha=1.0$ ,  $\beta=5$ , beam_width=5 | distances = [10, 5, 5, 8, 2, 16, 19, 15], avg_distance = 10.0
=====
Results |  $\alpha=1.0$ ,  $\beta=5$ , beam_width=10 | distances = [10, 5, 5, 8, 2, 15, 19, 15], avg_distance = 9.875
=====
Results |  $\alpha=3$ ,  $\beta=0.5$ , beam_width=5 | distances = [10, 5, 5, 8, 2, 16, 19, 15], avg_distance = 10.0
=====
Results |  $\alpha=3$ ,  $\beta=0.5$ , beam_width=10 | distances = [10, 5, 5, 8, 2, 17, 19, 15], avg_distance = 10.125
=====
Results |  $\alpha=3$ ,  $\beta=5$ , beam_width=5 | distances = [10, 5, 5, 8, 2, 16, 19, 15], avg_distance = 10.0
=====
Results |  $\alpha=3$ ,  $\beta=5$ , beam_width=10 | distances = [10, 5, 5, 8, 2, 17, 19, 15], avg_distance = 10.125
=====

```

Видимо это все еще недостаточно хорошая модель, потому что увеличение влияния ее выходов ухудшает результат, но зато влияние длины предложения (коэффициент β) сказывается хорошо.

5. Ну и последний эксперимент когда LM помогает предсказывать данные до выбора top вероятностей

```

=====
Results |  $\alpha=0.5$ ,  $\beta=0.5$ , beam_width=5 | distances = [12, 7, 6, 8, 2, 20, 25, 21], avg_distance = 12.625
=====
Results |  $\alpha=0.5$ ,  $\beta=0.5$ , beam_width=10 | distances = [12, 10, 8, 10, 2, 20, 26, 21], avg_distance = 13.625
=====
Results |  $\alpha=0.5$ ,  $\beta=5$ , beam_width=5 | distances = [11, 10, 7, 12, 3, 21, 20, 18], avg_distance = 12.75
=====
Results |  $\alpha=0.5$ ,  $\beta=5$ , beam_width=10 | distances = [11, 18, 15, 13, 3, 20, 23, 26], avg_distance = 16.125
=====
Results |  $\alpha=1.0$ ,  $\beta=0.5$ , beam_width=5 | distances = [21, 21, 26, 16, 8, 26, 29, 22], avg_distance = 21.125
=====
Results |  $\alpha=1.0$ ,  $\beta=0.5$ , beam_width=10 | distances = [21, 21, 43, 22, 9, 47, 37, 23], avg_distance = 27.875
=====
Results |  $\alpha=1.0$ ,  $\beta=5$ , beam_width=5 | distances = [15, 20, 9, 15, 6, 17, 22, 21], avg_distance = 15.625
=====
Results |  $\alpha=1.0$ ,  $\beta=5$ , beam_width=10 | distances = [15, 19, 11, 16, 6, 24, 24, 21], avg_distance = 17.0
=====
Results |  $\alpha=3$ ,  $\beta=0.5$ , beam_width=5 | distances = [81, 148, 97, 98, 104, 43, 63, 58], avg_distance = 86.5
=====
Results |  $\alpha=3$ ,  $\beta=0.5$ , beam_width=10 | distances = [132, 190, 116, 134, 150, 55, 68, 58], avg_distance = 112.875
=====
Results |  $\alpha=3$ ,  $\beta=5$ , beam_width=5 | distances = [39, 77, 60, 54, 44, 34, 49, 57], avg_distance = 51.75
=====
Results |  $\alpha=3$ ,  $\beta=5$ , beam_width=10 | distances = [51, 108, 76, 65, 40, 51, 51, 60], avg_distance = 62.75
=====

```

Она не помогает (хотя результаты чуточку лучше, чем у малой LM).

В сухом остатке wav2vec работает хорошо сама по себе (по крайней мере на тестовых примерах) и единственное, что помогло улучшить результат – это увеличение сора для более длинных последовательностей.

Тем не менее, тестовая выборка не то, чтобы репрезентативная, поэтому как универсальный рецепт успеха этот алгоритм принимать нельзя. Стоит экспериментировать.