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.

2. Следующий эксперимент с разными гиперпараметрами с перевзвешиванием LM

```
================================================================================
Results | α=0.5, β=0.5, beam_width=5 | distances = [10, 5, 5, 8, 2, 16, 19, 15], avg_distance = 10.0
================================================================================
Results | α=0.5, β=0.5, beam_width=10 | distances = [10, 5, 5, 8, 2, 17, 19, 15], avg_distance = 10.125
================================================================================
Results | α=0.5, β=5, beam_width=5 | distances = [10, 5, 5, 8, 2, 15, 19, 15], avg_distance = 9.875
================================================================================
Results | α=0.5, β=5, beam_width=10 | distances = [10, 5, 5, 8, 2, 15, 19, 15], avg_distance = 9.875
================================================================================
Results | α=1.0, β=0.5, beam_width=5 | distances = [10, 5, 5, 8, 2, 16, 19, 15], avg_distance = 10.0
================================================================================
Results | α=1.0, β=0.5, beam_width=10 | distances = [10, 5, 5, 8, 2, 17, 19, 15], avg_distance = 10.125
================================================================================
Results | α=1.0, β=5, beam_width=5 | distances = [10, 5, 5, 8, 2, 16, 19, 15], avg_distance = 10.0
================================================================================
Results | α=1.0, β=5, beam_width=10 | distances = [10, 5, 5, 8, 2, 15, 19, 15], avg_distance = 9.875
================================================================================
Results | α=3, β=0.5, beam_width=5 | distances = [10, 5, 5, 8, 2, 16, 19, 15], avg_distance = 10.0
================================================================================
Results | α=3, β=0.5, beam_width=10 | distances = [10, 5, 5, 8, 2, 17, 19, 15], avg_distance = 10.125
================================================================================
Results | α=3, β=5, beam_width=5 | distances = [10, 5, 5, 8, 2, 16, 19, 15], avg_distance = 10.0
================================================================================
Results | α=3, β=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 эксперменте, где LM помогает перевзвесить выход до выбора top лучей:

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

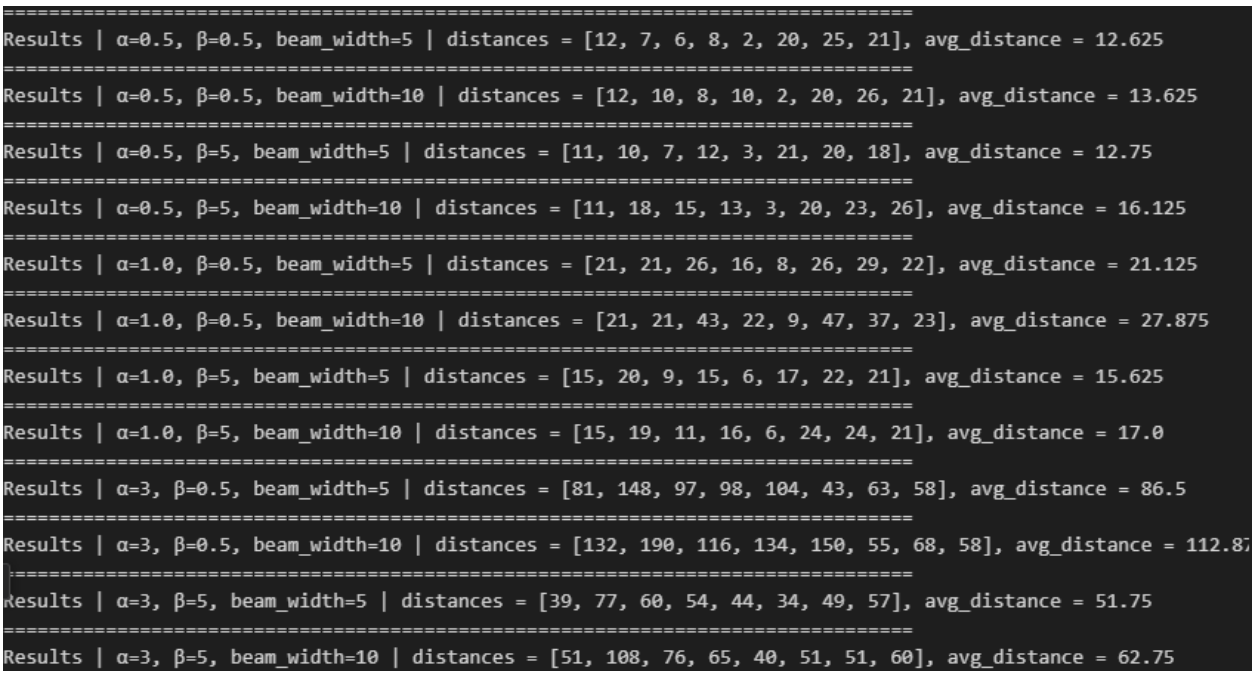
Здесь заметно влияние LM, которая при больших коэффициенте alpa (влияние скоров модели) сильно портит результат. Лучше всего работает малое количество лучей и отсутствие вклада модели. Возможно тут как раз стоит смотреть, как себя ведет LM побольше и получше.

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

```
================================================================================
Results | α=0.5, β=0.5, beam_width=5 | distances = [10, 5, 5, 8, 2, 16, 19, 15], avg_distance = 10.0
================================================================================
Results | α=0.5, β=0.5, beam_width=10 | distances = [10, 5, 5, 8, 2, 17, 19, 15], avg_distance = 10.125
================================================================================
Results | α=0.5, β=5, beam_width=5 | distances = [10, 5, 5, 8, 2, 15, 19, 15], avg_distance = 9.875
================================================================================
Results | α=0.5, β=5, beam_width=10 | distances = [10, 5, 5, 8, 2, 15, 19, 15], avg_distance = 9.875
================================================================================
Results | α=1.0, β=0.5, beam_width=5 | distances = [10, 5, 5, 8, 2, 16, 19, 15], avg_distance = 10.0
================================================================================
Results | α=1.0, β=0.5, beam_width=10 | distances = [10, 5, 5, 8, 2, 17, 19, 15], avg_distance = 10.125
================================================================================
Results | α=1.0, β=5, beam_width=5 | distances = [10, 5, 5, 8, 2, 16, 19, 15], avg_distance = 10.0
================================================================================
Results | α=1.0, β=5, beam_width=10 | distances = [10, 5, 5, 8, 2, 15, 19, 15], avg_distance = 9.875
================================================================================
Results | α=3, β=0.5, beam_width=5 | distances = [10, 5, 5, 8, 2, 16, 19, 15], avg_distance = 10.0
================================================================================
Results | α=3, β=0.5, beam_width=10 | distances = [10, 5, 5, 8, 2, 17, 19, 15], avg_distance = 10.125
================================================================================
Results | α=3, β=5, beam_width=5 | distances = [10, 5, 5, 8, 2, 16, 19, 15], avg_distance = 10.0
================================================================================
Results | α=3, β=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 beta) сказывается хорошо.

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



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 скора для более длинных последовательностей.

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