Su Zhang
Latent Dirichlet Allocation Explanations

Show the inferred beta vectors and indicate how they map to the true topics above.

The true topics distribution (beta) is below figure:

```
vocabulary = [
    "bass",
    "pike",
    "deep",
    "tuba",
    "horn",
    "catapult",
]
beta = np.array(
    [
        [0.4, 0.4, 0.2, 0.0, 0.0, 0.0],
        [0.0, 0.3, 0.1, 0.0, 0.3, 0.3],
        [0.3, 0.0, 0.2, 0.3, 0.2, 0.0],
    ]
)
```

The inferred beta output is below:

```
Topic 0:
bass: 0.2972
pike: 0.2451
deep: 0.1667
tuba: 0.1074
horn: 0.1017
catapult: 0.0820
Topic 1:
bass: 0.2274
pike: 0.1737
deep: 0.2284
tuba: 0.1655
horn: 0.1678
catapult: 0.0371
Topic 2:
bass: 0.2091
pike: 0.2780
deep: 0.1240
tuba: 0.0800
horn: 0.1835
catapult: 0.1254
```

As the training process is unsupervised learning, the topic orders of inferred beta are not necessarily the same with the topic orders of input beta. From the above specific output, we could tell that the inferred beta's topic 0 matched relatively well with the input beta's topic 0 (first row) – with "bass" and "pike" being the highest frequency, and "tube", "horn", and "catapult" being the lowest frequency.

For the inferred beta's topic 1, it seemed that the best match is the input beta's topic 2 (third row). However, "pike" should have lower frequency and "tuba" should have higher frequency, and thus not a complete match.

For the inferred beta's topic 2, it seemed that the best match is the input beta's topic 1 (second row). However, "bass" should have lower frequency and "horn" and "catapult" should have higher frequency, and thus not a complete match.

I think the key reasons for the mismatch between input beta and inferred beta are because the corpus is relatively small and training data is limited, leading to limited accuracy of output. If we could enlarge the training data, I believe the trained output would be closer to the true topics distribution.