This notebook displays visualizations for subsequent runs of a LDA training on the same corpus of data.
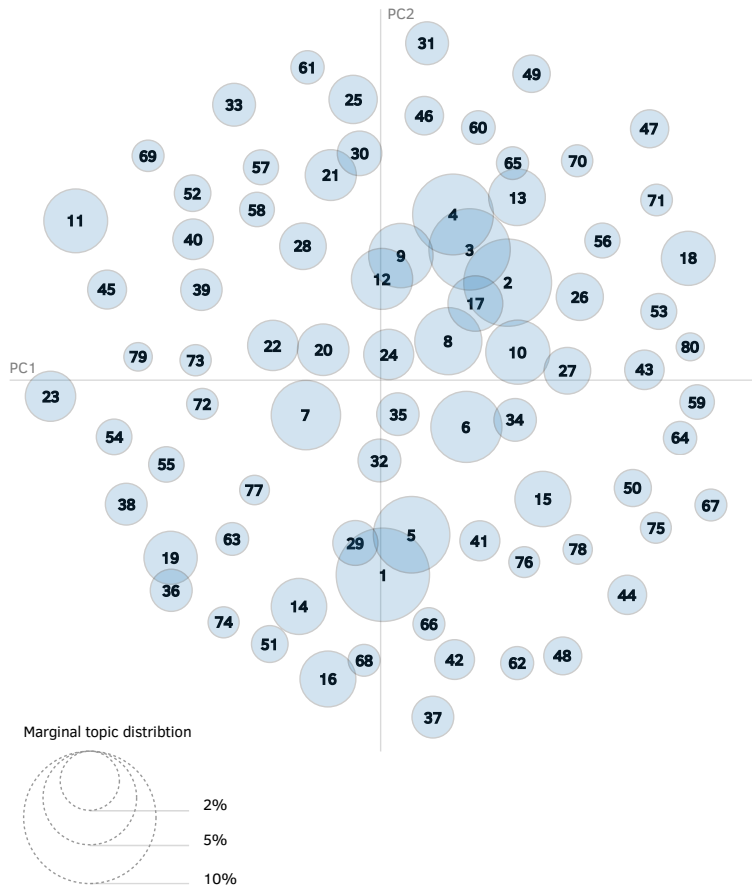
In [5]:
```python
from IPython.core.display import display, HTML
import ktm_prepviz, pyLDAvis
vis = ktm_prepviz.prepviz("topic_number")
```

In [6]:
```
display(HTML("<h1>20 topics</h1>"))
pyLDAvis.display(vis[0])
```
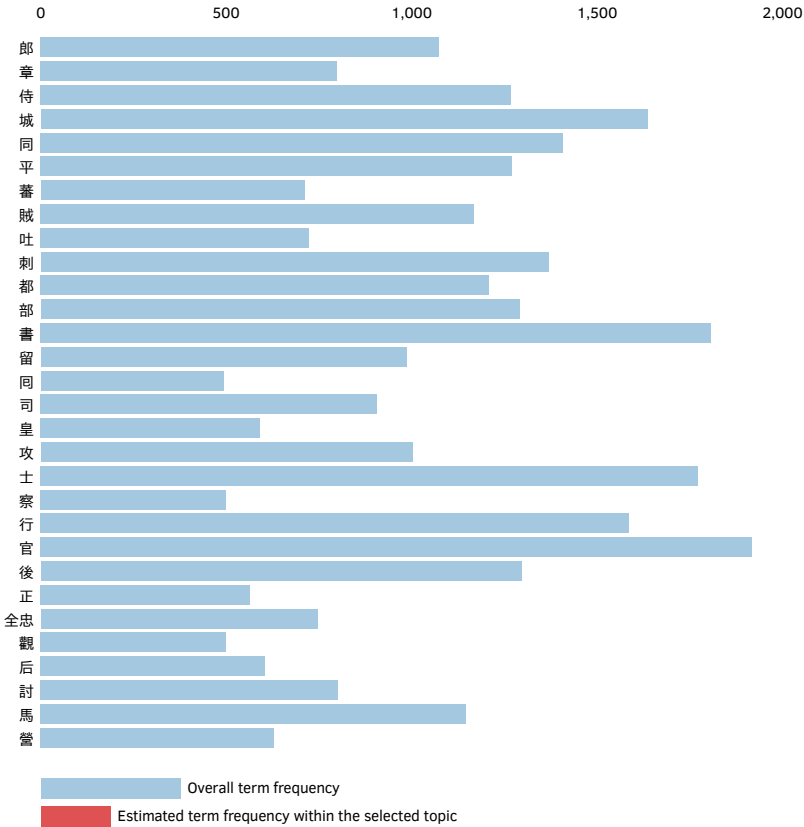
20 topics

Out[6]:

Selected Topic: 0 | Previous Topic | Next Topic | Clear Topic

Slide to adjust relevance metric:(2)   λ = 1

0.0   0.2   0.4   0.6   0.8   1

Intertopic Distance Map (via multidimensional scaling)

Top-30 Most Salient Terms(1)

Marginal topic distribtion

2%

5%

10%

Overall term frequency
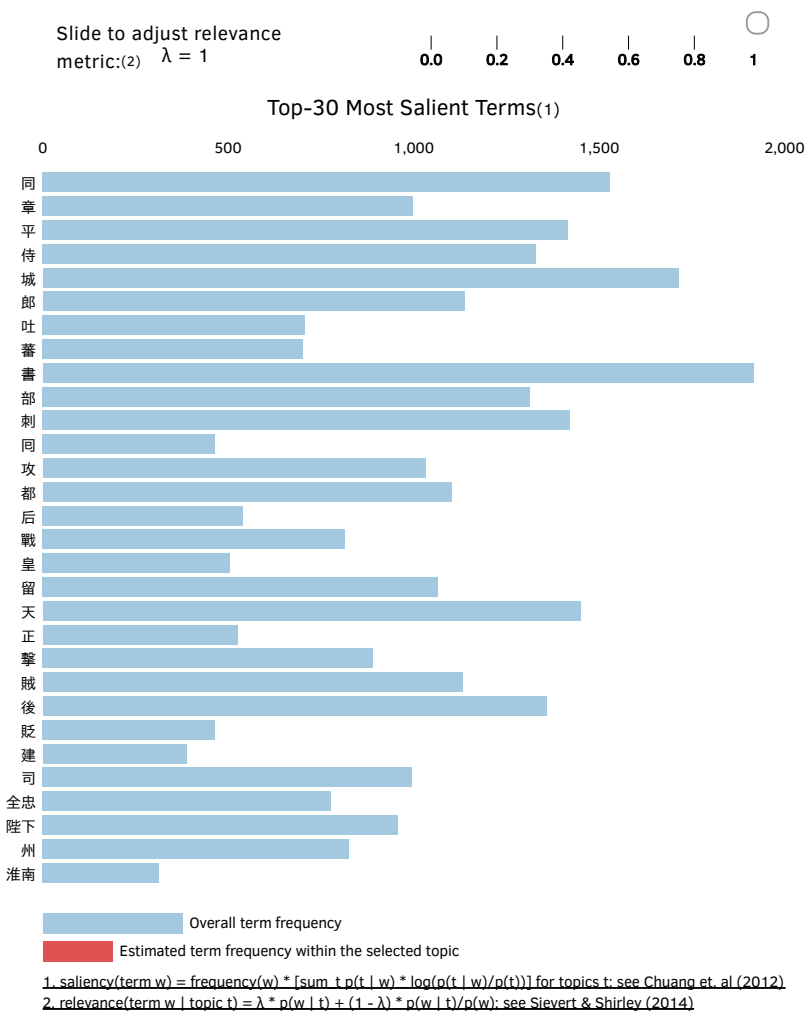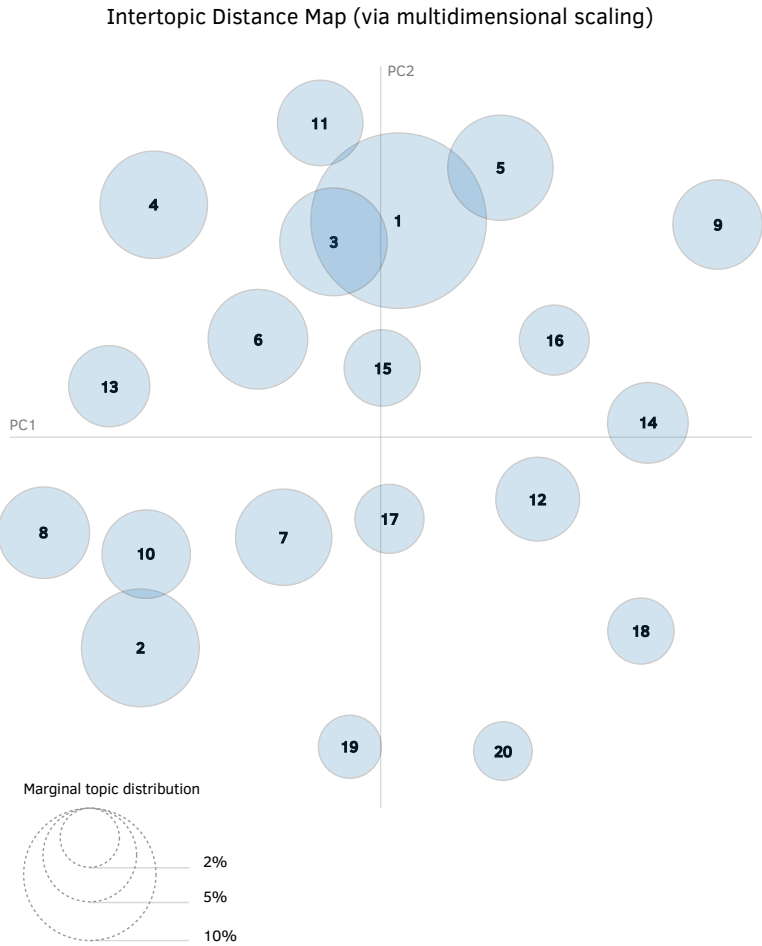
Estimated term frequency within the selected topic

1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t: see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w): see Sievert & Shirley (2014)

In [7]:
```
display(HTML("<h1>40 topics</h1>"))
pyLDAvis.display(vis[1])
```

40 topics

Out[7]:

Selected Topic: 0 | Previous Topic | Next Topic | Clear Topic |

Slide to adjust relevance metric:(2)   λ = 1

| 0.0 | 0.2 | 0.4 | 0.6 | 0.8 | 1 |

Intertopic Distance Map (via multidimensional scaling)

PC2

11

5

4

1

3

9

6

15

16

13

14

PC1

8

10

7

17

12

2

18

19

20

Marginal topic distribution

2%

5%

10%

Top-30 Most Salient Terms(1)

| 0 | 500 | 1,000 | 1,500 | 2,000 |

同
章
平
侍
城
郎
吐
蕃
書
部
刺
同
攻
都
后
戰
皇
留
天
正
擊
賊
後
貶
建
司
全忠
陛下
州
淮南

Overall term frequency

Estimated term frequency within the selected topic

1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t: see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w): see Sievert & Shirley (2014)
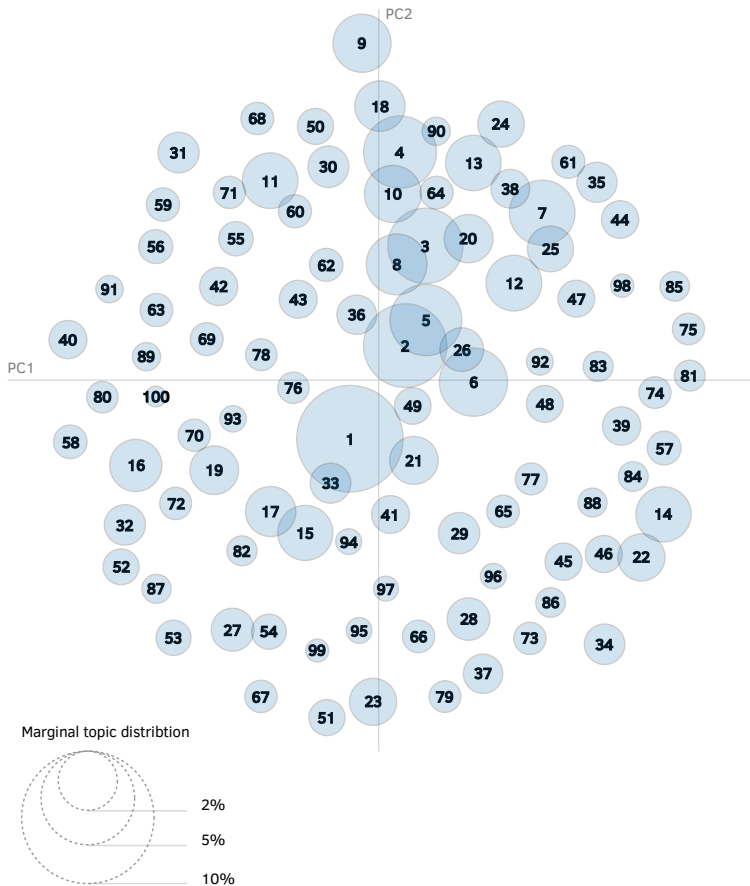
```
In [8]:  display(HTML("<h1>60 topics</h1>"))
         pyLDAvis.display(vis[2])
```

60 topics

Out[8]:

Selected Topic:[0] | Previous Topic | Next Topic | Clear Topic |

Slide to adjust relevance metric:(2)   λ = 1

| | | | | | |
|0.0|0.2|0.4|0.6|0.8|1|

Intertopic Distance Map (via multidimensional scaling)

Marginal topic distribtion

2%

5%

10%

Top-30 Most Salient Terms(1)

書
城
蕃
都
吐
郎
章
刺
后
平
賊
侍
皇
同
留
加
士
尚
全忠
司
部
奏
同
擊
相
正
攻
陛下
天
突

■ Overall term frequency
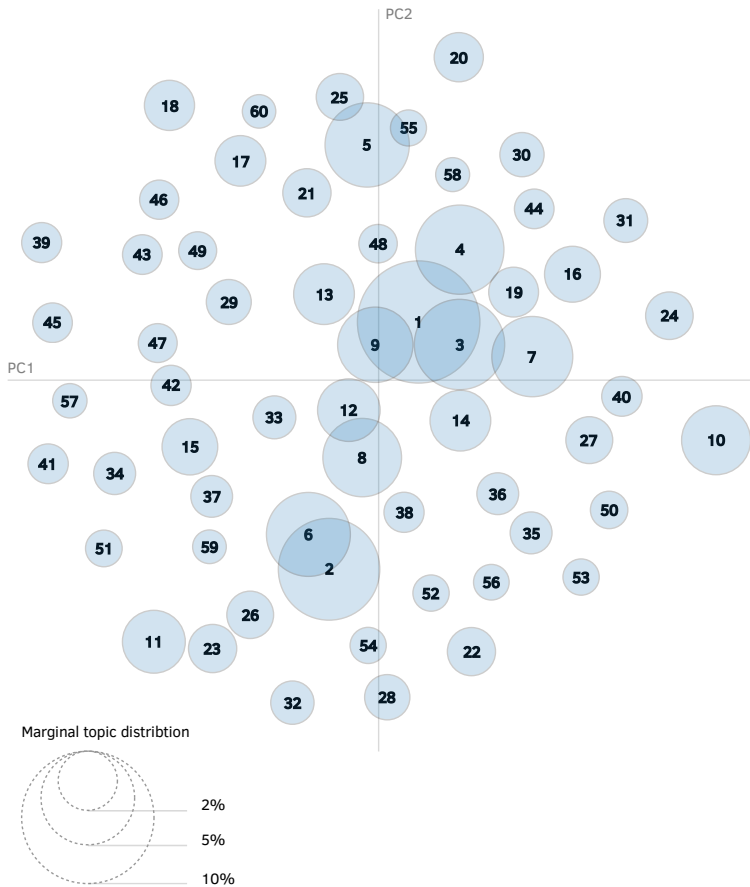
■ Estimated term frequency within the selected topic

1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t: see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w): see Sievert & Shirley (2014)

```
In [9]:  display(HTML("<h1>80 topics</h1>"))
         pyLDAvis.display(vis[3])
```
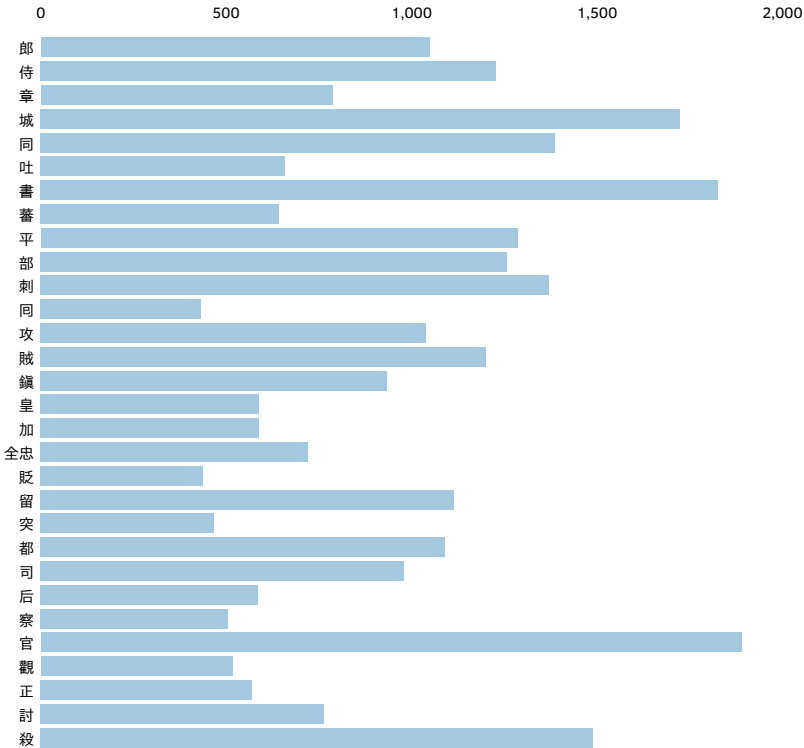
80 topics

Out[9]:

Selected Topic: 0 | Previous Topic | Next Topic | Clear Topic |

Slide to adjust relevance metric:(2)   λ = 1

| 0.0 | 0.2 | 0.4 | 0.6 | 0.8 | 1 |

Intertopic Distance Map (via multidimensional scaling)

PC2

PC1

Marginal topic distribtion

2%

5%

10%

Top-30 Most Salient Terms(1)

| 0 | 500 | 1,000 | 1,500 | 2,000 |

郎
侍
章
城
同
吐
書
蕃
平
部
刺
同
攻
賊
鎮
皇
加
全忠
貶
留
突
都
司
后
察
官
觀
正
討
殺

▇ Overall term frequency
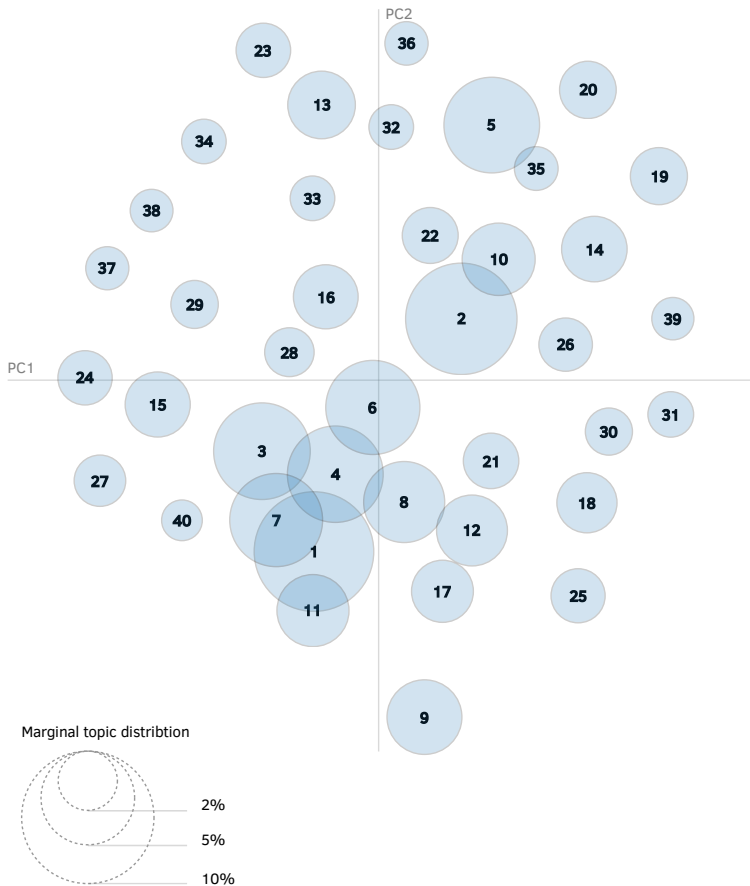
▇ Estimated term frequency within the selected topic

1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t: see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w): see Sievert & Shirley (2014)

In [10]:
```
display(HTML("<h1>100 topics</h1>"))
pyLDAvis.display(vis[4])
```
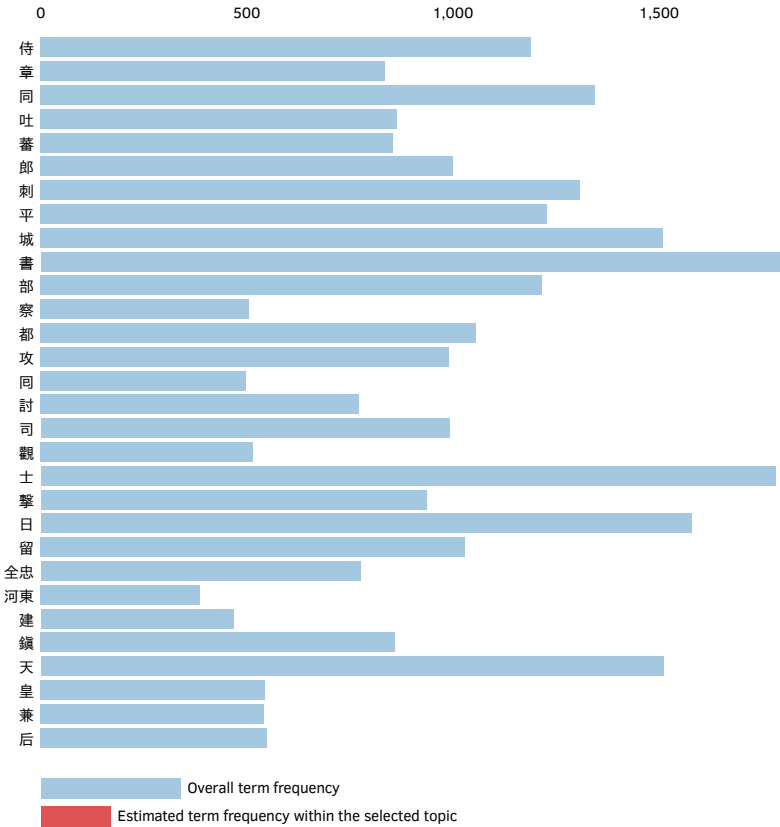
100 topics

Out[10]:

Selected Topic: 0 | Previous Topic | Next Topic | Clear Topic |

Slide to adjust relevance metric:(2) $\lambda = 1$

| | | | | | |
|0.0|0.2|0.4|0.6|0.8|1|

Intertopic Distance Map (via multidimensional scaling)

Marginal topic distribtion

2%

5%

10%

Top-30 Most Salient Terms(1)

Overall term frequency

Estimated term frequency within the selected topic

1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t: see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)