

```
In [ ]: import pyLDavis
topic_term_dists = topicmod.cv_model.get_topics() # transpose to make shape (num_terms, num_topics)
doc_topic_dists = topicmod.doc_mat# cv_model.get_document_topics(topicmod.tfidf_mat, minimum_probability=0)
# doc_topic_dists = [[tup[1] for tup in lst] for lst in doc_topic_dists] # convert list of tuples to just list
doc_lengths = [len(doc) for doc in gensim_statements]
vocab = list(dict_gensim_statements.token2id.keys())
term_frequency = dict_gensim_statements.cfs

# Use pyLDavis
vis_data = pyLDavis.prepare(
    topic_term_dists=topic_term_dists,
    doc_topic_dists=doc_topic_dists,
    doc_lengths=doc_lengths,
    vocab=vocab,
    term_frequency=list(term_frequency.values())
)

print("Intertopic distance map for C_V Score\n\n")
pyLDavis.display(vis_data)
```

Intertopic distance map for C_V Score

Out[]:

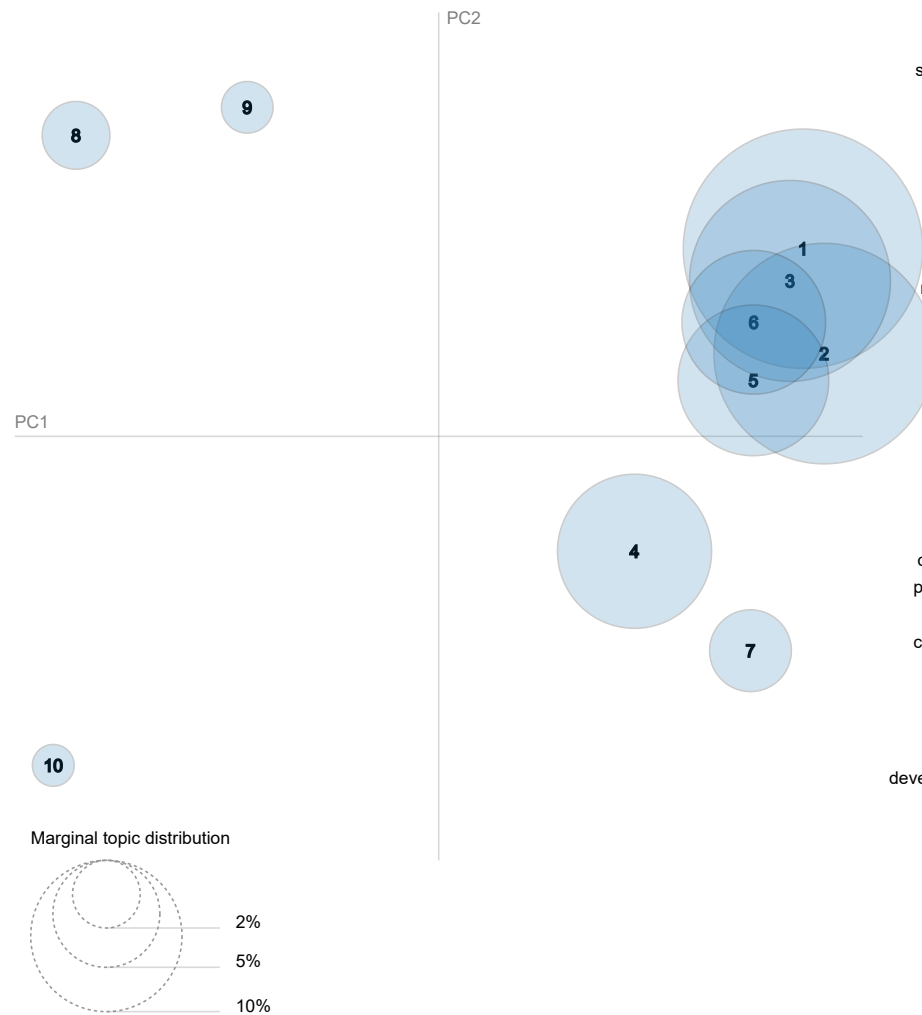
Selected Topic:

Slide to adjust relevance metric:(2)

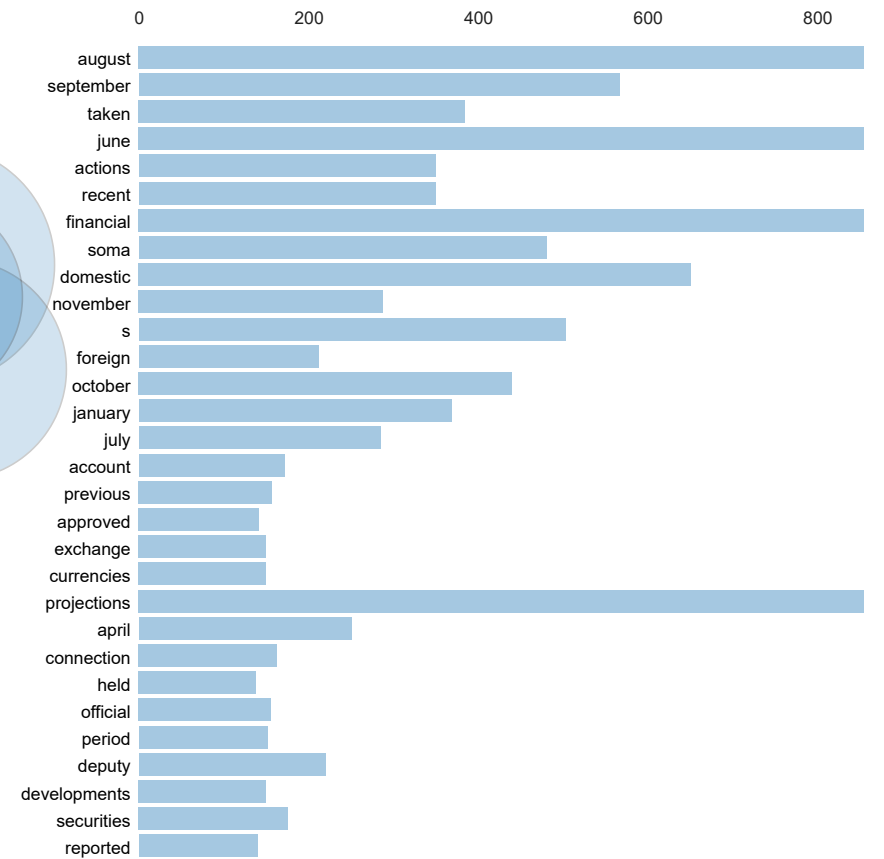
$\lambda = 1$

0.0 0.2 0.4 0.6 1.0

Intertopic Distance Map (via multidimensional scaling)



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
2. relevance(term w | topic t) = $\lambda * p(w | t) + (1 - \lambda) * p(w | t)/p(w)$; see Sievert & Shirley (2014)

```
In [ ]: topic_term_dists = um_topicmod.cv_model.get_topics() # transpose to make shape (num_terms, num_topics)
doc_topic_dists = um_topicmod.doc_mat# cv_model.get_document_topics(topicmod.tfidf_mat, minimum_probability=0)
# doc_topic_dists = [[tup[1] for tup in lst] for lst in doc_topic_dists] # convert list of tuples to just list
```

```
doc_lengths = [len(doc) for doc in gensim_statements]
vocab = list(dict_gensim_statements.token2id.keys())
term_frequency = dict_gensim_statements.cfs

# Use pyLDAvis
vis_data = pyLDAvis.prepare(
    topic_term_dists=topic_term_dists,
    doc_topic_dists=doc_topic_dists,
    doc_lengths=doc_lengths,
    vocab=vocab,
    term_frequency=list(term_frequency.values())
)
print("Intertopic distance map for UMass score\n\n")
pyLDAvis.display(vis_data)
```

Intertopic distance map for UMass score

Out[]:

Selected Topic:

Previous Topic

Next Topic

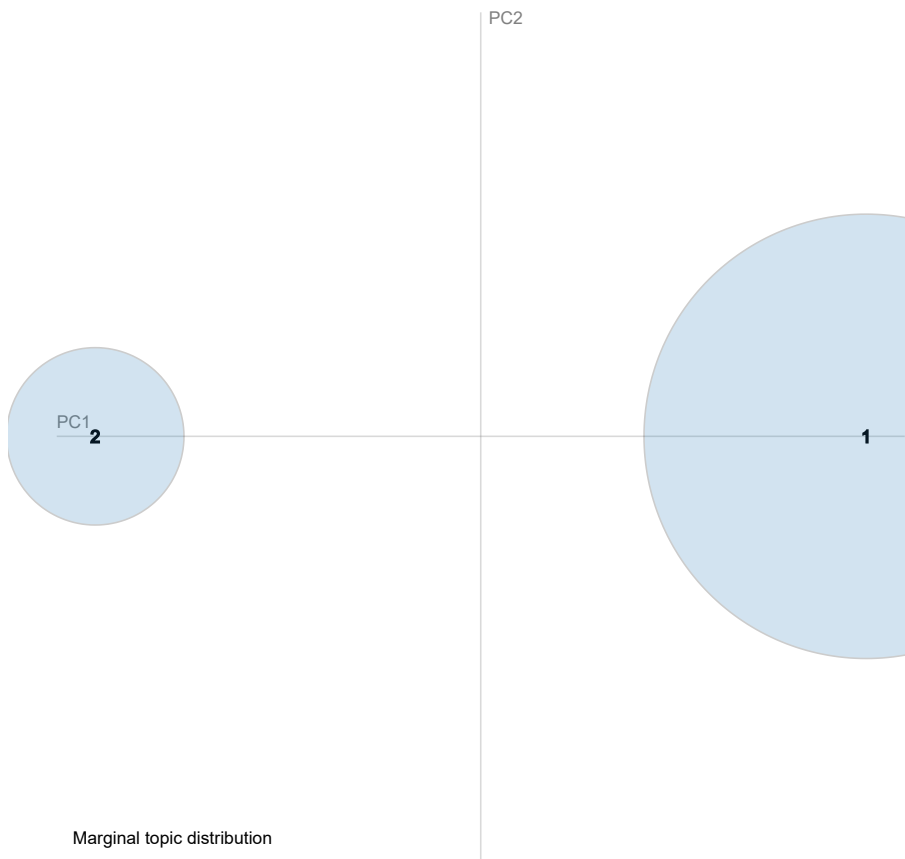
Clear Topic

Slide to adjust relevance metric:(2)

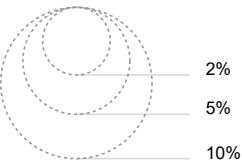
$\lambda = 1$

0.00.20.40.60

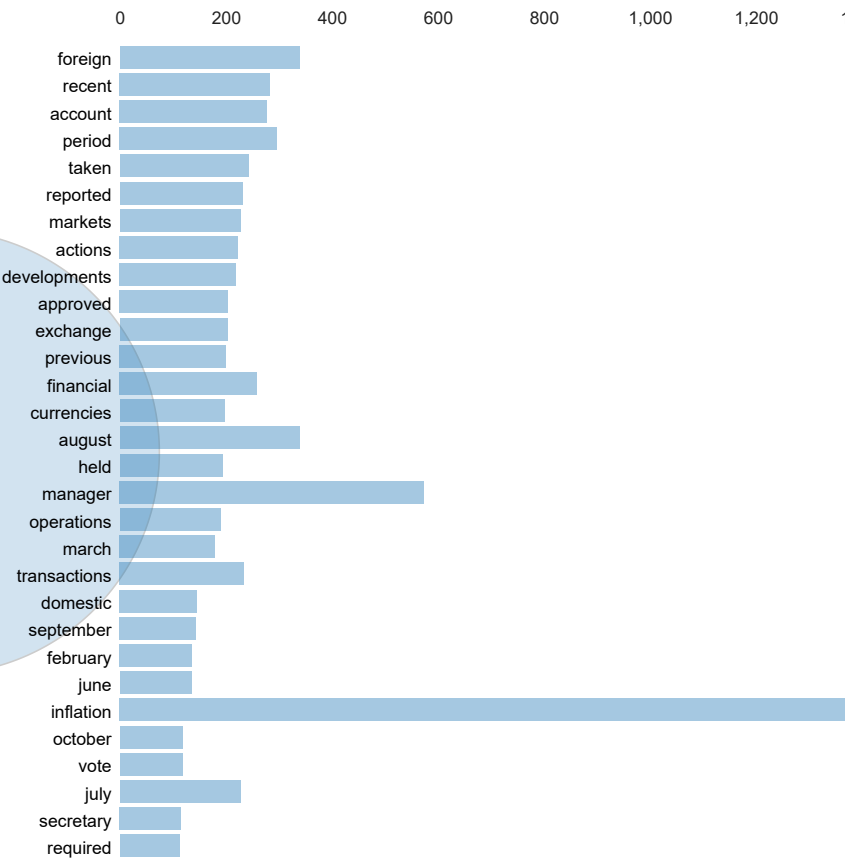
Intertopic Distance Map (via multidimensional scaling)



Marginal topic distribution



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
2. relevance(term w | topic t) = $\lambda * p(w | t) + (1 - \lambda) * p(w | t)/p(w)$; see Sievert & Shirley (2014)

In []: