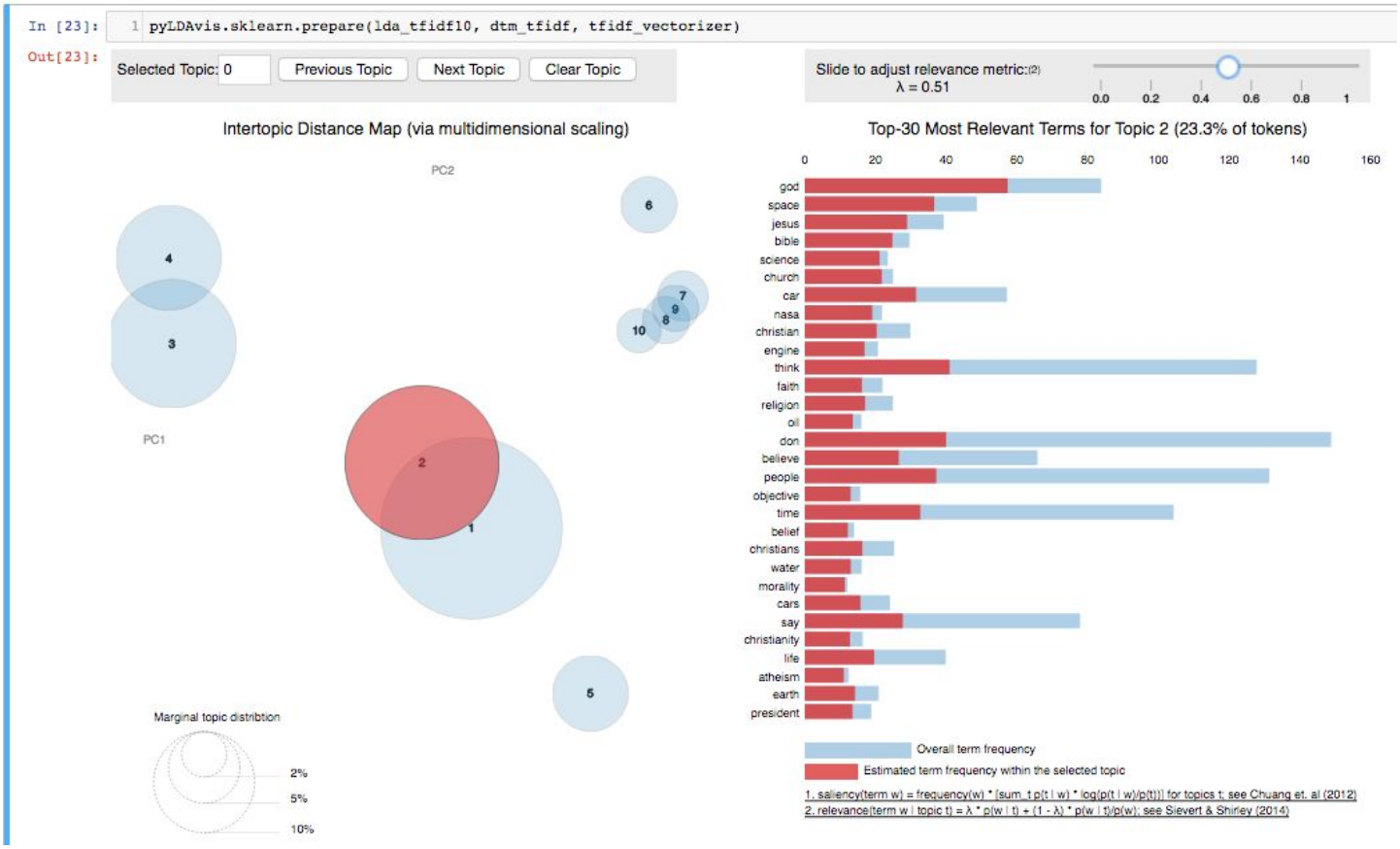
The following project relied on the 20newsgroups dataset to explore different ways of analyzing topics. Using the appropriate regular expressions, the data was prepared and tested on CountVectorizer to ensure the model was reading the text properly. This was first done in the Jupyter notebook to avoid complications with the tokens later on.

With temporal-based algorithms it can make sense to include stop words, yet in this case it was not helpful to include them. To prove this two CountVectorizer models were instantiated: one model with stop words and another model without stop words. The model with stop words had 264 additional words. Using yellowbrick's Frequency Distribution Visualizer (FreqDistVisualizer), it was clear that the model including stop words had ~20 words that occurred between 3-4 times more frequently than the top 5 words in the model with no stop words.

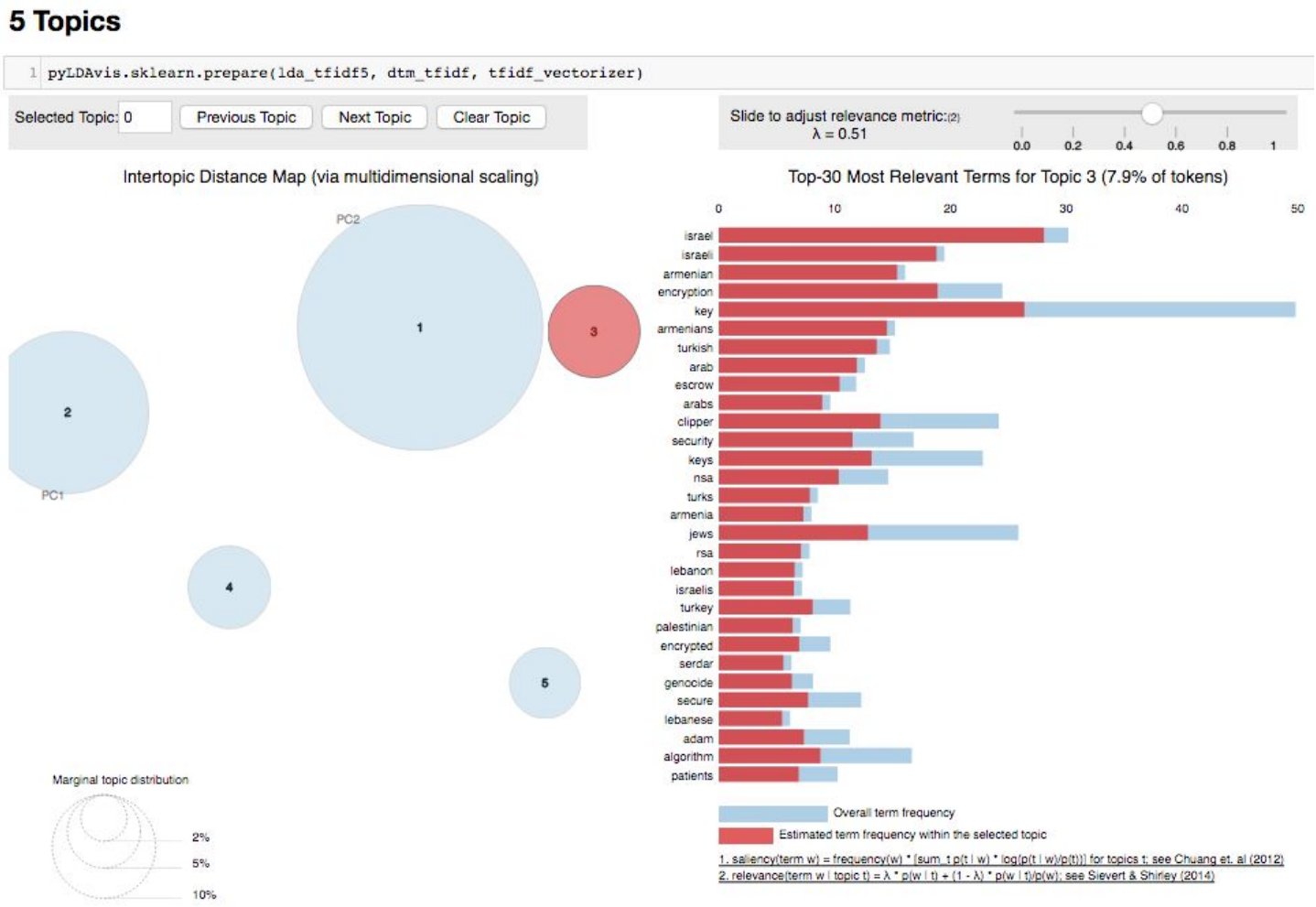
To get an idea of what clusters would naturally appear in the corpus, yellowbrick's t-SNE visualizer showed the total number of documents and how they organize by newsgroup. In the list of newsgroups one can identify between 6-8 topics depending on context (religion can be political; electronics can be in both scientific papers and consumer digests).

After instantiating pyLDAvis for both models (with and without stop words), each model was tested using a predetermined number of topics (5 and 10) to judge model performance. 5 and 10 topics were chosen since one could judge that the newsgroups had 6-8 different themes depending on how specific or general one was judging. Because this is unsupervised data, there is no objective measure of goodness. Using subjective investigation of the data for its distribution and topic identification led to the conclusion that the 10 topic model was best. The same process was applied using TF-IDF vectorizing and it too showed that the 10-topic model performed best.

10 Topics



For example, the ten topic model was able to visualize and discern the subtleties between articles about religion in the Middle East and US political news in the Middle East.



To further enhance standard approaches using SVD on a term frequency data set, Latent Dirichlet Analysis (LDA) was investigated. In addition to the term frequency embedding a second term topic matrix was calculated by statistical sampling methods.

The tool pyLDAvis allows for the user to influence the clustering based on term topic information. The amount of influence the term-topic matrix provides is controlled by a lambda parameter. A lambda of 0 means how exclusive a word is to a topic (jargon), whereas a lambda of 1 means that how probable a word is to appear in a topic (common or colloquial terms).

To better visually demonstrate how the topics interact, the three inputs for the multi-dimensional scaling hyperparameter were compared to see which best demonstrated to an audience the interrelatedness of these topics. Of the three, metric multidimensional scaling (mmds) was best able to sort the topics in four quadrants that made sense. Topics 1 and 2 were able to group articles about the Middle East, 3 and 4 were about computers, 5 was about sports, 6 and 7 were about security and defense, while 8, 9, and 10 were clustered as science-related.

Metric Multidimensional Scaling

In [28]:
```
1 pyLDAvis.sklearn.prepare(lda_tfidf10, dtm_tfidf, tfidf_vectorizer, mds='mmds')
```

Out[28]: