**Topic Modeling**

**Intro**

My goal was to use modern topic modeling techniques to identify topics in a custom domain. Topic models are generally illustrated with pre-prepared toy datasets which result in high performance, but will perform quite differently with smaller unrelated datasets. For example, topic models often use the 20 News Groups dataset which has 20,000 news articles, an entirely different size and domain than the couple hundred abstracts I will use from a small catalog of scientific articles. I will describe what I learned about building and assessing topic models for a small scientific corpus as well some of the challenges along the way which led me to BERTopic, an easy-to-use robust topic model.

**The Old Standbys**

Topic modeling is challenging in that not only is it generally unsupervised learning, but topics generated are a list of words which must be interpreted. Not only that, but documents can be a mixture of topics. I tend to try the simplest, most common method first. K-means doesn’t work well for topic modeling for multiple reasons, one being it cannot account for multiple topics in one document. Fuzzy clustering methods are more appropriate and density-based clustering methods such as HDBSCAN tend to work better for topic modeling. Latent Dirichlet Allocation (LDA) has been the go-to method for topic modeling for quite a while and uses the bag-of-words approach. In addition to LDA, one of the other most common topic models is Latent Sematic Analysis (LSA or LSI). I experimented with HDBSCAN, LSA, and LDA before attempting the deeper models to see how well they performed and to compare results.

**BoW Data Preparation**

I used NLTK and Gensim for the bag-of-words (BoW) data preprocessing, but SpaCy is a fine substitute for the same task. NLTK, or Natural Language Toolkit, as its name suggests, is a broad library for natural language processing, whereas Gensim is mainly a package for topic modeling and document similarity.

1. Tokenization – split words, lowercase, and remove punctuation
2. Stopwords are removed using Genism
3. Words are stemmed with NLTK SnowballStemmer
4. Using Genism, create a dictionary from the processed documents containing the number of times the word appears
5. Create TF-IDF of Bag of Words

**HDBSCAN**

Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) works better than centroid-clustering methods when clusters are of irregular shapes, noisy, and with different densities. Unlike LSA and LDA, HDBSCAN does not require a number of clusters as input, which is convenient when the number of topics is unknown. Higher dimensional embeddings from transformer models can be reduced for use in this model as well, as seen in Top2Vec and BERTopic. I will talk about this more below.

* **Sklearn TF\_IDF Vectorization** Why doesn’t Gensim models TF-IDF transformed corpus work?
* **Train**
* **Test/Analyze**
* **Visualize**

**LSA Model**

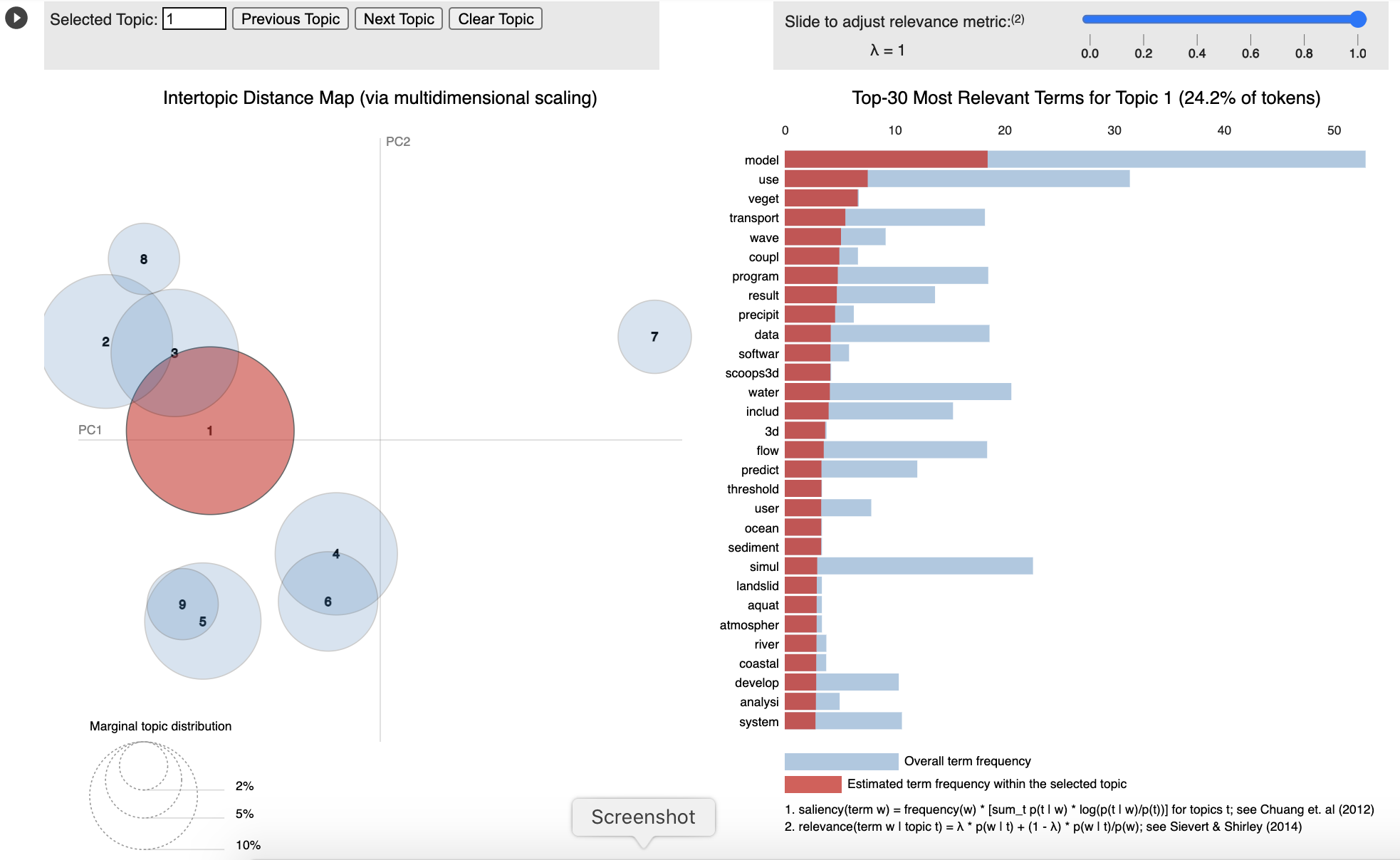
LSA assumes that words close in meaning will occur in similar texts. The input is a document-term tf-idf matrix. Singular value decomposition (SVD) is applied to the matrix. Interpreting LSA coefficients to determine which topics relate to each document is difficult. The coefficients are not easily interpretable from SVD, unlike the coefficients from LDA. I attempt to get the best related topic for each document by selecting the coefficient with the highest absolute value, not ideal.

* **BoW Data Preparation**
* **Train**
* **Test/Analyze**
* **Visualize**

**LDA Model**

Latent Dirichlet Allocation (LDA) is a generative statistical model which is the Bayesian version of probabilistic LSA (pLSA). LDA is an improvement over pLSA in that it yields better disambiguation of words, is not as vulnerable to overfitting with large text corpus, and is easier to scale.

* **BoW Data Preparation**
* **Train**
* **Test/Analyze**
* **Visualize** The pyLDAvis package helps visualize the overall frequency of terms, the topic clusters, and terms associated with each topic using the LDA topic model. The model needs a matrix of topic-term probabilities and document-topic probabilities, thus can’t be used with LSA or HDBSCAN.



**BERT/Sentence Transformer Data Preparation**

Lacking domain-expert annotators and a deep understanding of the scientific corpus, it’s necessary to extract as much information from the observable features from the text. Pre-trained transformer models, such as BERT, should be able to extract more complex features from the text for the topic models. Before finding BERTopic, I started working with Contextualized Topic Models (CTM) and Top2Vec.

* **SPECTER**
* **Other embeddings:** Doc2Vec, Word2Vec, GloVe, ELMo, BERT, fastText, Gensim, etc.

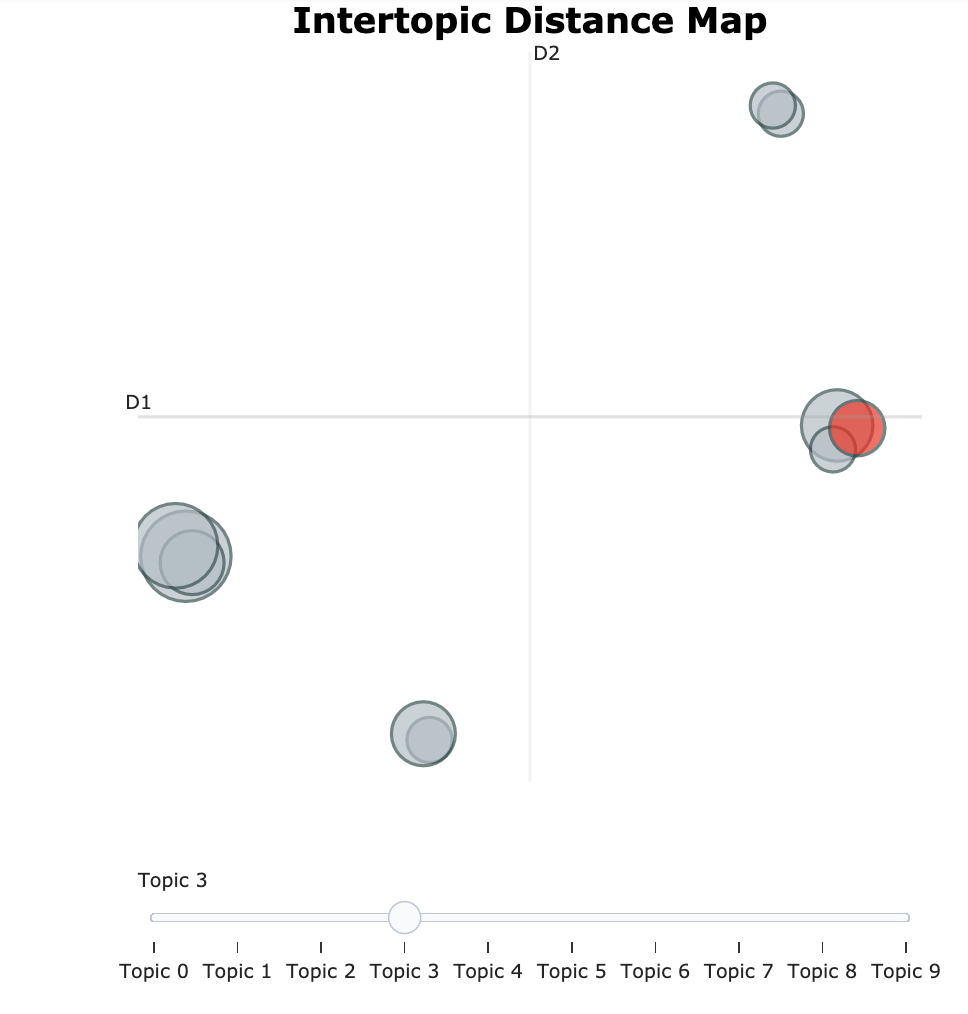
**Top2Vec**

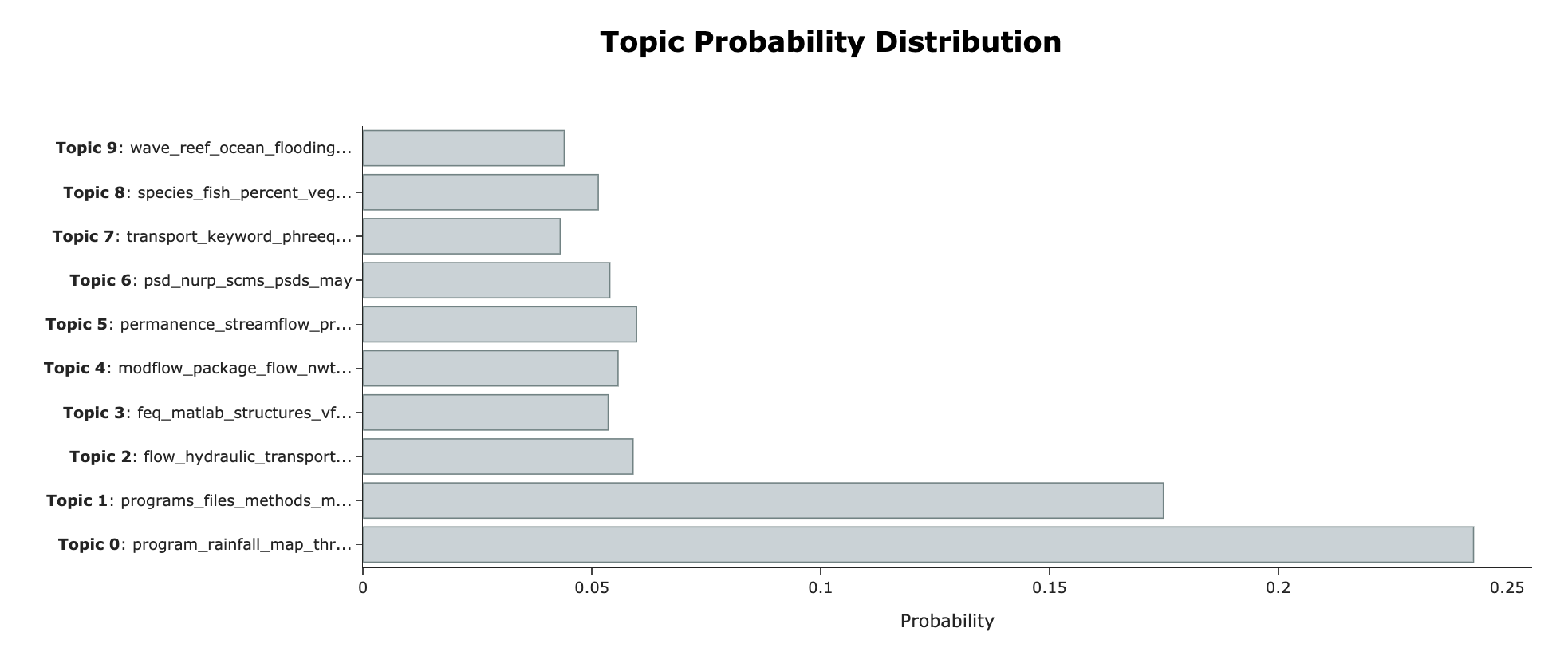
Error for small datasets because it places everything into one topic and crashes. Should be able to adjust the HDBSCAN parameters, but didn’t work passed in as a dictionary as per the documentation. Other limitations include the type of embedding. It accepts: \_\_\_

**BERTopic**

built on top of Top2Vec with better API. results and vis

* **Little to No Preparation for SentenceTranformers**
* **Train**
* **Test/Analyze**
* **Visualize**





CTM – results

Uses BoW and BERT embeddings,

**Comparing Results**

Evaluating results with a test set can reveal how much the model relies on spurious relationships.

* Perplexity doesn’t consider semantic associations between words and may not be the best metric for topic models.
* Coherence measures the score of a single topic by measuring the semantic similarity between words in the topic. If two words in a topic belong together, you would expect them to show up together frequently.
* Note that the coherence model requires the original text (not tf-idf, etc.)
* Can’t get the coherence score for HDBSCAN and BERTopic – needs ‘get\_topics’ method
* The closer UMass coherence score to 0, the better topic coherence.

**Conclusion**

Density-based clustering (HDBSCAN) relies on having enough data to separate dense areas. The higher the dimension, the more data needed. There may not be enough data for this model.