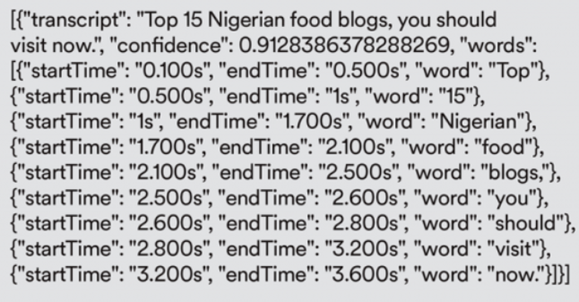
Topic Modeling using Latent Dirichlet Allocation

The purpose of the LDA model is to use unsupervised learning to classify all transcripts into topics. These are predefined topics that Spotify uses to categorize podcasts. Once a user searches for an ad-hoc the LDA model will classify the ad-hoc into one of the pre-fitted 62 topics. The tools used to build the LDA model are MySQL database and Python’s nltk, joblib and genism libraries. For all incoming podcast transcripts, it is assumed that a transcript is a combination of topics and the LDA model will fit the transcript into the best fit topic(s). Having all transcripts fit into topics will allow for an easier retrieval step when the DAN is performed to retrieve a sub-corpus of transcripts.

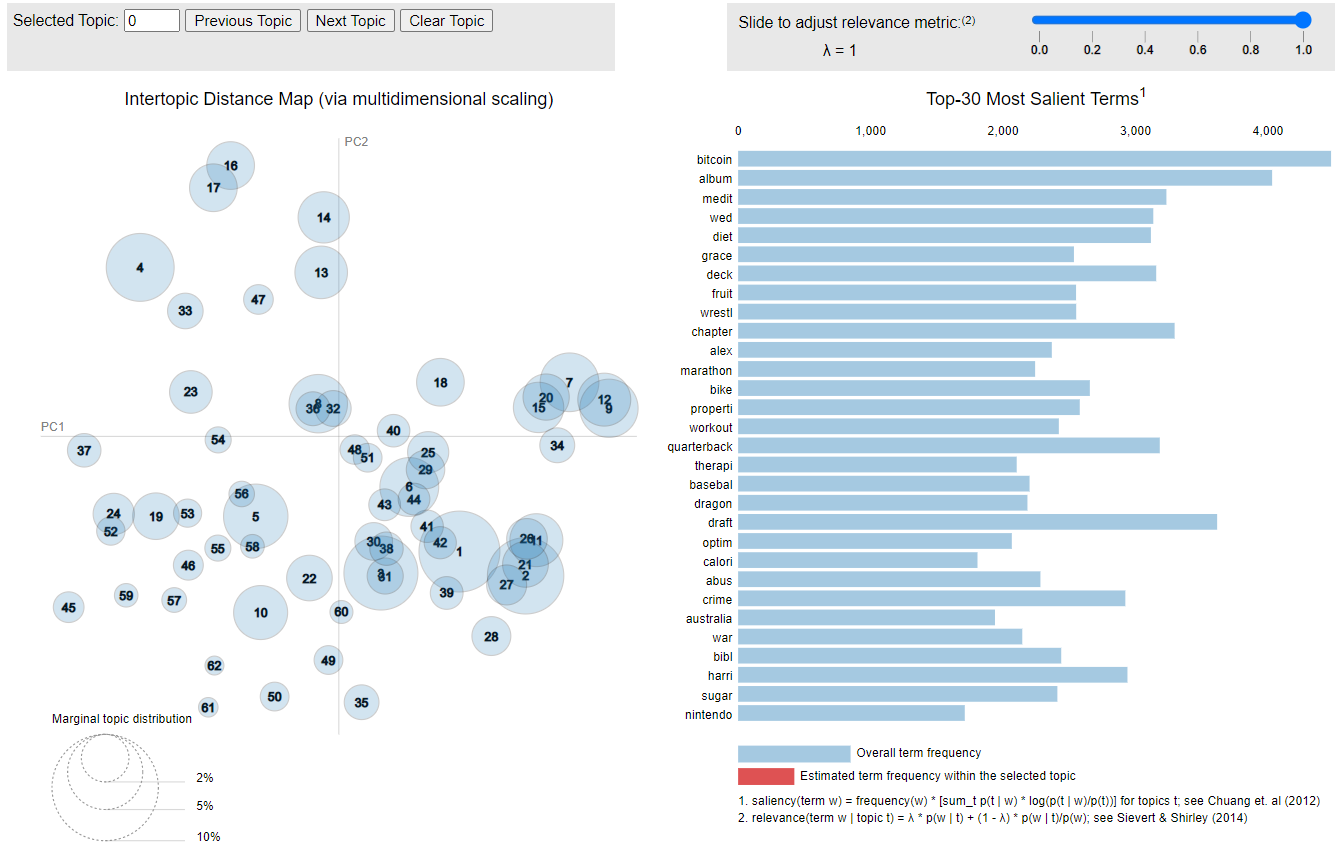
To create the LDA model a training set is needed. An arbitrary amount of 10% of the original 100,000 set of transcripts was chosen at random to be in the training set. Originally all the transcripts were saved into JSON files where each file was one single episode, see Fig. 1. The entire transcript is split into 30sec chunks and each word have a start and ending timestamp. For building the LDA model the timestamps and individual words were not needed. A Python script is used to parse out only the transcript and combined all 30sec chunks into one large text that would be placed into a MySQL table. See the file lda\_mysql.py on GitHub for the complete implementation of this process.

  
Fig. 1 JSON File Example

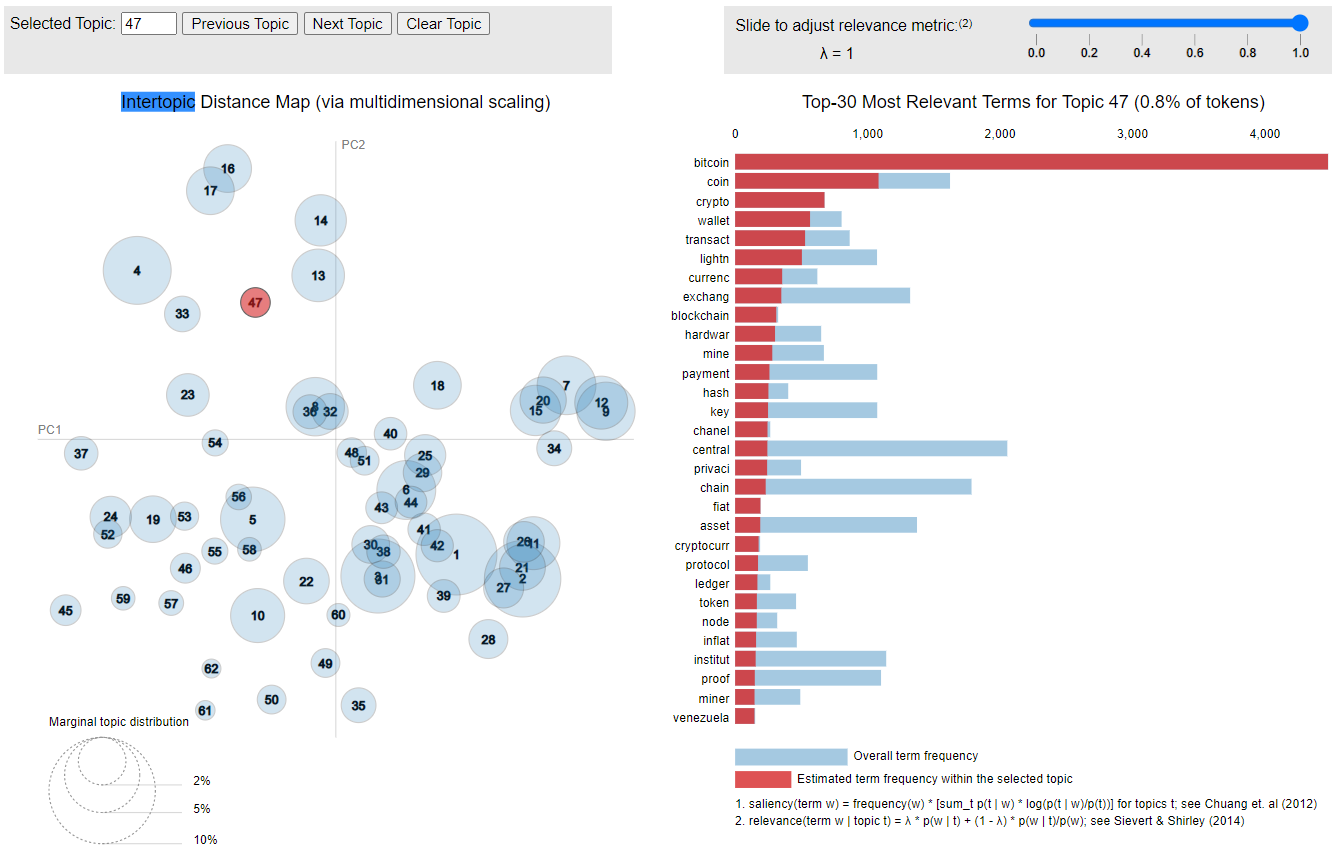
Once the table was created all the transcripts first underwent tokenization to remove any punctuation and the boundary of each word is found to split the text into smaller units. Then words that have fewer than 3 characters are removed such as “I”, “you”, “we”, etc. Each word is then lemmatized to find a common root between words and then the words are stemmed into root form. Now that the preprocessing step is complete a dictionary is created to tell us the number of times a word appears in the training set. Then the dictionary filters out very rare and very common words so that common words in speech, such as “thing”, will not have a high word distribution over several topics. A bag of words corpus is created, which is a bag of words vector for each transcript within the training set. A bag of words vector tells us how many times a word appears in the transcript.

Now that the data has been preprocessed the LDA model is trained by using 62 topics and 30 passes. To ensure that this model is not being rebuilt every time we do an ad-hoc retrieval it is pickled into a file to be later loaded and reduce run times, see files 62topiclda.pkl, bow\_corpus.pkl, and dicitionary.pkl on GitHub.

Fig. 2 shows the visualization of the LDA model and the distribution of words over all 62 topics. The word “bitcoin” appears over 4,000 times and it appears solely in topic 47, see fig. 3.

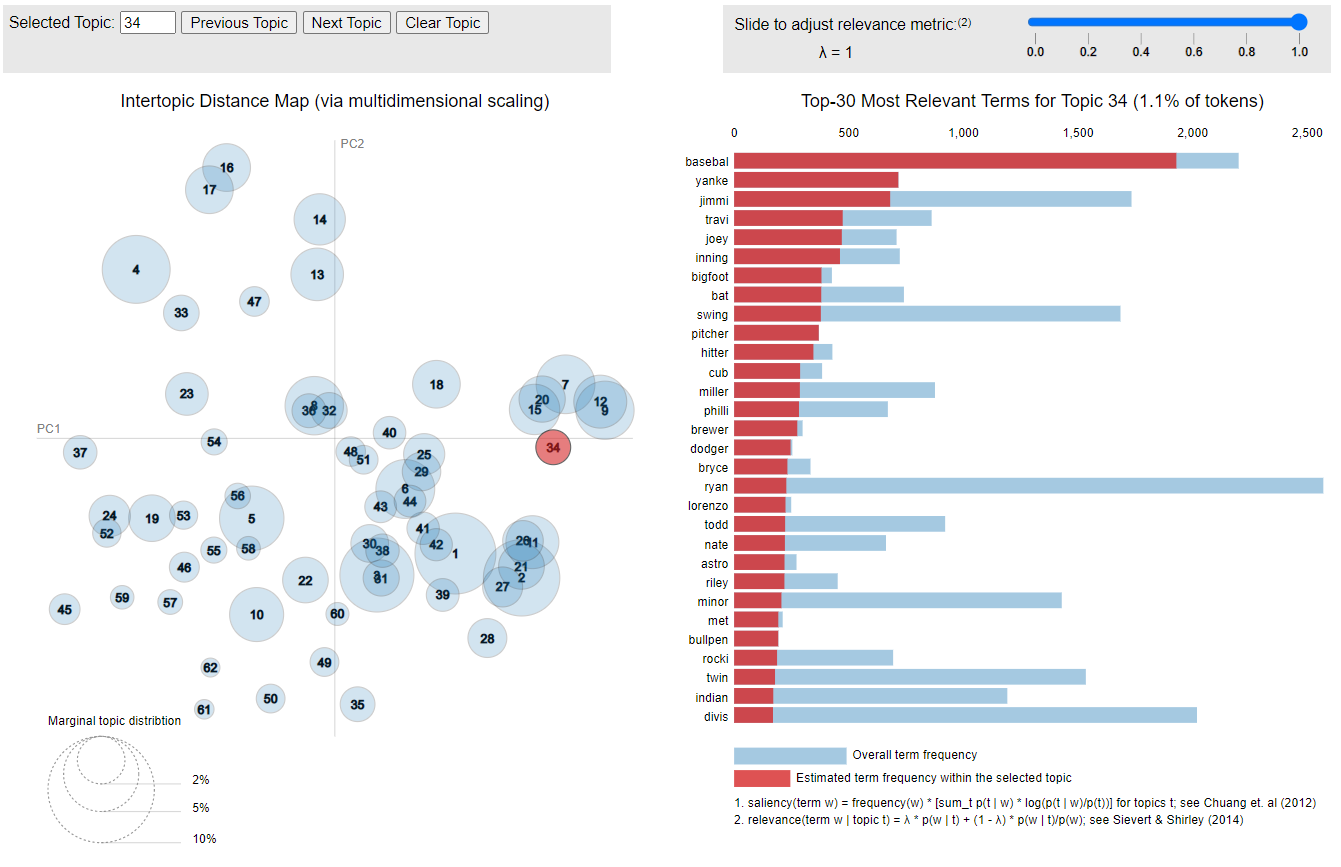
  
Fig. 2 LDA Model Visualization

Other relevant words in topics 47 are mainly about cryptocurrency and therefore we may classify that topic 47 is the technology topic.

  
Fig. 3 Topic 47 Word Distribution

To determine if the LDA model results are meaningful the user could subjectively interpret the results as I did for topic 47; However, human judgement does not tell us the usefulness of the word to topic distribution. For this, the coherence score is calculated. The coherence score reveals how similar the most relevant terms in a single topic are to each other. For example, in fig.3 we can see that ‘bitcoin’ and ‘blockchain’ are semantically similar. To compute the coherence score, a conditional probability called, normalized pointwise mutual information along with cosine similarity are used to measure the distance between the terms x and y within a topic. The results are on a [-1,+1] scale where -1 never occurs together, 0 for independence, and +1 for complete co-occurrence. The coherence score of the LDA model is 0.537.

For testing this model, the user will enter in an ad-hoc such as ‘Babe Ruth’ and a Wikipedia search is used to grab the summary of Babe Ruth. The summary is then preprocessed and the LDA model is applied to find that the highest topic probability is 0.456 which is topic 34, baseball. See fig. 4.

  
Fig. 4 Topic 34 Baseball

A user may also enter in an ad-hoc such as ‘How to breath while running?’. A Wikipedia search yields no results, so a Google searched is used to find the first relative article and a summary is extracted. The summary is preprocessed and the LDA model is applied to find the topics with the highest probability. Since every podcast may consist of multiple topics it is also true that an ad-hoc may consist of multiple topics. Let the minimum probability be .2 and the model will ignore low scoring topics and the two highest scoring topic probabilities are:

Topic Distribution:

[(25, 0.28938222), (32, 0.33610022)]

Topic: 25

Words: 0.060\*"marathon" + 0.053\*"workout" + 0.053\*"bike" + 0.033\*"runner" + 0.023\*"olymp" + 0.019\*"boston" + 0.015\*"jason" + 0.014\*"trainer" + 0.012\*"recoveri" + 0.010\*"trail"

Topic: 32

Words: 0.042\*"yoga" + 0.032\*"exhal" + 0.031\*"inhal" + 0.026\*"squat" + 0.021\*"ankl" + 0.019\*"pose" + 0.018\*"bend" + 0.018\*"spine" + 0.016\*"hip" + 0.015\*"toe"

Although the training set used is small in proportion to the dataset, the result is an accurate topic classification of an ad-hoc with an LDA model of a coherence score of .537. For the purpose of this project the proposed LDA model is accepted and will be utilized for the retrieval phase.